Advances in Intelligent Systems and Computing 566

Phayung Meesad
Sunantha Sodsee
Herwig Unger *Editors*

Recent Advances in Information and Communication Technology 2017

Proceedings of the 13th International Conference on Computing and Information Technology (IC2IT)

Springer

Advances in Intelligent Systems and Computing

Volume 566

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within "Advances in Intelligent Systems and Computing" are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Advisory Board

More information about this series at http://www.springer.com/series/11156

Phayung Meesad · Sunantha Sodsee
Herwig Unger
Editors

Recent Advances in Information and Communication Technology 2017

Proceedings of the 13th International Conference on Computing and Information Technology (IC2IT)

Springer

Editors
Phayung Meesad
Faculty of Information Technology
King Mongkut's University of Technology
North Bangkok
Bangkok
Thailand

Herwig Unger
Lehrgebiet Kommunikationsnetze
FernUniversität in Hagen
Hagen
Germany

Sunantha Sodsee
Faculty of Information Technology
King Mongkut's University of Technology
North Bangkok
Bangkok
Thailand

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-60662-0 ISBN 978-3-319-60663-7 (eBook)
DOI 10.1007/978-3-319-60663-7

Library of Congress Control Number: 2017943856

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Nowadays, the Internet is the biggest medium providing, maintaining and transporting different kinds of data for its users around the world. Not only with its evolution into the 'Internet of Things' and the tendency towards cloud computing, the Internet transformed itself into a significant source of (precious and usable) 'Big Data'. Therefore, it becomes increasingly clear that data and network structures as well as user behaviour cannot be considered in an isolated manner any longer. These changes, developments and multiple cross-connections are reflected by the new term 'Data Networks'.

All data generated by the Internet and its use definitely belong to the category of big data and require, due to their volume and dynamics, a new kind of efficient data mining methods to turn them into useful information and, ultimately, knowledge. Since a large portion of the data available are sensitive and often of a very private nature, their safe, secure and well-protected handling is increasingly gaining importance. Last but not least, interfaces are more and more incorporating multimedia and natural language processing, adding to their complexity of design and execution.

Considering these trends with special respect to big data, the purpose of the 13th International Conference on Computing and Information Technology (IC^2IT) is to present emerging algorithms, methods and technologies with a high degree of originality, novelty and innovation addressing the conference theme 'Mastering Data and Networking'. The 34 contributions to the conference were selected by the programme committee for oral presentation and inclusion in this book out of 101 submissions.

Following the above argument, the volume's major part constituted by the first two sections discusses various aspects of data mining and corresponding applications. Feature selection in high-dimensional spaces, suitable clustering mechanisms and predictions are still the major, not yet fully solved problems in this area. The key to most networking optimisation problems addressed in the subsequent Section 3 is to properly determine critical parameters. Beside speed and overhead optimisation, energy problems of autonomous systems become here more and more important. Then, in Section 4, light is shed on natural language processing, where

the aspects extraction of trends and popularity and recognition of emotions are becoming key issues beside the classical topics detection and classification.

The editors hope and wish that this collection of contributions covering different areas of computer science will spark, not only during the conference, inspiring discussions among colleagues. They intend to contribute to the deep understanding that is required to solve the problems raised by today's complex environments, and that interdisciplinary cooperation is beneficial for this purpose.

Last but not least, the editors would like to thank all authors for their submissions and the programme committee members for their great work and valuable time. This book could not have been completed without the effort of staff, especially of Ms. Watchareewan Jitsakul, of the Information Technology Faculty at King Mongkut's University of Technology North Bangkok as well as of all other collaborators in Thailand and abroad. Finally, we are grateful to Springer-Verlag and Janusz Kacprzyk as the editor responsible for the series 'Advances in Intelligent System and Computing' for their great support in publishing these conference proceedings already for the fifth time in sequence as well as for the smooth cooperation all along.

March 2017

Phayung Meesad
Sunantha Sodsee
Herwig Unger

Organization

Program Committee

M. Aiello	RUG, The Netherlands
T. Anwar	UTM, Malaysia
S. Auwatanamongkol	NIDA, Thailand
G. Azzopardi	RUG, The Netherlands
T. Bernard	Université de Reims, France
S. Boonkrong	KMUTNB, Thailand
D. Brodic	BOR, Serbia
J. Brown	ECU, Australia
S. Butcharoen	TOT, Thailand
S. Butdisuwan	MSU, Thailand
M. Caspar	TU Chemnitz, Germany
J. Chatree	SSKRU, Thailand
T. Chintakovid	KMUTNB, Thailand
T. Eggendorfer	HS Weingarten, Germany
M. Ghazali	UTM, Malaysia
M. Hagan	OSU, USA
W. Halang	FernUni, Germany
C. Haruechaiyasak	NECTEC, Thailand
K. Hashimoto	PSU, Thailand
S. Hengpraprohm	NPRU, Thailand
K. Hengproprohm	NPRU, Thailand
U. Inprasit	UBU, Thailand
U. Inyaem	RMUTT, Thailand
C. Jareanpon	MSU, Thailand
J. Kacprzyk	Polish Acad. of Sci., Poland
M. Kaenampornpan	MSU, Thailand
V. Khonchoho	PBRU, Thailand
M. Komkhao	RMUTP, Thailand

A. Kongthon	NECTEC, Thailand
P. Kovintavewat	NPRU, Thailand
S. Krootjohn	KMUTNB, Thailand
P. Kropf	Neuchatel, Switzerland
P. Kuacharoen	NIDA, Thailand
M. Kubek	FernUni, Germany
K. Kyamakya	AAU, Austria
U. Lechner	UniBw, Germany
Z. Li	FernUni, Germany
P. Meesad	KMUTNB, Thailand
A. Mikler	UNT, USA
A. Mingkhwan	KMUTNB, Thailand
L. Nguyen The	HNUE, Vietnam
K. Nimkerdphol	RMUTT, Thailand
S. Nuanmeesri	RSU, Thailand
J. Polpinij	MSU, Thailand
N. Porrawatpreyakorn	KMUTNB, Thailand
P. Prathombutr	NECTEC, Thailand
P. Saengsiri	TISTR, Thailand
P. Sanguansat	PIM, Thailand
T. Sarawong	RMUTK, Thailand
S. Smanchat	KMUTNB, Thailand
M. Sodanil	KMUTNB, Thailand
S. Sodsee	KMUTNB, Thailand
G. Somprasertsri	MSU, Thailand
O. Sornil	NIDA, Thailand
T. Srikhacha	TOT, Thailand
W. Sriurai	UBU, Thailand
T. Sucontphunt	NIDA, Thailand
P. Sukchovna	KRU, Thailand
S. Suranauwarat	NIDA, Thailand
W. Tang	CityU, Hongkong
D. Thammasiri	NPRU, Thailand
P. Thongchumnum	PSU, Thailand
J. Thongkam	MSU, Thailand
S. Tongngam	NIDA, Thailand
D.H. Tran	HNUE, Vietnam
H.-M. Tsai	NTU, Taiwan
P. Tuanpusa	RMUTT, Thailand
D. Tutsch	Wuppertal, Germany
H. Unger	FernUni, Germany
N. Utakrit	KMUTNB, Thailand
S. Valuvanathorn	UBU, Thailand
M. Weiser	OSU, USA
N. Wisitpongphan	KMUTNB, Thailand

N. Witthayawiroj	RMUTT, Thailand
A. Woodward	ECU, Australia
K. Woraratpanya	KMITL, Thailand
P. Wuttidittachotti	KMUTNB, Thailand

Organizing Partners

In cooperation with
King Mongkut's University of Technology North Bangkok, Thailand (KMUTNB)
FernUniversität in Hagen, Germany (FernUni)
Chemnitz University of Technology, Germany (TU Chemnitz)
Oklahoma State University, USA (OSU)
Edith Cowan University, Australia (ECU)
Hanoi National University of Education, Vietnam (HNUE)
Gesellschaft für Informatik, Germany (GI)
Rajamangala University of Technology Thanyaburi, Thailand (RMUTT)
Rajamangala University of Technology Krungthep, Thailand (RMUTK)
Mahasarakham University, Thailand (MSU)
Ubon Ratchathani University, Thailand (UBU)
Prince of Songkla University, Phuket Campus, Thailand (PSU)
National Institute of Development Administration, Thailand (NIDA)
Kanchanaburi Rajabhat University, Thailand (KRU)
Nakhon Pathom Rajabhat University, Thailand (NPRU)
Phetchaburi Rajabhat University, Thailand (PBRU)
Sisaket Rajabhat University, Thailand (SSKRU)
Council of IT Deans of Thailand (CITT)

Contents

Data Mining Applications

Complex Networks and Systems

Data Mining Methodologies

Applying Stochastic Evolutionary Algorithm for Correlation Control in Monte Carlo Simulation

Anamai Na-udom[1(✉)] and Jaratsri Rungrattanaubol[2]

[1] Department of Mathematics, Faculty of Science, Naresuan University, Phitsanulok, Thailand
anamain@nu.ac.th
[2] Department of Computer Science and Information Technology, Faculty of Science, Naresuan University, Phitsanulok, Thailand
jaratsrir@nu.ac.th

Abstract. This paper presents an application of stochastic evolutionary algorithm (SE) in generating correlated multivariate random samples for Monte Carlo simulation. The algorithm is applied to impose the correlation structure when the marginal distribution and correlation matrix are pre-specified. The performance of a proposed method is compared with the existing methods namely simulated annealing algorithm (SA). The results show that SE performs well and is comparable to SA for all case studies under consideration. Further, SE is capable to impose the correlation structure in a hard case such that the correlation structure is nearly positive definite. Hence, SE seems to be a good approach to use in any Monte Carlo simulations such as risk analysis model and computer simulations.

Keywords: Monte Carlo simulation · Stochastic evolutionary algorithm · Simulated annealing algorithm · Correlated multivariate random samples

1 Introduction

Monte Carlo simulation technique has been extensively used in various applications to explore the relationship between input variables and output response especially in complex phenomena. It has been accepted as a powerful computational tool for modeling and simulating complex systems in science, engineering, financial and risk analysis etc. [1]. Typically, when classical mathematical methods are not applicable for solving the solution, Monte Carlo technique could be used as an alternative approach to find out the solution [2]. In general, Monte Carlo simulation is performed by controlling the pre-specified conditions so that the simulated model is as closely related as possible to the actual model [3]. Although Monte Carlo simulations have been successfully used in many applications, their processes are time-consuming. Moreover, the cost of running complex codes such as finite element codes or fluid dynamics calculating codes is expensive. Hence, the efficient method to generate the random sample for Monte Carlo simulation is very critical. In order to generate multivariate random variables for Monte Carlo simulation technique, two main conditions have to be defined by experts. Firstly,

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_1

the probability distribution of each input variable must be specified and secondly, the correlation structure in a form of a correlation matrix between all input variables must be determined prior to generate the random samples. The key approach in this context is to generate multivariate random samples with probability distribution and correlation structure which are close to the real system. The performance of this approach can be validated by the moment of each input variable and the obtained correlation matrix which follows the pre-specified correlation matrix. Originally, the process of generating multivariate random samples starts with the sampling technique to select the sample from each marginal distribution, and then the mathematical transformation method is applied to find the best solution. If mathematical transformation is not applicable, the permutation technique, which aims to rearrange the sample points, can be used as an alternative method. This method is very flexible as it can be applied to generate any types of distribution and any forms of correlation coefficient [4].

In this paper, we apply a combinatorial optimization method of rearranging the sample points from initial multivariate random samples to impose the correlation coefficient, which is close to the specified correlation structure. Suppose there are d input variables and each variable has its own marginal distributions, which can be either same or different distribution. The aim of this method is to generate a d-variate sample of X of size n with specified marginal distributions and correlation structure between all input variables in a form of correlation matrix $c^*(d \times d)$. The marginal samples of X can be written as

$$x_1 = \begin{bmatrix} x_{11} \\ x_{21} \\ \cdots \\ x_{n1} \end{bmatrix}, x_2 = \begin{bmatrix} x_{12} \\ x_{22} \\ \cdots \\ x_{n2} \end{bmatrix}, \ldots, x_d = \begin{bmatrix} x_{1d} \\ x_{2d} \\ \cdots \\ x_{nd} \end{bmatrix}$$

Thus the matrix X can be expressed as $X = [x_1, x_2, \ldots, x_d]$.

In the past three decades, there have been a range of approaches to generate the multivariate random samples with pre-specified marginal distributions and a given correlation matrix [5, 6]. For instance, Iman and Conover [7] proposed the permutation algorithm for generating multivariate random samples under a rank correlation structure. The method was based on the calculation of van der Waerden scores, which aims to rearrange the position of the samples in order to achieve the target rank correlation matrix. Lurie and Goldberg [8] modified the algorithm by using the iterated optimization technique instead of the double integral, and also applied Cholesky factorization along with Gauss-Newton algorithm to control the correlation structure of the generated samples. This method has worked well in a case of multivariate normal distribution only. Charmpis and Panteli [4] proposed the combinatorial optimization technique namely Simulated annealing algorithm (SA) to generate multivariate random samples with any types of marginal distribution, i.e. continuous or discrete, parametric or non-parametric. Vorechovsk and Novak [9] also proposed a modified version of SA in the context of correlation control in small samples for Monte Carlo simulation. The results of their work indicated that SA performed well with high accuracy. A similar work can be found in Chakraborty [2], who proposed the PERMCORR algorithm to generate multivariate correlated samples with high accuracy and introduced a method

of checking the validity of the initial samples prior to run the algorithm. Ilich [10] proposed a matching algorithm for inducing Pearson correlations among random variables with any distribution functions. The performance of the proposed method is validated through various numerical examples and compared with @RISK commercial package. The features of a matching algorithm are simplicity, ease of implementation and the ability to handle any kinds of distribution functions.

As presented in Jin et al. [11], SE has proved to perform well in generating data points for computer simulation. Hence, we apply it for generating correlated random samples. The performance of SE will be compared with SA for various dimensions of samples. In the next section, we describe the process of multivariate random samples generation and the details of how we apply SE, and SA algorithm in this context. The results will be presented in Sect. 3 and the conclusion is delivered in Sect. 4 respectively.

2 Research Methods

The generation of correlated multivariate random samples consists of two distinct steps. The first step is the generation of d variate sample X of size n with specified marginal distributions and correlation matrix c^*. The second step aims to apply the optimization algorithm to rearrange the elements in each marginal sample to form a matrix c that approximates the target correlation matrix c^*.

In practice, after random samples from each marginal are generated the optimization algorithm is performed by rearranging any pairs of the elements in 2nd column such that the correlation coefficient between x_1 and x_2 is as close to the target correlation as possible. Then the elements in the 3rd column are swapped in the sense that the correlation between x_3 and x_1, the correlation between x_3 and x_2 are close to the pre-specified correlation coefficients and so on. The process continues until the termination criterion of the algorithm is met. Hence the rearrangement to reach the target correlation matrix can be described as follows:

- Rearrange x_2 such that $c_{12} \cong c^*_{12}$
- Rearrange x_3 such that $c_{13} \cong c^*_{13}, c_{23} \cong c^*_{23}$.
- …
- Rearrange x_j such that $c_{1j} \cong c^*_{1j}, \ldots c_{j-1,j} \cong c^*_{j-1,j}$.
- …
- Rearrange x_d such that $c_{1d} \cong c^*_{1d}, \ldots c_{d-1,d} \cong c^*_{d-1,d}$.

The distances between the target and the achieved correlation values have been extensively used as the objective function for the optimization algorithms. In this paper we use the root mean square error (RMSE) value proposed by Lurie and Goldberg [8] and it was also used in Charmpis and Panteli [4]. For the rearrangement of the elements in jth marginal sample x_j, the RMSE value can be calculated by (1)

$$RMSE(x_j) = \sqrt{\frac{2}{j(j-1)} \sum_{i=2}^{j} \sum_{k=1}^{i-1} (c_{ki} - c^*_{ki})^2} \tag{1}$$

The optimization algorithm tends to rearrange the elements in each marginal in a way that the RMSE is minimized. The final RMSE to be minimized is given by

$$RMSE(x_d) = \sqrt{\frac{2}{d(d-1)} \sum_{i=2}^{d} \sum_{k=1}^{i-1} (c_{ki} - c_{ki}^*)^2} \tag{2}$$

Other distance measures between the target and the achieved correlation coefficients can be used. More details of these measures can be seen in Lurie and Goldberg [8].

2.1 Simulated Annealing Algorithm (SA)

Simulated annealing algorithm has been widely used in various optimization problems. It was first proposed to use in thermodynamics to simulate the cooling of solid. Charmpis and Panteli [4] applied SA for generating correlated multivariate random samples. The steps of SA are given below.

Step 1: Set initial values

$I_{\max}$ (maximum number of perturbation to seek improvement)

t_0 (initial cooling temperature)

C_t (factor by which t_0 is reduced when no improvement in RMSE value)

n (Number of samples)

d (Number of input variables)

Step 2: Generate a random sample x_j. Let $x_{j,best} = x_j, t = t_0$.

Step 3: Set $I = 1, Counter = 0$

Step 4: Let $x_{j,try} = x_j$

Randomly select a column say j, of vector $x_{j,try}$ and exchange two randomly selected elements of column j, say $x_{aj} \Leftrightarrow x_{bj}$.

Step 5: Set $x_j = x_{j,try}, Counter = 1$

If $RMSE(x_j) - RMSE(x_{j,try}) \geq tol$ or with probability $e^{-\left[RMSE(x_{j,try}) - RMSE(x_j)\right]/t}$

Step 6: If $RMSE(x_{j,try}) < RMSE(x_{j,best})$, set $I = 1$ and $x_{j,best} = x_{j,try}$, else $I = I + 1$

Step 7: If $I < I_{\max}$, go to Step 4.

Step 8: If $Counter = 1$, set $t = t \times C_t$ and go to Step 3.

Step 9: Stop and report $x_{j,best}$.

The performance of SA is mainly based on the optimal setting of parameters. For a given dimension of random sample, the empirical study must be employed to obtain the optimal parameters prior to perform the optimization algorithm.

2.2 Stochastic Evolutionary Algorithm (SE)

Jin et al. [11] proposed an algorithm called stochastic evolutionary (SE) to construct an optimal design for computer simulation. The algorithm performs searching process in 2 steps, a local search called inner loop and updating a global best and fine tuning probability of accepting a worse design called outer loop. The steps of SE are presented below.

1. Initialize Th, J, M
2. Generate random sample X and set $X_{best} = X$
3. Set $X_{old\_best} = X_{best}$, $i = 0$, $n_{acpt} = 0$ and $n_{imp} = 0$
4. Randomly create J distinct X by element exchanging X at column(i mod d)
5. Select the best X from J distinct X and assign it to X_{try}
6. If $RMSE(X)$-RMSE(X_{try})$\geq tl$ or $RMSE(X_{try})$-RMSE(X)$\leq Th*random(0,1)$ then $X = X_{try}$, $n_{acpt} = n_{acpt} + 1$
   If $RMSE(X_{try})$< RMSE(X_{best}) then $X_{best} = X_{try}$, $n_{imp} = n_{imp}+1$
7. If $i < M$ then $i=i+1$, go to 4
   Else go to 8
8. If RMSE(X_{old_best})-RMSE(X_{best}) $\geq tl$ then $flag_{imp} = 1$
   Else $flag_{imp}=0$
9. If $flag_{imp}=1$ then
   If $n_{acpt}/M > \beta_1$ and $n_{imp}/M < n_{acpt}/M$ then
   $Th = Th*\alpha_1$
   Else if $n_{acpt}/M > \beta_1$ and $n_{imp}/M = n_{acpt}/M$ then
   $Th = Th$
   Else
   $Th = Th/\alpha_1$
   Else
   If $n_{acpt}/M \geq \beta_1$ and $n_{acpt}/M \leq \beta_2$ and step=0 then
   If step=0 or step=1 then $Th = Th/\alpha_3$
   Else If step=2 then $Th = Th*\alpha_2$
   Else If $n_{acpt}/M \leq \beta_1$ then
   $Th = Th/\alpha_3$
   step =1
   Else If $n_{acpt}/M \geq \beta_2$ then
   $Th = Th*\alpha_2$
   Step =2
10. If $RMSE(X_{old\_best})$- $RMSE(X_{best}) \geq tl$ then $X_{best}=X_{old\_best}$, $i_{out}=1$
    Else $i_{out}= i_{out}+1$
11. If $i_{out} < Max$ then Goto 3
    Else Report X_{best}

From steps of SE described above, the inner loop performs in 4–7, and repeats with the maximum loop (M). The *Xbest* and X are updated with the acceptance criteria. The outer loop controls the process by updating the value of temperature Th. Unlike SA, the process of updating the temperature (Th) is not fixed, but is controlled by the performance of searching in terms of the inner loop improvement in terms of number of improvement (n_{imp}) and number of acceptance (n_{acpt}). There are two processes of updating Th called improving process and exploration process. The process is described in 9, when *Xbest* get improved in the inner loop and better than the previous best design (X_{old_best}), the improving process is active otherwise exploration process is active. The parameter setting for SE can be found in Jin et al. [11].

3 Results

In this section we present the performance of SE in the generation of correlated multivariate random samples. A comparison of SE and SA is employed using three test examples used in Charmpis and Panteli [4] and Ilich [10]. The Pearson correlation coefficient is used to specify the correlation structure. All case studies are written in R codes and each case is repeated for 10 times.

3.1 Results from Bivariate Normal Distribution

In this case study we consider a bivariate normal distribution. Two random variables x_1 and x_2 have means 3.0 and 7.7 and standard deviations 0.04 and 0.08, respectively. The target correlation coefficient between x_1 and x_2 is 0.80. The results obtained from this case are given in Tables 1 and 2. The results in Table 1 indicate that the generation of bivariate normal distribution with either SA or SE has the marginal moments very close to the specified target values.

Table 1. Result from bivariate normal distribution

Marginal	Moment	Target	Achieved	
			SA	SE
x_1	Mean	3.000	3.00028	3.00029
	Var	0.0016	0.00162	0.00158
x_2	Mean	7.700	7.6989	7.7001
	Var	0.0064	0.0065	0.0065

Table 2 shows the RMSE values obtained from various sample sizes(n). It can be clearly seen that both of SA and SE provide 5 to 9 digits accuracy measured from the distance between the target and the achieved correlation coefficient as given in Eq. (2). It is also observed from the table that the accuracy is increased when the sample size increases. This observation is straightforward as the optimization algorithms are suitable for NP-hard problem. Hence they are able to search for the optimal solution in such a very large space. Once again, based on RMSE values, SE performs well and is

Table 2. RMSE values from bivariate normal distribution

Algorithms	Sample size			
	$n = 20$	$n = 50$	$n = 100$	$n = 500$
SA	2.8E−05	8.9E−05	4.7E−08	6.4E−09
SE	2.7E−05	8.7E−05	4.8E−08	6.2E−09

comparable to SA when sample size is small. When sample size is increased, SE performs slightly better than SA.

3.2 Results from 5-Variate Distribution

The second case study was used in Lurie and Goldberg [8] and Charmpis and Panteli [4]. It consists of two non-parametric (triangular with different moments) and three parametric marginal distributions (gamma, normal, and lognormal). The results of the target marginal moments obtained from SA and SE for this case study are given in Table 3.

Table 3. Results from 5-variate distribution

Marginal	Mean			Variance		
	Target	SA	SE	Target	SA	SE
Triangular 1	8.000	8.000	7.998	2.176	2.166	2.175
Triangular 2	−2.000	−2.000	−1.999	4.667	4.670	4.665
Gamma	2.000	1.998	2.001	1.000	1.020	1.014
Normal	−1.000	−1.011	−1.006	0.250	0.248	0.251
Lognormal	5.000	5.022	5.018	2.250	2.272	2.270

The results from Table 3 indicate that both of SA and SE provide very good result in the sense that the target and the achieved moment values are very close to each other. Next we consider the results on the correlation coefficient values obtained from SA and SE. The RMSE values obtained from 10 replications with various sample sizes are presented in Table 4. The results reveal that SE is comparable to SA when the sample size is small. When the sample sizes increase, SE is slightly better than SA. The accuracy obtained from each marginal distribution is also considered and is presented in Table 5.

Table 4. RMSE values from 5-variate distribution

Algorithms	Sample size		
	n = 50	n = 100	n = 500
SA	6.2E−04	2.1E−06	8.0E−08
SE	4.5E−04	2.4E−06	7.9E−08

Table 5. RMSE values from each marginal of the 5-variate random samples

Marginal	n = 100		n = 500	
	SA	SE	SA	SE
x_2	2.0E−07	2.5E−07	5.0E−09	5.2E−09
x_3	3.0E−06	3.0E−06	5.0E−08	4.5E−08
x_4	1.0E−05	1.1E−05	2.0E−07	1.0E−07
x_5	0.0021	0.0022	8.5E−05	7.8E−04

From the Table 5, the accuracy obtained from SA and SE is very high especially when the rearrangement is performed the first four marginal distributions. The accuracy is lower for the last marginal rearrangement as the rearrangement of the marginal x_{j+1} is harder than the rearrangement of marginal x_j since there are more restrictions to be considered.

3.3 Results from 8-Variate Distribution

This test problem consists of eight random variables with a mix of positive and negative correlation between any pair of input variable. Further, the marginal is also a mix of random input variable with various distributions that include both of discrete and continuous random variables. The target correlation structure can be found in Ilich [10]. The results, obtained from SA and SE are given in Tables 6 and 7 respectively.

Table 6. Results from 8-variate distribution

Marginal	Mean			Variance		
	Target	SA	SE	Target	SA	SE
Weibull	2.65	2.64	2.66	10.33	11.20	10.22
Extreme value	7.65	7.66	7.65	2.76	2.56	2.75
Lognormal	13.26	13.25	13.24	4.53	4.45	4.52
Binomial	19.0	18.90	19.01	0.46	0.42	0.44
Gamma	4.48	4.25	4.40	1.24	1.25	1.24
Poisson	8.26	8.20	8.24	8.26	8.28	8.25
Pearson V	7.45	7.50	7.42	60.15	60.22	60.18
Chisquare	10.0	9.50	9.80	20.0	19.85	19.90

The results presented in Table 6 suggest that the 8-variate distribution samples generated by SA and SE have marginal moments approximately equal to the target values. The accuracy results are given in Table 7.

Table 7. RMSE values from 8-variate distribution

Algorithms	Sample size		
	n = 50	n = 100	n = 500
SA	5.6E−02	3.4E−04	8.6E−06
SE	4.8E−02	3.6E−04	8.2E−06

Table 7 presents the RMSE values obtained from various sample sizes. It reveals that both of SA and SE provide 2 to 6 digits of accuracy. A similar result can be observed from this test problem. It is, however, the accuracy is slightly lower than that of 5-variate distribution. This finding is not unusual since this test problem is more complex in terms of mixed marginal and correlation structure. When RMSE values are considered, it can be concluded that SE is slightly better than SA as it provides lower RMSE values than SA.

4 Conclusions

This paper presents the application of optimization algorithm for generating correlated multivariate random samples for Monte Carlo simulation. SE algorithm is adopted in the field of Monte Carlo simulation and its performance is compared with the existing method called SA algorithm. A range of test problems consist of various statistical distribution functions and different correlation structures are stimulated using R program. The results presented in Sect. 3 indicate that SE performs well and is comparable to SA while its performance is superior over SA when a sample size is large. Hence SE is recommended to use as an alternative to SA, especially when marginal distribution and the correlation are more complex. The results from this study also confirm the capability of optimization algorithm in generating multivariate random samples with mixed marginal distribution. In order to extend the conclusions, other optimization algorithms like clever algorithm should be further studied. Furthermore, other dimensions of samples should also be explored.

References

1. Hass, H.C.: On modeling correlated random variables in risk assessment. Risk Anal. **19**(6), 1205–1214 (1999)
2. Chakraborty, A.: Generating multivariate correlated samples. Comput. Stat. **21**(2), 103–109 (2006)
3. Vose, D.: Quantitative Risk Analysis: A guide to Monte Carlo Simulation Modeling. Wiley, New York (2007)
4. Charmpis, D., Panteli, A.: A heuristic approach for the generation of multivariate random samples with specified marginal distribution and correlation matrix. Comput. Stat. **19**(2), 283–300 (2004)
5. Dukic, V.M., Maric, N.: Minimum correlation in construction of multivariate distributions. Phys. Rev. E **87**(3), 032114 (2013)
6. Reinartz, W.J., Echambadi, R., Chin, W.W.: Generating non-normal data for simulation of structural equation models using Mattson's method. Multivar. Behav. Res. **37**(2), 227–244 (2002)
7. Iman, L.R., Conover, W.J.: A distribution-free approach to inducing rank correlation among input variables. Commun. Stat. Part B Simul. Comput. **11**(3), 311–334 (1982)
8. Lurie, P., Goldberg, M.: An approximate method for sampling correlated random variables from partially specified distributions. Manag. Sci. **44**(2), 203–218 (1998)

9. Vorechovsky, M., Novak, D.: Correlation control in small-sample Monte Carlo type simulations I: a simulated annealing approach. Probab. Eng. Mech. **24**, 452–462 (2009)
10. Ilich, N.: A matching algorithm for generation of statistically dependent random variables with arbitrary marginal. Eur. J. Oper. Res. **192**, 468–478 (2009)
11. Jin, R., Chen, W., Sudjianto, A.: An efficient algorithm for constructing optimal design of computer experiments. J. Stat. Plan. Inference **134**, 268–287 (2005)

Impurity Measurement in Selecting Decision Node Tree that Tolerate Noisy Cases

Benjawan Srisura(✉)

Information Technology Laboratory, Vincent Mary School of Science and Technology, Assumption University, Bangkok, Thailand
benjawan@scitech.au.edu

Abstract. In a recent years, recommending an appropriate attribute of binary decision tree under unusual circumstances – such as training or testing with noisy attribute, has become more challenge in researching. Since, most of traditional impurity measurements have never been tested how much they can tolerate with encountered noisy cases. Consequently, this paper studies and proposes an impurity measurement which can be used to evaluate the goodness of binary decision tree node split under noisy situation, accurately. In order to make sure that the accuracy of decision tree classification by using the proposed measurement has been yet preserved, setting up an experiment to compare with the traditional impurity measures was conducted. And the result shows that accuracy of the proposed measurement in classifying a class under noisy case is acceptable.

Keywords: Decision tree · Classification · Splitting node · Impurity measure

1 Introduction

One of well-known classification technique which has discussed in both data mining field and machine learning field is the decision tree. Especially in data mining, classifying using decision tree has been accepted as a popular technique used in the real world problems [1]. It classifies an encountered case based on its attributes. Classifying which class at the leaf level of decision tree that it is belonging to requires to traverse attribute node until arriving to an appropriate leaf node.

Decision binary tree is constructed from the collection of the appropriate attributes which are represented in term of decision root node and non-leaf nodes. At the bottom level of the tree, all leaf nodes are assigned to a class label of those concerned attributes. Therefore, all attributes of training's cases will be expressed in proper way to justify a suitable class, accurately. It has been found to be popular due to its practical use [2].

In part of statistical theory in classification problem, expressing an attribute test condition can be done based on the attribute type – including categorical (nominal and ordinal scale) and numeric (interval and ratio scale). Morgan and Sonquist [3] proposed the automatic interaction detection (AID method) which generated the regression tree to predict class by using ANOVA model to analyze the variance of a quantitative attribute. Ten year later, a modification of AID - CHAID classification tree [4], was

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_2

implemented by Kass to classify a categorical class from the categorical attributes by using a chi-square test statistic. This algorithm can be run in optimized time however its classification can only be done on the categorical attribute.

Another point of view under data mining theory, the first decision tree classification has been initiated by Quinlan [5]. The splitting criteria has been recommended in term of impurity measurement. The purity is evaluated in term of class distribution ratio before and after splitting. In the literature, traditional well-known decision tree algorithms which consider the impurity degree of the decision tree can be summarized into three main groups [6].

The first group of impurity measurement is entropy measure [6]. This measure will recommend an appropriate decision tree node split based on the logarithmic ratio consideration. The best-known decision tree algorithms which use this measure to recommend the best node split are ID3 [5] and C4.5 [7].

Next measurement whose name is gini index [8] which uses the generalization of the binomial variance. It was initiate to improve the performance of entropy measure and can support both binary and n-nary decision tree node split and it has been used in advance algorithm – such as CART [9].

Finally, the third impurity measure group is classification error or misclassification [5] which evaluates impurity degree in term of a liner curve. The computation time of these measures are minimized by considering the maximum or minimum distribution ratio. An example of algorithm which use this measure is THAID [10].

There are several classification research works that deal with getting rid of noises [11, 12]. Robust classification solutions were proposed in the various perspectives under each theoretical classification techniques – including the eager classifier (e.g., decision tree) and the lazy classifier (e.g., kNN classifier). In the part of lazy classifier, considering the correlations between attributes [13] was proposed to build a robust classifier. For eager classifier, there are several pivot improvement of decision trees that tolerate to the noisy big data, such as VFDT [14] and iOVFDT [15] by optimizing node split for maintaining the decision tree high to be balance state. Optimizing node split has been done by selecting the heuristic value - which can be entropy or gini index, that can built a high balance decision tree.

Generally, splitting an attribute condition in decision binary tree by considering the impurity measure properly works under normal environment. Unusual circumstance such as noisy attributes can occur anytime and uncontrollable. Therefore there are many researches [16–19] tried to examine whether the traditional impurity measures are sensitive with the presence of noise or not.

This paper illustrated how each impurity will recommend under the presence of noise. Based on the traditional way of considering impurity degree, this paper proposes an outlier statistical test to be another pivot step to ensure that the recommended attribute will not be noise attribute. However, the accuracy of classifying a class will be preserved, carefully.

2 Principle of Selecting the Best Node Split in the Literature

Generally, the impurity measurement will recommend the best attribute node split by examining the distribution – both before and after splitting. Then it will return a result in term of the degree of impurity at each child node as shown in Fig. 1.

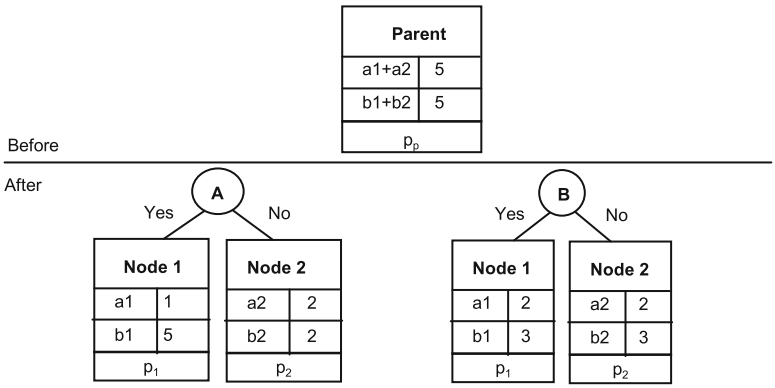


Fig. 1. Binary attribute node and case distribution associated to two-class classification

The impurity measure recommends the best attribute node split whose impurity degree is minimum. The small degree of impurity, the more skew distribution. Refer to the Fig. 1, the parent node has a uniform distribution (a1 + a2:b1 + b2 = 5:5, p = 5/10 = 0.5).

Given: p(i|t) is the ratio between the total cases of class i under child node. Since this situation is binary attribute node split, its maximum value is 2– including left child node (Node 1) and right child node (Node 2).

And c is the total concerned classes which is 2 for two-class classification problem. The degree of impurity measure of child node – I(Child), will be computed based on specific measures as follows:

$$\text{Entropy} = -\sum_{t=0}^{n-1}\sum_{i=0}^{c-1} p(i|t)\log_2 p(i|t) \tag{1}$$

$$\text{Gini} = 1 - \sum_{t=0}^{n-1}\sum_{i=0}^{c-1} [p(i|t)]^2 \tag{2}$$

$$\text{Classification Error} = 1 - max_{t=0}^{1} p(i|t) \tag{3}$$

$$\text{Miscalssification} = min_{t=0}^{1} p(i|t) \tag{4}$$

Those impurity measures – entropy, gini, and classification error (or misclassification) measure, preferably recommend minimum impurity degree value (the highest skew or choice A and B in Fig. 2) rather than maximum value (uniform distribution or choice C in Fig. 2) because they try to extract the impurity case as much as possible.

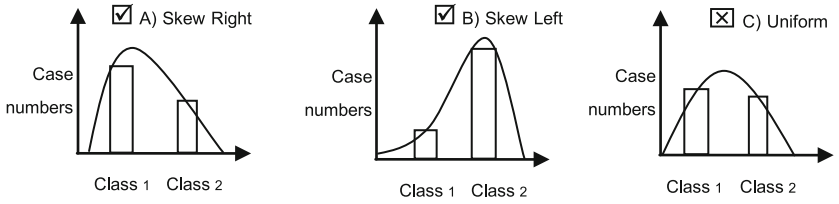


Fig. 2. Two-class distribution pattern

Currently, a noise can easily occur in constructing, training and testing of a decision tree. Selecting an appropriate attribute by considering existing measures may lead to the wrong attribute because of noise. In a recent year, we found that there is a measurement that concerning noisy class label – named Twoing rule [20]. It proposes a robustness of decision tree which against the class noise and all attributes that contain noise will not be recommended. However, the pure case - whose distribution is similar to the noise case, may be destroyed as well.

Refer to two-class distribution pattern in Fig. 1, the total number of cases found in the first class are a1 in node 1, a2 in node 2. And b1 and b2 represent total cases of class 2 found in node 1 and 2, consequently.

$$\text{Twoing rule} = \frac{(a1+b1)(a2+b2)}{4}\left(\left|\frac{a1}{a1+b1}-\frac{a2}{a2+b2}\right|+\left|\frac{b1}{a1+b1}-\frac{b2}{a2+b2}\right|\right) \quad (5)$$

3 Proposed Measurement

Since extracting the purer impurity degree can be viewed into two sides – pure case or noise case. If it is the noisy case, this attribute will be recommended to construct a poor decision tree. Detecting noise before recommending attribute node is therefore proposed to carefully be considered before recommending an appropriate attribute.

There are two main mechanisms involved in recommending the best attribute node as shown in Fig. 3.

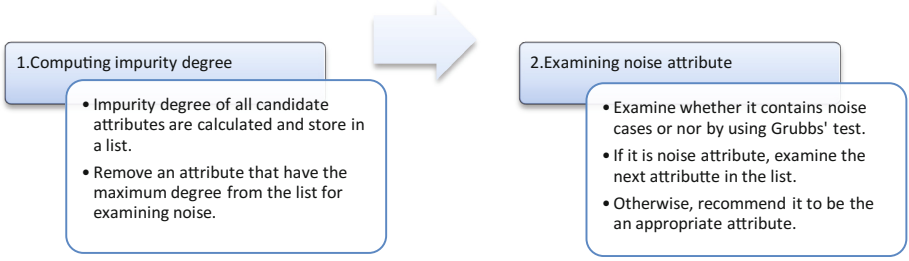


Fig. 3. Two mechanisms of the proposed measurement

3.1 Computing Impurity Degree

To determine which attribute node will be suggested, the impurity degree of all candidate attributes are calculated and summarized in the table below. The attribute that have maximum value will be selected to examine in next step.

Table 1. Impurity degrees of four impurity measures

Attribute	Node1		Node2		Entropy	Gini	Classification	Twoing
	C0	C1	C0	C1				
A	**4**	**7**	**6**	**3**	**(0.06)**	**(0.05)**	**(0.15)**	**0.19**
B	7	5	7	1	0.19	0.12	0.20	0.15
C	7	7	5	1	0.10	0.06	0.10	0.17
D	**6**	**4**	**5**	**5**	**0.01**	**0.01**	**0.05**	**(0.20)**
E	8	7	5	0	–	0.13	0.15	0.10

_is the first recommended attribute and () represent the final recommended attribute

Considering an example in Table 1 shown above. There are 5 candidate attributes (A, B, C, D and E). All un-bold attribute - including the attribute B, C and D are the noise attributes. In this study, all four measures – Entropy, Gini, Classification (or Misclassification) error and Twoing measure, were used to test together.

As we knows that entropy measure cannot evaluated any attributes – containing no members such as attribute E. Unfortunately, the noise attribute B could not be detected, it was firstly recommended as the best appropriate attribute because its degree is maximum. Considering A3 in Fig. 4, a class of noise attribute contains no data, cannot be detected by entropy measure.

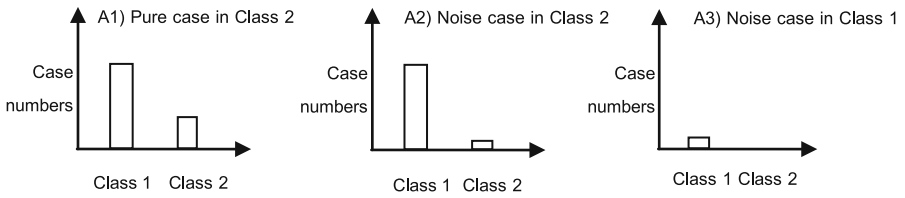


Fig. 4. Class distribution including noise

Gini and Classification error (or misclassification) have not yet tolerate with a noise attribute because they prefer to recommend the attribute whose degree is highest (or largest skew). It is high risk to recommend the noise attributes which are the attribute E (by Gini) and B (by Classification error), consequently. These measures are recommended by this study rather than entropy because the noise attribute – containing no data, can be evaluated. Finally, as expected, Twoing rule does not recommend any noise attributes. And attribute D is recommended to be the best attribute.

3.2 Examining Noise Attribute

The class distribution of an attribute node - including either pure cases or noise cases, is naturally skew. In order to preserve the accuracy of the classification, this paper requires to proof whether those cases are pure cases (A1) or noise cases (A2 and A3) by using Grubbs' test [21] – a statistical testing used to determine a single noise in an univariate data set, before recommending the best attribute.

In binary decision tree, an attribute node is always separated in two nodes or data sets. Therefore, the hypothesis is set up as follows:

H_0: There is no noise case in child nodes.

H_a: There is exactly one noise case in a child node.

The Grubbs' test statistic is defined as:

$$G = \frac{\max|Yi - \bar{Y}|}{s} \tag{6}$$

$$G_{crit} = \frac{(n-1)T_{crit}}{\sqrt{n(n-1+T_{crit}^2}} \tag{7}$$

Given: T_{crit} is the critical value of the T-test distribution which is $T_{(n-1),\ \propto/2}$.

Considering at the significance level $\propto$, for two-sided test, the hypothesis H_0 is rejected if:

$$G > G_{crit} \tag{8}$$

Finally, the attribute B, C and E were justified to be noisy attributes and removed from the recommendation. Then, the remaining attributes will be considered, the first maximum impurity degree attribute will be recommended which is the attribute A.

4 Experimental Design and Result

In order to preserve the accuracy of decision tree prediction, the accuracy matrix will be used to test the proposed measurement – named PM, as follows (Tables 2, 3, 4).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

Table 2. Classification performance matrix

True condition	Prediction	
	Positive	Negative
True	True positive (TP)	True negative (TN)
False	False positive (FP)	False negative (FN)

Since we require to study the performance of predicting a class with some noisy attributes, a number of the noise attributes – around ten percent of dataset size, are required to generate randomly. There are 10 datasets which require to classify two-class problem from UCI repository, including:

Table 3. Selected dataset from UCI repository for two-class classification

No.	Dataset name	No. of attributes	No.	Dataset name	No. of attributes
1	Haberman	3	6	Echocardiogram	9
2	Liver	6	7	Credit	14
3	Pima Indian Diabetes	8	8	Voting	16
4	Breast Cancer	9	9	Mushroom	21
5	Wisconsin BC	9	10	Ionosphere	34

Since this paper tries to prove how much the proposed step can improve the existing impurity measure that used in several well-known algorithm to tolerate to noise, the entire comparative sets studied in this paper will be summarized as follows.

Table 4. List of comparative sets

No.	Impurity measure	Decision tree algorithm
1	Entropy	ID3, C4.5
2	Gini	CART
3	Classification	THAID
4	Twoing rule	Twoing

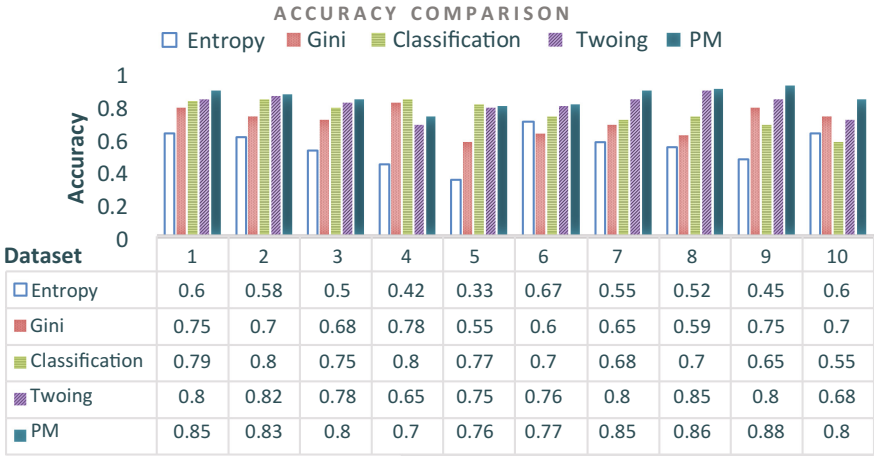


Fig. 5. Accuracy comparison

As a result - shown in Fig. 5, it shows that the accuracy of PM is acceptable although the data set contains some noisy attributes. When we compare with the entropy measure, PM can improve the accuracy of entropy around 53.80%. Entropy cannot tolerate with the noise data especially when some class contain no data. In the part of next measures – Gini and Classification error which no data class cannot affect to their consideration, we found that PM can improve their performance around 18.94 and 15.88. Finally, the latest measure – Twoing rule which most robust to the noise attribute rather previous measures, its accuracy was also acceptable. The accuracy of the proposed model has a little higher – approximately be 6.24%.

5 Conclusion and Recommendation

In summary, this paper proposed a measurement in recommending a binary decision tree attribute under noisy circumstance. Determining noise attribute with statistical testing is suggested to implement and add to be another step before justify the best nodes split when a noisy case is highly concerned.

The experiment shows that approaching this measure can improve the accuracy performance of decision binary tree classification, accurately. However, there are some studies suggested to cumulative research.

Firstly, since this paper proposes a process of filtering noisy attribute after the traditional impurity measure judging, a step refinement or new impurity measure generation which can recommend the best attribute from the noise attributes is recommended to be further present and compare to the traditional well-known decision tree algorithms – such as ID3, C4.5, CART, later.

Finally, in order to confirm the effectiveness of the proposed measurement in the real situation, we require to apply to an industry whose noise data is highly concerned.

References

1. Michael, J.A., Berry, G.S., Linoff, G.S.: Mastering Data Mining. Wiley, New York (2000)
2. Pang-Ning, T., Michael, S., Vipin, K.: Introduction to Data Mining. Addison Wesley, Boston (2000)
3. Morgan, J.N., Sonquist, J.A.: Problems in the analysis of survey data and a proposal. J. Am. Stat. Assoc. **58**(302), 415–434 (1963)
4. Kass, G.V.: An exploratory technique for investigation large quantities of categorical data. Appl. Stat. **29**, 119–127 (1980)
5. Quinlan, J.R.: Introduction of decision tree. J. Mach. Learn. **1**(1), 81–106 (1986)
6. Elomaa, T., Rousu, J.: General and efficient multi splitting of numerical attributes. J. Mach. Learn. **36**(3), 201–244 (1999)
7. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufman Publishers, Burlington (1993)
8. Chandra, B., Kothari, R., Paul, P.: A new node splitting measure for decision tree construction. J. Pattern Recognit. **43**(8), 2725–2731 (2010)

9. Singdong, W., Vipin, K.: The Top Ten Algorithm in Data Mining. CRC Press, Boca Raton (1984)
10. Morgan, J.N.: THAID: A Sequential Analysis Program for Analysis of Nominal Scale Dependent Variables. Survey Research Center, Institute for Social Research, University of Michigan (1973)
11. Luis, P.F.G., Andre, C.P.L.F., Ana, C.L.: Effect of label noise in the complexity of classification problems. Neurocomputing **160**, 108–119 (2015)
12. Jakramate, B.: A generalized label noise model for classification in the presence of annotation errors. Neurocomputing **192**, 61–71 (2016)
13. Alexandros, N., Apostos, N., Yannis, M.: Robust classification based on correlations between attributes. In: Data Warehousing and Mining: Concepts, Methodologies, Tools and Applications, vol. 3, pp. 3212–3221. IGI Global (2008)
14. Yang, H., Fong, S.: Moderated VFDT in stream mining using adaptive tie threshold and incremental pruning. In: 13th International Conference on Data Warehousing and Knowledge Discovery. LNCS, pp. 471–483, Springer, Berlin (2011)
15. Hang, Y., Simon, F.: Incrementally optimized decision tree for noisy big data. In: 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Application. pp. 36–44. ACM, New York (2012)
16. Gama, J., Rocha, R., Medas, P.: Accurate decision trees for mining high-speed data streams. In: 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03), ACM, pp. 523–528, New York (2003)
17. Carla, E.B., Mark, A.F.: Identifying mislabeled training data. J. Artif. Intell. Res. **11**, 131–167 (1999)
18. Frenay, B., Michel, V.: Classification in the presence of label noise: a survey. IEEE Trans. Neural Netw. Learn. Syst. **25**, 845–869 (2014)
19. Aritra, G., Manwani, N., Sastry, P.S.: Making risk minimization tolerant to label noise. Neurocomputing **160**, 93–107 (2015)
20. Aritra, G., Manwani, N., Sastry, P.S.: On the Robustness of Decision Tree Learning Under Label Noise. Math Pubs Publication (2016)
21. Frank, E.G., Beck, G.: Extension of sample size and percentage points for significant tests of outlier observation. Technometrics **14**(4), 847–854 (1972)

OMFO: A New Opposition-Based Moth-Flame Optimization Algorithm for Solving Unconstrained Optimization Problems

Wirote Apinantanakon[1(✉)] and Khamron Sunat[2]

[1] Computer Science, Faculty of Arts and Science, Rajabhat Chaiyaphum University, Na Fai, Thailand
wirotta@gmail.com
[2] Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen, Thailand
khamron_sunat@yahoo.com

Abstract. The Moth-Flame Optimization (MFO) algorithm is a nature-inspired search algorithm that has delivered good performance and efficiency in solving various optimization problems. In order to avoid local optimum and increase global exploration, each moth of MFO updates its position with respect to a specific MFO operation. However, MFO tends to suffer from a slow convergence speed and produces a low quality solution. This paper presents a new opposition-based scheme and embeds it into the MFO algorithm. The proposed algorithm is called OMFO. The experiments were conducted on a set of commonly used benchmark functions for performance evaluation. The proposed OMFO was compared with the original MFO and four other well-known algorithms, namely, PSO, DE, GSA and GWO. The results clearly showed that OMFO outperformed MFO and the four other algorithms used.

Keywords: Moth-Flame Optimization · Opposition-based learning · Nature-inspired algorithm · Unconstrained optimization problems

1 Introduction

Natural behaviors have inspired researchers to produce intelligent systems called nature-inspired algorithm such as Particle Swarm Optimization (PSO) [1], Artificial Bee Colony (ABC) [2], Ant Colony Optimization (ACO) [3] and Fish School Search (FSS) [4]. The processes of these algorithms consist of important steps like determining population, initializing random position, updating position and finding the optimal solution. Approaches used by these well-known algorithms have led to the successful optimization of various complex optimization problems [5–8]. These approaches have also been extensively applied to solve real world problems in the fields of science, engineering, image processing, financing and networking problems.

MFO is a nature-inspired algorithm proposed by Mirjalili in 2015 [9]. It could successfully be employed intelligent system by mimicking moths. MFO outperformed several well-known algorithms in finding solution accurately, which were presented in the original article. The navigation method used by moths in nature is called trans-verse

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_3

orientation and it is the key to MFO's success. The "Flame" variable in the transverse orientation process is an important parameter that MFO uses to update the new position of the population. However, the reducing of flame cause slow convergence speeds, which affects to the quality of the final solution. To overcome this drawback, our paper presents a new opposition-based MFO algorithm called OMFO that adds a new moth generating scheme. This function is a new form of opposition-based learning (OBL) and it is placed at the updating of the moth swarming step. OMFO is more powerful than MFO, although only one moth will be generated by OBL.

2 Related Works

2.1 Moth-Flame Optimization Algorithm

The MFO algorithm is designed to solve optimization problems by mimicking the navigation method used by moths at night. The procedure of the algorithm is based on the initial number of moths, the output of fitness values were obtained from the corresponding moths which help find a better solution and the movement of moths around the search space during the updating step. The representation of MFO structure can be described as follows.

Firstly, a matrix representing the set of n moth's populations is:

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & \cdots & m_{1,d} \\ m_{2,1} & m_{2,1} & \cdots & \cdots & m_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{n,1} & m_{n,2} & \ldots & \ldots & m_{n,d} \end{bmatrix}, \tag{1}$$

where M is the matrix of moths, n is the number of moths and d is the number of variables (or dimensions). Moreover, MFO randomly initializes the position of moths with Eq. (2), as follows:

$$M(i,j) = (ub(i) - lb(i)) \times rand() + lb(i), \tag{2}$$

where *ub* is the upper bound of *i-th* moth, *lb* is the lower bound of *i-th* moth and *rand* () is a uniform random number in [0, 1]. The MFO also provides the matrix *FM* for keeping the fitness value of all moths, which are presented as follows:

$$FM = \begin{bmatrix} fm_1 \\ fm_2 \\ \vdots \\ fm_n \end{bmatrix}, \tag{3}$$

where *FM* is the matrix of fitness value of each moth and n is the number of moths. The fitness value depends on the values of M and each row of *FM* corresponds to the moth's fitness.

Secondly, there is a key component, which is called flame. Flame is a matrix with a similar structure to the matrix of moths, M, as follows:

$$FL = \begin{bmatrix} fl_{1,1} & fl_{1,2} & \cdots & \cdots & fl_{1,d} \\ fl_{2,1} & fl_{2,1} & \cdots & \cdots & fl_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ fl_{n,1} & fl_{n,2} & \cdots & \cdots & fl_{n,d} \end{bmatrix}, \tag{4}$$

where n is the number of flames and d is the number of variables (or dimensions).

For the flames, MFO algorithm designates the flames matrix just like an array for storing the corresponding fitness values as follows:

$$FF = \begin{bmatrix} ff_1 \\ ff_2 \\ \vdots \\ ff_n \end{bmatrix}, \tag{5}$$

where FF is the fitness of moths and n is the number of moths. The difference between moths (M) and flames (FL) is that moths are populations that move within the search space in iterations to find the best solution, while flames would sort the positions that are the best corresponding position of moths obtained so far. Flames are also used for updating positions of the moths. The processes within the MFO algorithm can be broken down into three important functions, which are presented as follows:

$$MFO = (I, P, T), \tag{6}$$

where I is the function model that MFO used to generate random populations of moths and corresponding fitness values. The model of function I can be described as:

$$I : \theta \rightarrow \{M, FM\} \tag{7}$$

P is the main function that moves the moths around the search space to find a better solution.

$$P : M \rightarrow M \tag{8}$$

T is the function that is used in the stop condition. The function returns true if the termination criterion is satisfied, otherwise it returns false.

$$T : M \rightarrow \{\text{true}, \text{false}\} \tag{9}$$

Note that, in updating a position, MFO used a logarithmic spiral as the main mechanism to update the positions of the moths. A logarithmic spiral is defined in the MFO algorithm as follows:

$$S(M_i, FM_j) = D_i.e^{bt}.\cos(2\pi t) + FL_j, \tag{10}$$

where D_i indicates the distance between *i-th* moth and *j-th* flame, b is a constant value for forming the shape of the logarithmic spiral, and t is the random number in $(-1, 1)$. The resulting value of D is calculated as follows:

$$D_i = |FL_j - M_i|, \tag{11}$$

where D_i is the distance of *i-th* moth for *j-th* flame, M_i indicates the *i-th* moth and FL_j indicates the *j-th* flame, $|.|$ is the metric function, e.g. the Euclidean distance. In order to balance the exploration and exploitation in the updating process, MFO resolves this step with an adaptive mechanism, which is presented in Eq. (12) as follows:

$$flame\_no = round\big(N_f - t * \big((N_f - 1)/T\big)\big), \tag{12}$$

where t is the current number of iteration, N_f is the maximum number of flames, and T indicates the maximum number of iterations. A more detailed explanation of the MFO can be read in [9].

2.2 Opposition-Based Learning

OBL [10] was originally proposed by Tizhoosh in 2005. The main procedure of this scheme was to find a better candidate solution, the simultaneous consideration of an estimate and its corresponding opposite estimate, which was the closer to the global optimum. There are several OBL schemes and in a very short period of time it has been utilized in different areas of computing. The general scheme of OBL can describe as follows:

Definition 1. Let $x \in [a, b]$ be a real number. The opposite number $\breve{x}$ is represented by

$$\breve{x} = a + b - x. \tag{13}$$

Similarly, in the field of complex optimization problems which were contained with the high dimensions. The opposite point in d-dimensional is defined by

Definition 2. *Let* $M = (x_1, x_2, \ldots, x_d)$ be a position in d-dimensional space, where $x_1, x_2, \ldots, x_d \in \mathcal{R}$ and $x_i \in [a_i, b_i] \forall_i \in \{1, 2, \ldots, d\}$. The opposite point $\breve{M} = (\breve{x}_1, \breve{x}_2, \ldots, \breve{x}_D)$ is completely defined by its components

$$\breve{x} = a_i + b_i - x_i. \tag{14}$$

In the proposed OMFO, the opposite point of the moth is generated by OBL in order to enhance MFO by

$$\begin{aligned} m_i &= (m_1, m_2, \ldots, m_d) \\ \breve{m}_i &= a_i + b_i - m_i, \end{aligned} \tag{15}$$

where m_i is the best position of *i-th* moth (*best_moth*).

The proposed OMFO with a new OBL and details are presented in the next section.

3 The Proposed OMFO Optimization Algorithm

3.1 The Disadvantage of MFO

In MFO, flame is the key component that is used to update the position of moths around the search space. Flames are the sorting positions that correspond to the best position of the moths. The position of flames is generated from the boundary of moths that were initialized with the function in Eq. (2). Another purpose of "flames" is to prevent the MFO algorithm from getting quickly trapped in local optima. The design of MFO, which is based on Eq. (12) is to gradually decrement the number of flames to allow for the balanced exploration and exploitation of the search space. However, this is one of the disadvantages of MFO algorithm because the MFO processes reduce the number of flames that depend on the interval of iterations without other parameters or any function. As can be seen in Eq. (12), the boundary, which was obtained from any flames, would be large in the early stages and its boundary would continue to decrease slowly to a smaller size when the iterations increase. MFO deals with only a minimized solution problem. The minimized solution could not be found in the early stage. Accordingly, the MFO algorithm obtains a low accuracy solution because of the limited time for exploitations and by the termination criteria that limit the maximum round of iterations.

3.2 Opposition-Based MFO

To overcome the disadvantage of MFO, this paper presents the supplementary function H, which has been added to the original function of *MFO* (I, P, T). It is named *OMFO* (I, P, T, H). The advantages of the H function that it accelerates the convergence speed of the MFO processes and obtains a highly accurate solution. The OMFO function is presented as follows:

$$OMFO = (I, P, T, H), \tag{16}$$

where I, P and T are the same as they are in the original *MFO*, H is the supplementary function to accelerate the convergence speed of the P function in Eq. (8). The details of the H function are presented as follows:

r = a random integer in the range of [2, N], N is the number of moths

s = Iteration/Max_iterations, £ = max position of moths, ¥ = min position of moths

$A = \begin{cases} £, s > rand() \\ ub, otherwise \end{cases}$, ub is the upper bound

$B = \begin{cases} ¥, & s > rand() \\ lb, & otherwise \end{cases}$, lb is the lower bound

C = best position of flame sorted from moths which are obtained from the updated step in any iterations.

The opposition-based moth-generating scheme with the H function is

$$H = (B + A) - C \times rand() \tag{17}$$

$$M_r = H. \tag{18}$$

To the best knowledge of the authors, Eq. (17) is a new form of opposition-based point generation.

Considering the updating step of MFO, the moths updated new positions with respect to different location corresponding to their flames by using P function. To keep the updating pattern of the original MFO, OMFO only uses one moth among [2, N] will be replaced by Eq. (17). The first moth is not disturbed since it is the best position.

The steps of OMFO are as follow:

```
Initialize parameters, Max_iterations, sizepop
  Generate the random position of moths and flames
  While (Iteration<=Max_iterations)
  FM = FitnessFunction(M);
  Update MFO's flame number by using (12)
  if Iteration == 1
    FL = sort(M);
    FF = sort(FM);
  else
    FL = sort(M_{t-1},M_t); t is the current iteration
    FF = sort(M_{t-1},M_t);
  end
  for i=1:sizepop
    for j=1:d
      Calculate D using (11)
      Update M(i,j) using(10)
    end
  end
  Enhance MFO with H function by using (17)
    M_i = M_r;(18)
  End
```

Terminate and report the best global moth formation (*best_moth*).

4 Implementation Details and Results

The aim of the proposed model of OMFO is to accelerate the convergence speed of the MFO processes and to obtain a good accuracy solution. To show the abilities of the OMFO, 12 benchmark functions taken from [9, 11], were selected as testing functions. The functions were divided into two cases: $f1$–$f7$ are the uni-modal, where each function has only one global optima, $f8$–$f12$ are the multi-modal where each function has numerous numbers of local optima. Table 1 shows the results and details of 12 functions.

It should be noted that the percentage of success runs was used to evaluate the performances of each algorithm. The maximum iteration of a run is 1000 and the final searching quality is set at $10e-4$. If the difference of the best moth fitness value of the run and the optimum fitness value is less than $10e-4$ while $f8$ is less than -41800, that run is called a success run. The algorithm with a higher SR is a better algorithm. The percentage of success run is computed as follows:

$$SR = \frac{number\ of\ successful\ runs}{total\ number\ of\ runs} \times 100 \tag{19}$$

4.1 Experimental Setup

Two well-known algorithms, PSO [1], DE [12], three of the more recent nature-inspired algorithms, GSA [13], GWO [14] and the original MFO [9] were chosen to be competitors. The population size was 30, max-iterations was set at 1000, the dimensional of variables was 100 and each problem is run at 30 independent replications. The other parameters are the default values depending on the algorithm.

4.2 Results and Discussion

The proposed OMFO algorithm was established to improve the solution finding ability of the original MFO that had a faster convergence speed and that had greater accuracy. The results of the performance of accuracy that were obtained from each algorithm in this paper is presented with the quality of standard deviation (*Std*) value, average (*Ave*) value and success rate (*SR*) value. The optimum value of $f1$–$f7$ and $f9$–$f12$ are 0 and $f8$ is -418.9829×100. OMFO achieved a minimum optimum value *Std, Ave* and high *SR* value that outperforms the original MFO algorithm in every function (as can be seen in boldface in Table 1). Moreover, the results of the *Std* and *Ave* of OMFO also outperform the other four well-known algorithms. For example, function $f1$, the optimum values were (*Std*/*Ave*) = 3.1E+04/1.5E+04; (*Std*/*Ave*) = 6.7E+00/2.3E+01, (*Std*/*Ave*) = 2.2E −03/6.0E−04; (*Std*/*Ave*) = 6.0E+02/3.1E+02 and (*Std*/*Ave*) = 3.8E−29/6.5E−29 for MFO, PSO, DE, GSA and GWO respectively. However, function $f1$ and the optimum value of OMFO (*Std*/*Ave*) = 0/0. It can be clearly seen in Table 1 that the proposed opposition-based moth-generating scheme could improve the MFO in obtaining highly accurate results.

Table 1. Comparison of PSO, DE, GSA, GWO, MFO and OMFO on 100-dimensional benchmark functions.

Name	Criteria	PSO	DE	GSA	GWO	MFO	OMFO
*f*1	*Std*	6.7E+00	2.2E−03	6.0E+02	3.8E−29	3.1E+04	**0**
	Ave	2.3E+01	6.0E−04	3.1E+02	6.5E−29	1.5E+04	**0**
	SR	0%	100%	0%	100%	0%	**100%**
*f*2	*Std*	1.5E+01	7.3E−01	6.0E+00	3.1E−18	1.6E+02	**2.8E−153**
	Ave	5.1E+01	1.3E−01	2.2E+00	6.6E−18	3.5E+01	**1.3E−152**
	SR	0%	100%	0%	100%	0%	**100%**
*f*3	*Std*	4.2E+03	5.2E+03	9.0E+03	1.5E+00	1.9E+05	**0**
	Ave	1.3E+04	1.9E+03	1.9E+03	9.3E+01	5.3E+04	**0**
	SR	0%	0%	0%	0%	0%	**100%**
*f*4	*Std*	1.3E+01	3.7E+01	1.5E+01	1.0E−02	9.2E+01	**9.5E−149**
	Ave	1.1E+01	5.7E+01	1.6E+00	4.4E−03	2.3E+00	**4.2E−148**
	SR	0%	0%	0%	0%	0%	**100%**
*f*5	*Std*	1.0E+04	9.5E+03	1.6E+04	7.4E−01	6.2E+07	**3.2E+01**
	Ave	1.5E+04	1.5E+03	1.3E+04	9.7E+01	5.3E+07	**1.8E+01**
	SR	0%	0%	0%	0%	0%	0%
*f*6	*Std*	8.8E+00	3.7E+06	6.9E+02	9.5E−01	3.4E+04	**1.2E−01**
	Ave	2.0E+01	9.0E−06	3.4E+02	9.6E+00	1.5E+04	**8.4E−03**
	SR	0%	100%	0%	0%	0%	**60%**
*f*7	*Std*	8.2E+00	1.4E−01	2.1E+00	1.5E−03	1.6E+02	**3.0E−04**
	Ave	1.8E+01	1.0E+00	1.1E+00	2.2E−03	8.6E+01	**2.5E−04**
	SR	0%	0%	0%	0%	0%	**100%**
*f*8	*Std*	2.3E+03	1.6E+03	9.6E+02	1.8E+03	2.4E+04	**3.4E+03**
	Ave	−2.2E+04	−1.6E+04	−4.7E+03	−1.6E+04	2.4E+03	**−4.0E+04**
	SR	0%	0%	0%	0%	0%	**60%**
*f*9	*Std*	5.2E+01	5.8E+01	1.3E+01	2.2E+00	7.3E+02	**0**
	Ave	5.6E+02	7.5E+02	2.0E+02	6.9E−01	8.2E+01	**0**
	SR	0%	0%	0%	0%	0%	**100%**
*f*10	*Std*	3.6E−01	5.2E−01	3.1E−01	8.3E−15	1.9E+01	**0**
	Ave	3.7E+00	9.1E−02	6.4E+00	1.1E−13	1.9E−01	**8.8E−16**
	SR	0%	90%	0%	100%	0%	**100%**
*f*11	*Std*	7.8E−02	2.0E−01	1.0E+01	2.8E−03	2.9E+02	**0**
	Ave	2.6E−01	4.9E−02	9.5E+01	5.1E−04	1.1E+02	**0**
	SR	0%	83%	0%	100%	0%	**100%**
*f*12	*Std*	1.5E+00	2.8E+02	1.3E+00	7.5E−02	1.3E+08	**2.0E−04**
	Ave	3.2E+00	5.1E+01	5.0E+00	2.5E−01	1.4E+08	**2.2E−04**
	SR	0%	0%	0%	0%	0%	**100%**

Figure 1 also shows clearly the different abilities of the compared algorithms. The OMFO, illustrated with a red line, possess a faster convergence curve and is capable of finding a faster solution than the original MFO algorithms and PSO, DE, GSA and GWO, in every function (*f*1–*f*12), as the example of the three functions *f*1, *f*5 and *f*8.

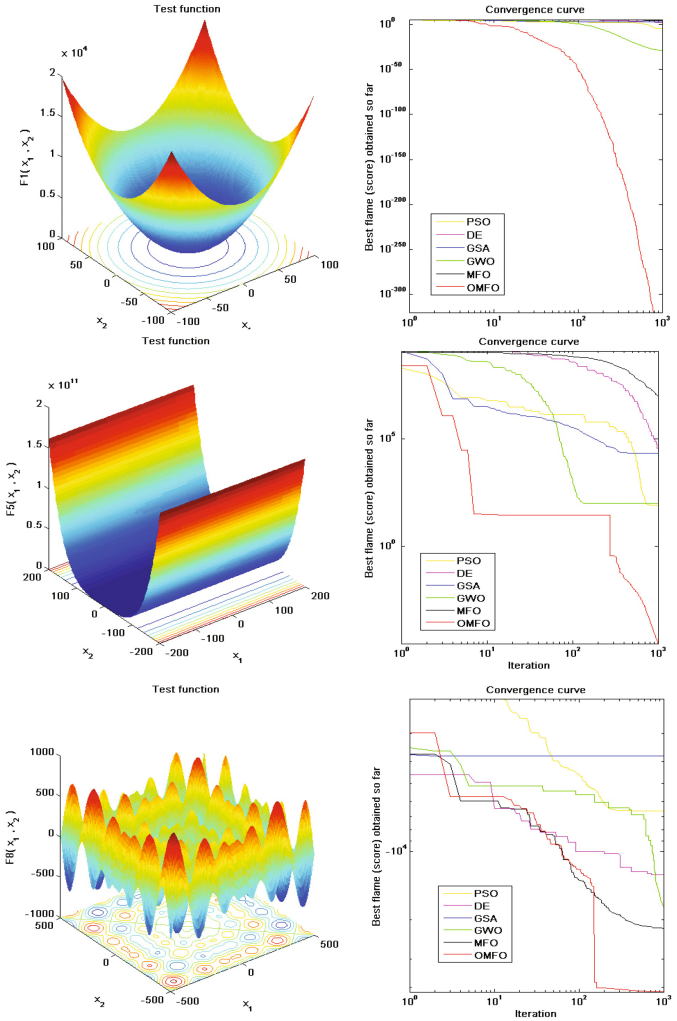


Fig. 1. The sample convergence curve of function *f*1, *f*5 and *f*8 compared between OMFO and MFO, PSO, DE, GSA and GWO.

5 Conclusion

An opposition-based moth-generating scheme was proposed and was used to overcome the disadvantage of existing MFO operations that produced a low convergence speeds and affected the quality of the final solution. The proposed method presents the OMFO algorithm by adding the enhanced function of a new opposition-based moth-generating scheme into the updating swarm of moths step. The experiment was conducted on a set of 12 commonly used benchmark functions, which was provided to evaluate the experiment. The results of OMFO's abilities were compared to the original MFO and

four well-known algorithms that included PSO, DE, GSA and GWO. The experimental results showed that OMFO outperformed MFO and other algorithms. For future works, OMFO will be applied to the real world application such as data exploration and data clustering.

References

1. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: 6th IEEE International Symposium on Micro Machine and Human Science, pp. 39–43. IEEE Press, New York (1995)
2. Karaboga, D., Basturk, B.: On the performance of Artificial Bee Colony (ABC) algorithm. Appl. Soft Comput. **8**, 687–697 (2008)
3. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. IEEE Comput. Intell. Mag. **1**, 28–39 (2006)
4. Filho, C.J.A.B., et al.: Fish school search. Nature-Inspired Algorithms for Optimization, vol. 193, pp. 261–277. Springer, Berlin (2009)
5. Corazza, M., Fasano, M., Gusso, R.: Particle swarm optimization with non-smooth penalty reformulation for a complex portfolio selection problem. Appl. Math. Comput. **224**, 611–624 (2013)
6. Yang, J., Zhuang, J.: An improved ant colony optimization algorithm for solving a complex combinatorial optimization problem. Appl. Soft Comput. **10**, 653–660 (2010)
7. Brajevic, I., Tuba, M.: An upgraded Artificial Bee Colony (ABC) algorithm for constrained optimization problems. J. Intell. Manuf. **24**, 729–740 (2013)
8. Boulkabeit, I., et al.: Finite element model updating using fish school search optimization method. In: 11th Brazilian Congress on Computational Intelligence, pp. 447–452 (2013)
9. Mirjalili, S.: Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl. Based Syst. **89**, 228–249 (2015)
10. Tizhoosh, H.R.: Opposition-based learning: a new scheme for machine intelligence. In: International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA 2005), pp. 695–701 (2005)
11. Mirjalili, S.: The ant lion optimizer. Adv. Eng. Softw. **83**(C), 80–98 (2015)
12. Tasgetiren, M.F., et al.: Differential evolution algorithms for the generalized assignment problem. In: IEEE Congress on Evolutionary Computation, pp. 2606–2613 (2009)
13. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. Inf. Sci. **179**, 2232–2248 (2009)
14. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)

A Hybrid of Fractal Code Descriptor and Harmonic Pattern Generator for Improving Speech Recognition of Different Sampling Rates

Rattaphon Hokking(✉) and Kuntpong Woraratpanya

Faculty of Information Technology,
King's Mongkut Institute of Technology Ladkrabang, Bangkok, Thailand
ratthapon.h@gmail.com, kuntpong@gmail.com

Abstract. Currently, the different sampling rate for speech recognition is a grand challenge due to supporting applications of divergent platform devices, such as mobile device interaction, interactive voice response system, voice search, voice dictation and voice identification. Furthermore, such applications require efficient speech features to represent input signals. However, the different sampling rates of speech signals lead to the different features. This phenomenon comes from speech harmonic signal lost. It becomes a key factor that decreases the speech recognition rate. Therefore, this paper proposes a hybrid of fractal code descriptor and harmonic pattern generator to convert all different sampling rate signals to standardized signals. In this method, an independent resolution property of fractal code descriptor is applied to training and testing speech signals. Then, the pitches of such signals are used to recover harmonic pattern of lost signals. This method can effectively reconstruct speech signals at any sampling rates. When its performance is evaluated with AN4 corpus of CMU Sphinx speech recognition engine, the experimental results show that the proposed method can significantly improve the speech recognition rate, even if the sampling rate of testing speeches differs from that of training speeches.

Keywords: Fractal code descriptor · Mel frequency cepstral coefficient · Speech recognition · Different sampling rate · Resolution independent · Harmonic reconstruction

1 Introduction

Different sampling rates in speech recognition become a grand challenge in many applications such as mobile device interaction, interactive voice response system, voice search and voice dictation [1–3]. In general of speech recognition, mel frequency cepstral coefficient (MFCC) presented in [4] is the most commonly used method to extract features from speech signals due to providing good discrimination and compactness properties [5]. The success of applying MFCC in speech recognition comes from the same sampling rate used for training and testing speech signals [5]. Nevertheless, in real world applications, the different sampling rate of speech signals is

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_4

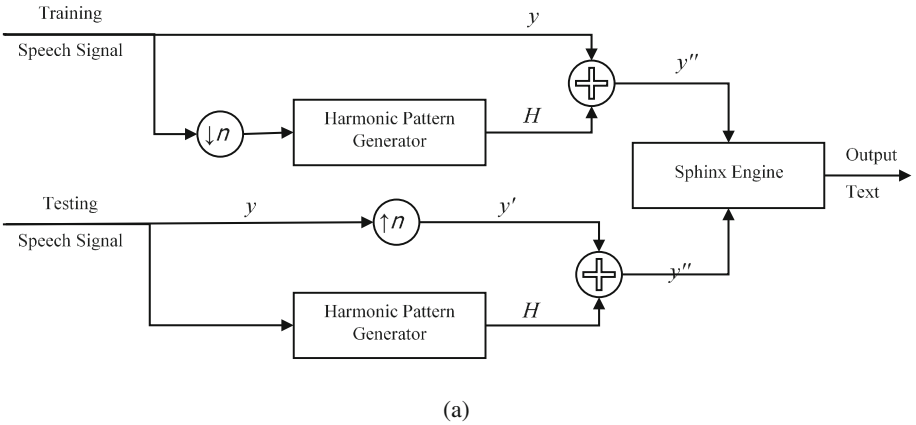
frequently used for supporting divergent platform devices. This decreases speech recognition rate as reported by Sanderson et al. [5].

In this issue, many methods have been proposed to improve the accuracy of speech recognition of different sampling rates. Hirsch et al. [6] presented MFCC feature extraction obtained from three different sampling rates—8 kHz, 11 kHz, and 16 kHz. The comparative performance evaluation discloses that the word error rate is least possible when the sampling rate of the training and testing sets is 11 kHz. However, when the testing set is sampled with 8 kHz and 16 kHz whereas the training set is still retained at 11 kHz sampling rate, the word error rate goes up. Also, Kopparapu et al. [7] introduced six methods for extracting MFCC features, namely MFCC features at different sampling rates. One of them is the down-sampling frequency spectrum method, which successfully achieves in terms of high Pearson correlation between reconstructed and original MFCCs. In this method, the high mel frequency spectrum is depressed due to the down-sampling technique, thus making the recognition accuracy dropping as well. To recover the high mel frequency spectrum, Kopparapu et al. [8] modified the mel filter bank proposed in [7] to synthesize the high mel frequency spectrum. That is, the mel frequency spectra of 8 kHz and 16 kHz sampling speech signals are extracted, and then the high mel frequency spectrum is generated from the low mel frequency spectrum instead. In this way, the extracted features still provide high Pearson correlation. However, the high mel frequency spectrum of the down-sampling speech is not generated from the exact high frequency spectrum, thus the accuracy of speech recognition depends on the sampling rate of the testing set.

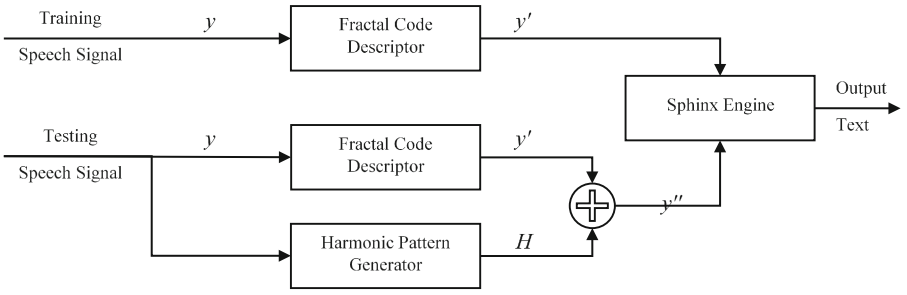
Although existing methods can solve different sampling rate problem by using MFCC feature extraction, they cannot extract the efficient high-frequency spectrum. Especially at a low sampling rate, some important spectrum is lost [9]. It makes the recognition accuracy decreasing.

Therefore, this paper proposes a hybrid of fractal code descriptor and harmonic pattern generator to convert all speech signals at different sampling rates to standardized signals. As illustrated in Fig. 1, a framework of the proposed method is comprised of two important procedures: (i) fractal code descriptor [10, 11] and (ii) harmonic pattern generator. In the first procedure, the fractal code descriptor transforms an input speech signal at any sampling rate to an output speech signal at desired sampling rate. Usually, the desired sampling rate should be set to the highest sampling rate for training. In the second procedure, the harmonic pattern generator provides a suitable harmonic pattern constructed from pitches of the input speech signal. Then the output of the former procedure is recovered harmonic pattern by adding the output of the latter procedure. Finally, the reconstruct ted speech signal is fed into CMU Sphinx speech recognition engine [12]. In this way, the experimental results show that the proposed method can significantly improve the accuracy of speech recognition at different sampling rates.

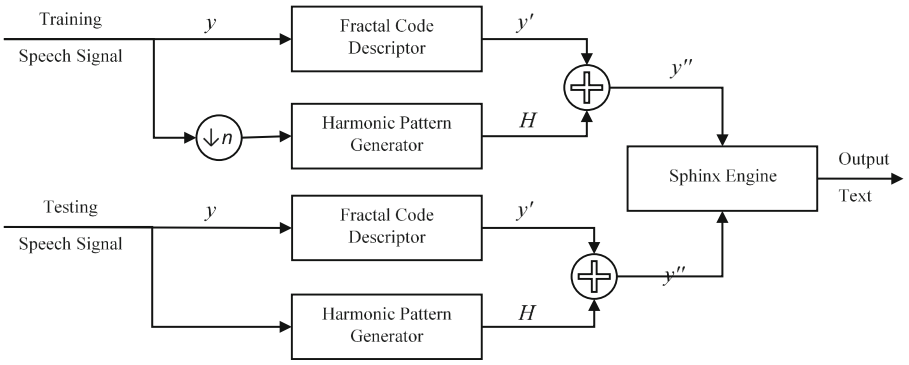
The remainder of this paper is organized as follows. An analysis of different sampling rate, fractal code descriptor, and harmonic pattern generator are described in Sect. 2. Experiments are discussed in Sect. 3. Finally, the conclusion is presented in Sect. 4.



(a)



(b)



(c)

Fig. 1. A framework of speech recognition: (a) a proposed harmonic pattern generator, (b) a semi-hybrid of fractal code descriptor and harmonic pattern generator, and (c) a hybrid of fractal code descriptor and harmonic pattern generator.

2 Proposed Method

In this section, the impact of different sampling rates is analyzed by determining the frequency spectrum. After that, the use of fractal code descriptor is explained to reduce the effect of the problem. In the last subsections, the hybrid of fractal code descriptor and harmonic pattern generator is described to improve the performance of fractal code descriptor in order to overcome the different sampling rate problem.

2.1 Analysis of Different Sampling Rate

In general, the different sampling rates of two speech signals provide the different features, even though both signals are from the same speech. This phenomenon has a great impact on efficient speech features. It becomes a key factor decreasing accuracy rate in speech recognition. As shown in Fig. 2, when a speech signal is sampled with different sampling rates, namely 8 kHz and 16 kHz, its frequency spectra are evidently different (see Figs. 2(a) and (b)). In the same way, its mel frequency spectra are also clearly different (see Figs. 2(c) and (d)). Furthermore, it is proved that the different sampling rates are a cause of the different features as illustrated in Figs. 3(a) and (b) and reported in [8]. This phenomenon is proved and overcome as illustrated in experiment section.

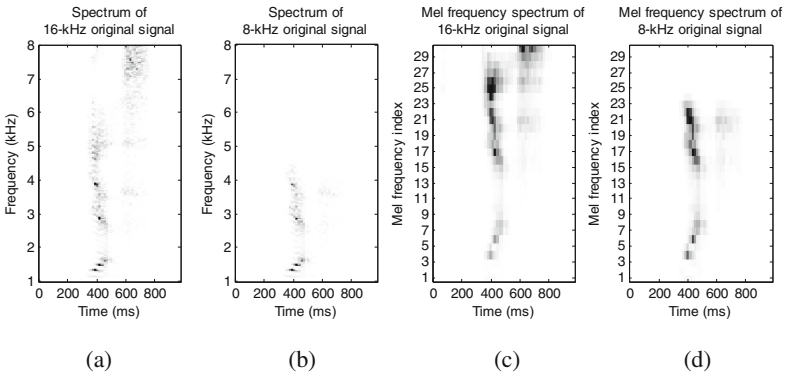


Fig. 2. A comparison of frequency spectra of an original speech signal: (a) 16-kHz spectrum, (b) 8-kHz spectrum, (c) 16-kHz mel frequency spectrum, and (d) 8-kHz mel frequency spectrum.

Although a recent paper [11] has proposed the fractal code descriptor to recover the high frequency spectrum of an input speech signal from the low sampling rate, it still cannot perfectly recover the harmonic pattern signal as shown in Fig. 4. Figure 4(d) shows that only fractal code descriptor cannot restore high frequency harmonic pattern of a reconstructed signal, 8 to 16 kHz sampling rate. The following subsections describe a method of recovering high frequency spectra of speech signals by means of fractal code descriptor and harmonic pattern generator in order to solve the different sampling rate problem.

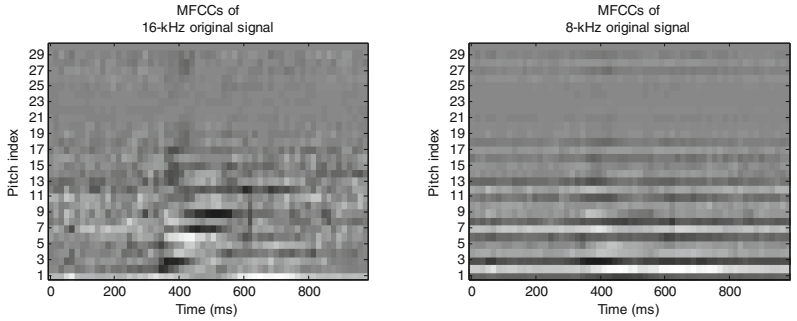


Fig. 3. A comparison of mel frequency cepstral coefficients (MFCCs) of an original speech signal: (a) MFCCs of 16-kHz original signal and (b) MFCCs of 8-kHz original signal.

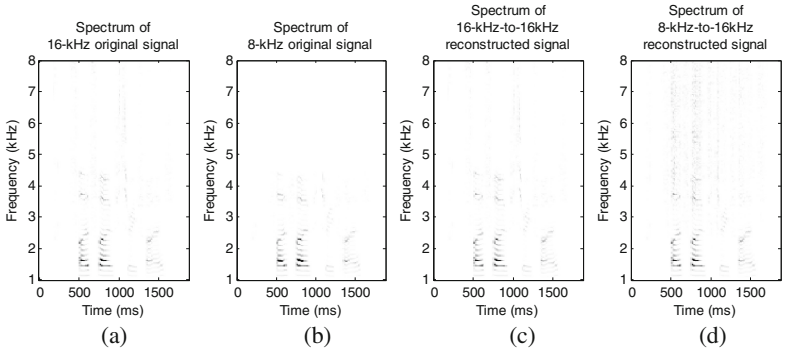


Fig. 4. A comparison of frequency spectra of original and reconstructed speech signals: (a) 16-kHz original spectrum, (b) 8-kHz original spectrum, (c) 16-kHz-to-16-kHz reconstructed spectrum, and (d) 8-kHz-to-16-kHz reconstructed spectrum. Note: All reconstructed spectra are restored by fractal code descriptor.

2.2 Fractal Code Descriptor

A fractal code descriptor generated by fractal coding technique is a method that can reconstruct the high frequency spectrum of an input speech signal from the low sampling rate by exploiting the property of resolution independent [10, 11]. This property is used to encode the low sampling rate speech signal, and then decode it with up sampling to the higher rate speech signal. It can help generate an efficient acoustic model with independent sampling rates. In other words, the input speech signal can be classified by using the trained speech model whose sampling rates are different from the test speech signal. To apply the fractal code descriptor technique, the feature with resolution independent of an input speech signal is extracted by the following steps.

Step 1: Generate the fractal code descriptor from an input speech signal using fractal encoding.
Step 2: Reconstruct the new speech signal from the fractal code descriptor by using fractal decoding. At this stage, an important α-parameter is computed from a ratio, f_s^{train}/f_s^{test}, where f_s is a sampling frequency. For more details of manipulation and parameter setting of fractal coding, it can be found in [11].

2.3 Harmonic Pattern Generator

Although fractal code descriptor is a successful method that can recover the higher frequency spectrum from the speech signal at a low sampling rate, it cannot recover the harmonic pattern in the high frequency spectrum. The harmonic pattern is a characteristic pitch of a speech signal. As already described in Sect. 2.1, the low sampling rate commonly depresses the harmonic signals, thus making its harmonic pattern lost. It is crucial when the MFCC is extracted from an input speech signal (see Fig. 3). Therefore, this paper proposes the harmonic pattern generator for restoration. Here, a pitch of speech signal is used to reconstruct the harmonic pattern. Moreover, the multiple pitches also can be used to generate the more complicated harmonic pattern. As a result, when a hybrid of fractal code descriptor and harmonic pattern generator is applied to speech recognition engine, the speech recognition rate is significantly improved.

Harmonic frequency of a speech signal is a fundamental frequency multiplying with a positive integer, and a harmonic pattern is a periodic pattern of magnitude in frequency spectrum of a speech signal. The speech signal spectrum with a strong harmonic pattern is essential for extracting the MFCC, especially for cepstrum extraction. The cepstrum represents pitches of fundamental frequencies in a speech signal. In this paper, these pitches are used to reconstruct harmonic signals to improve harmonic patterns in the high frequency spectrum, whereas the harmonic pattern in the low frequency spectrum is intact. The harmonic pattern construction can be described as follows

Step 1: Transform an original input speech signal y to frequency spectrum Y using Eq. (1). At the same time, the reconstructed speech signal y' is also transformed to frequency spectrum Y'.

$$Y_u = \sum_{n=0}^{N-1} y_n \cdot e^{-2\pi i\left(\frac{un}{N_u}\right)} \tag{1}$$

where Y_u, n, N, u and N_u are the u^{th}-spectrum, an index of speech samples, a total number of speech samples, an index of frequency spectra, and a total number of frequency spectra, respectively.

Step 2: Extract pitches P from log of original frequency spectrum Y_u by using cepstrum extraction as defined in Eq. (2).

$$P_v = \sum_{u=1}^{N_u} \log(Y_u) \cdot \cos\left(\frac{\pi}{N_u}\left(uv + \frac{u}{2}\right)\right) \tag{2}$$

where v is a pitch index.

Step 3: Select the peaks p of all pitches P by using Eq. (3).

$$p = \{v | P_{v-1} < P_v \text{ and } P_v > P_{v+1}\} \tag{3}$$

Step 4: Compute the frequency of the i^{th}-peak of pitch f_{p_i} using Eq. (4).

$$f_{p_i} = \frac{f_s^{in}}{p_i} \tag{4}$$

where p_i and f_s^{in} are the i^{th}-peak of pitches and the sampling rate of an input signal, respectively.

Step 5: Generate the harmonic pattern H_u from the N_p largest peaks of pitches using Eq. (5).

$$H_u = \prod_{i=1}^{N_p} \sin\left(2\pi u \frac{f_{p_i}}{f_s^{rec}} + \frac{\pi}{2}\right) \tag{5}$$

where f_s^{rec} is the sampling rate of a reconstructed signal. Note that p is sorted by $P_{p_i} \leq P_{p_{i+1}}$.

Step 6: Weight the harmonic pattern to suppress the noise from harmonic pattern using Eq. (6).

$$H_u^{'} = H_u \cdot e^{-\frac{u}{N_u}} \tag{6}$$

Step 7: Recover the harmonic pattern of high frequency spectrum by using Eq. (7).

$$Y_u'' = \begin{cases} Y_u & ; f_u \leq \frac{f_s^{rec}}{2\alpha} \\ H_u^{'} \cdot Y_u^{'} & ; otherwise \end{cases} \tag{7}$$

where f_u and α are the frequency of u^{th}-spectrum and the ratio of f_s^{train}/f_s^{test}, respectively.

In practice, a speech signal is processed on a frame-wise operation in order to maximize the recognition performance. In this paper, multiple pitches are used to construct the harmonic pattern and to reconstruct the high frequency spectrum of reconstructed signal as well.

3 Experiments and Discussion

In order to test performance of the proposed method in terms of recognition accuracy, the experiments are set up. The dataset used is AN4 speech corpus consisting of 948 training and 130 testing speech signals. All speech signals are automatically segmented by Sphinx engine for 7,338 training and 773 testing words. This paper uses two different feature sets as introduced in [8]. (i) A feature set A is 30 MFCCs and (ii) a feature set B is 39 dimensional feature vectors from 13 MFCC concatenated with 13 Δ MFCC and 13 Δ^2 MFCC coefficients. Then, these features are recognized by means of the CMU Sphinx ASR.

For performance comparison, three proposed methods are introduced: the proposed harmonic pattern method (HP), a semi-hybrid of fractal code descriptor and harmonic pattern generator (Proposed Method #1) and a hybrid of fractal code descriptor and harmonic pattern generator (Proposed Method #2). The baselines are Kopparapu's methods [7, 8] and fractal code descriptor (FCD) [11]. All methods except [11] are tested in two cases: (i) the different sampling rate ($f_s^{train} \neq f_s^{test}$), a training set is sampled at 16 kHz and a testing set is sampled at 8 kHz; and (ii) the same sampling rate ($f_s^{train} = f_s^{test}$), both training and testing sets are sampled at 16 kHz. The frameworks of three proposed methods are illustrated in Fig. 1. The range block size of fractal code descriptors is 4, α-parameter [11] is calculated by f_s^{train}/f_s^{test}, the number of decoding iterations is set to 15, and the N_p is set to 3.

Based on the experimental design as mentioned in previous paragraph, there are three experiments for performance comparisons. In the first experiments, the results show that in case of different sampling rate, the proposed HP method only outperforms Kopparapu's methods [7] in both feature sets. In case of the same sampling rate, the proposed HP method outperforms all baseline methods as shown in Table 1. This improvement confirms that harmonic pattern has an impact on accuracy rate. In the second experiment for the proposed method #1, the training set is sampled with 16-kHz sampling rate for providing to acoustic model and the harmonic pattern generator is not yet applied for this stage as shown in Fig. 1(b). In the testing stage, the harmonic pattern generator is applied to testing set. Based on this experimental setup, the recognition rate of the same sampling rate is higher than all baseline methods, including the proposed HP method, in both feature sets. On the other hand, the recognition rate of the different sampling rate is higher than all baseline methods in case of feature set A, but 4.14% lower than Kopparapu's method [8] in case of feature set B. Finally, in the third experiment, the harmonic pattern generator is applied to both training and testing sets as schematically demonstrated in Fig. 1(c). In this case, the input speech signals of training set are down-sampling by half prior to feed to the harmonic pattern generator. The experimental results reveal that the proposed method #2 outperforms all baseline methods. This achievement comes from the better improvement of recovering the high frequency spectrum and harmonic pattern for both training and testing speech signals. Figures 5(a), (b), and (c) demonstrate the spectra of 16-kHz original training signals, 16-kHz-to-16-kHz reconstructed training signals from FCD, and 16-kHz-to-16-kHz reconstructed training signals from the proposed method #2, respectively. In order to distinguish the different spectra between the original one (Fig. 5(a)) and two

Table 1. A comparison result of recognition rates between the proposed and baseline methods based on the different and same sampling rates (training/testing: 16 kHz/8 kHz and 16 kHz/16 kHz).

Method	Different sampling rate (16/8)		Same sampling rate (16/16)	
	Feature set A	Feature set B	Feature set A	Feature set B
Kopparapu's method [7]	3.88	11.77	43.21	81.11
Kopparapu's method [8]	37.00	77.23	43.21	81.11
FCD method [11]	41.27	–	53.30	–
Proposed HP method	28.07	62.87	54.46	83.05
Proposed method #1	44.89	73.09	54.46	83.05
Proposed method #2	**48.00**	**77.49**	**56.53**	**83.57**

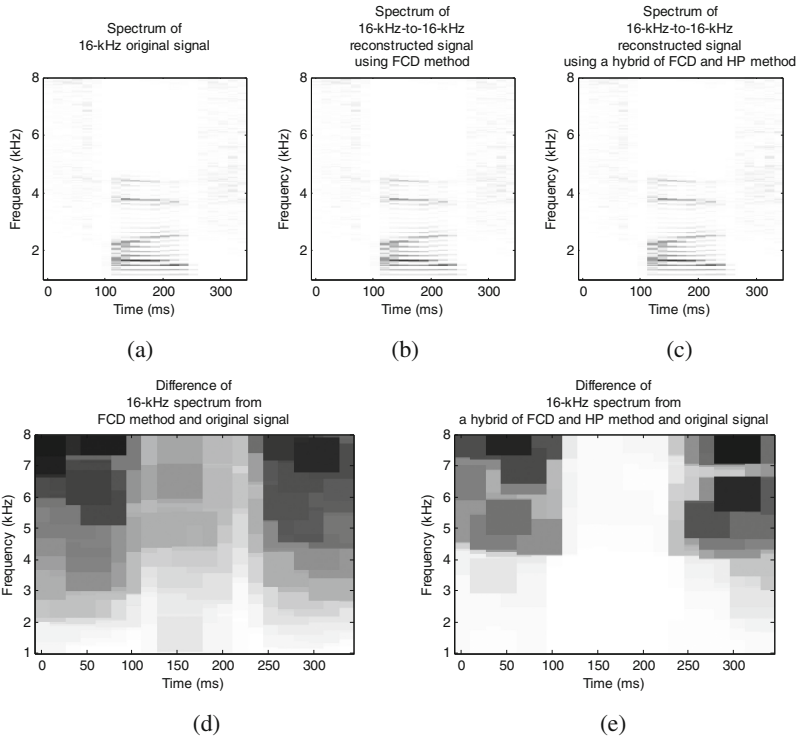


Fig. 5. A comparison of frequency spectra of original and reconstructed speech signals recovered by the proposed and baseline methods: (a) 16-kHz original spectrum, (b) 16-kHz-to-16-kHz reconstructed spectrum using fractal code descriptor, (c) 16-kHz-to-16-kHz reconstructed spectrum using fractal code descriptor and harmonic pattern generator, (d) the difference of original and reconstructed spectra using fractal code descriptor, and (e) the difference of original and reconstructed spectra using a hybrid of fractal code descriptor and harmonic pattern generator.

reconstructed spectra (Figs. 5(b) and (c), (d) and (e)) illustrate the difference of the original training and reconstructed spectra using two methods: the FCD and hybrid methods. The dark regions of Figs. 5(d) and (e) are much more different, especially at the higher frequency spectrum. The FCD and hybrid methods cannot recover the high frequency spectrum in silence intervals. However, the hybrid method can recover the harmonic pattern in higher frequency of speech intervals while FCD method cannot.

4 Conclusion

This paper proposes a hybrid of fractal code descriptor and harmonic pattern generator to improve speech recognition at different sampling rates. In this method, the fractal code descriptor is used to reconstruct the high frequency spectrum of speech signals that is sampled at a low sampling rate. At the same time, the harmonic pattern generator is used to recover the harmonic pattern based on pitches of speech signals. As a result, when the proposed method is evaluated with AN4 corpus of CMU Sphinx speech recognition engine, the experimental results show that the proposed method can significantly improve the speech recognition rate, even if the sampling rate of testing speeches differs from that of training speeches.

References

1. Feng, J.: A general framework for building natural language understanding modules in voice search. In: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5362–5365 (2010)
2. Rabiner, L.R.: Applications of speech recognition in the area of telecommunications. In: 1997 IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 501–510 (1997)
3. Jin, Q., Toth, A.R., Schultz, T., Black, A.W.: Speaker de-identification via voice transformation. In: IEEE Workshop on Automatic Speech Recognition Understanding, ASRU 2009, pp. 529–533 (2009)
4. Zigelboim, G., Shallom, I.D.: A comparison study of cepstral analysis with applications to speech recognition. In: International Conference on Information Technology: Research and Education, pp. 30–33 (2006)
5. Sanderson, C., Paliwal, K.K.: Effect of different sampling rates and feature vector sizes on speech recognition performance. In: Proceedings of IEEE TENCON 1997, IEEE Region 10 Annual Conference, Speech and Image Technologies for Computing and Telecommunications, vol. 1, pp. 161–164 (1997)
6. Hirsch, H.G., Hellwig, K., Dobler, S.: Speech recognition at multiple sampling rates. In: EUROSPEECH 2001 Scandinavia, 7th European Conference on Speech Communication and Technology, 2nd INTERSPEECH Event, Aalborg, Denmark (2001)
7. Kopparapu, S., Laxminarayana, M.: Choice of mel filter bank in computing MFCC of a resampled speech. In: 2010 10th International Conference on Information Sciences Signal Processing and their Applications (ISSPA), pp. 121–124 (2010)
8. Kopparapu, S., Bhuvanagiri, K.: Recognition of subsampled speech using a modified mel filter bank. Comput. Electr. Eng. **39**(2), 655–662 (2013)

9. Jeff, R.: The effect of bandwidth on speech intelligibility. Technical report, Polycom Inc., USA (2003)
10. Jacquin, A.: Fractal image coding: a review. Proc. IEEE **81**(10), 1451–1465 (1993)
11. Hokking, R., Woraratpanya, K., Kuroki, Y.: Speech recognition of different sampling rates using fractal code descriptor. In: 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE), pp. 1–5, Khon Kaen (2016)
12. Lee, K.F., Hon, H.W., Reddy, R.: An overview of the SPHINX speech recognition system. IEEE Trans. Acoust. Speech Signal Process. **38**(1), 35–45 (1990)

Ensemble Features Selection Algorithm by Considering Features Ranking Priority

Puripat Thongkam(✉) and Pakorn Leesutthipornchai

Department of Computer Science, Faculty of Science and Technology, Thammasat University, Pathum Thani, Thailand
th.puripat.t@gmail.com, pakornl@cs.tu.ac.th

Abstract. Feature selection is a pre-processing for choosing relevant features and ignores features that tend to have no predictive information. Feature selection is applied to improve the accuracy of classification process. High relevant features have a tendency to get high classification performance. This paper proposed the ensemble of multiple feature ranking techniques by considering ranker priority for feature selection. Five individual feature ranking algorithms (information gain, gain ratio, symmetrical uncertainty, reliefF and oneR) are investigated and considered together as ensemble, based on ranking priority. The lung cancer, lymphoma, breast cancer, ovarian cancer and leukemia datasets were gathered from Kent Ridge bio-medical data and Machine Learning data repository. The datasets are applied to ensemble features selection algorithm. The obtained results are compared to results from individual feature ranking algorithms and the existing ensemble algorithm. The selected features are applied to classification algorithms. Area under the curve (AUC), precision and recall values from six classification algorithms are used to evaluate the obtained features. The experimental results show that the selected features from proposed ensemble features selection algorithm are greater than those of individual feature ranking techniques and the existing ensemble features selection algorithm.

Keywords: Feature selection · Ranker · Ensemble

1 Introduction

Data mining is the technique for discovering relationship, pattern, or knowledge from database, data warehouse, or information storage. In present day, the size of data is increasing every hour according to records and dimensions. Knowledge extraction from large size of data is difficult and time consuming. Feature selection is a pre-process task to reduce dimension of data. Feature selection removes irrelevant features and determines features that have high correlation with the output class [1]. The selected features after the feature selection process can represent the whole set of data with less dimension. The knowledge that is discovered from less dimension data is easy to explain, less processing time, and get higher precision. Many feature selection

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_5

techniques have been proposed to choose the best set of features (e.g., information gain and gain ratio). The combinations of feature selection techniques called "ensemble features selection" [1–5] have been proposed to improve the performance of classification algorithm. This paper proposed the ensemble features selection technique by considering ranker priority to select significant features. The performance of the selected features is evaluated by AUC, precision and recall values that are calculated from classification results. The obtained results are compared to results from individual feature ranking algorithms and the existing ensemble algorithm.

This paper is outlined as follows: Sect. 2 reviews and discusses on feature selection techniques. Section 3 shows the proposed ensemble features selection algorithm. Section 4 shows data sets and the obtained results. Lastly, Sect. 5 concludes the research work.

2 Feature Selection Techniques

Feature selection is an important pre-processing technique that has been used in many fields of researches and analysis for many years. Since the data nowadays in many fields such as biomedical research, military thread detection, traffic accident are getting bigger in terms of records and features. All these enormous datasets cause large scalability and have problem in performance of learning algorithm [6]. Feature selection solves the issue of scalability and the performance of classification models by eliminating redundant or unrelated features from datasets. Typically, feature selection consists of three main procedures [7]. First procedure removes the irrelevant features. Second procedure removes the redundant features. Third procedure applies a feature selection algorithm to select remaining features. Feature selection can be categorized into two techniques. First technique is feature ranking. Feature ranking calculates the score of each attribute and then sorts them according to their scores. Second technique is feature subset selection. Feature subset selection chooses a subset of attributes which collectively increases the performance of the model. This paper is mainly focus on feature ranking technique. The feature ranking techniques that are considered in this research are Information Gain (IG), Gain Ratio (GR), Symmetrical Uncertainty (SU), ReliefF (RFF) and OneR Attribute Evaluation (OneR).

2.1 Information Gain

Information Gain (IG) selects feature based on entropy [8]. IG provides the mutual information of target variable (Y) and independent variable (X). Information gain reduces entropy of target variable (Y) by learning the state of independent variable (X). For computing information gain, let X be attribute and Y be class attribute. The information gain of a given attribute X with respect to class attribute Y is the reduction in uncertainty about the value of Y when the value of X is known. The value of Y is

measured by its entropy, $H(Y)$. The uncertainty of Y given the value of X is given by the conditional probability of Y given X, $H(Y|X)$ as Eq. (1).

$$I(Y;X) = H(Y) - H(Y|X) \tag{1}$$

When Y and X are discrete variables that have values in $\{y_1 \ldots y_k\}$ and $\{x_1 \ldots x_k\}$ then the entropy of Y is given by

$$H(Y) = -\sum_{i=1}^{i=k} P(Y = y_i) log_2 P(Y = y_i) \tag{2}$$

The condition entropy of Y given X is:

$$H(Y|X) = -\sum_{i=1}^{i} P(X = x_i) log_2 H(Y|X = x_i) \tag{3}$$

This paper uses IG for the reason that it defines which features in a given set of training feature vectors is most useful for discriminating between the classes to be learned by using criteria of entropy.

2.2 Gain Ratio

Gain Ratio (GR) is an extension of information gain. It is proposed to solve the bias problem of information gain [9]. Let D be a set that consists of d data samples with n distinct classes. The expected information for classifying a given sample is $I(D)$.

$$I(D) = -\sum_{i=1}^{n} p_i log_2 (p_i) \tag{4}$$

where p_i is the probability that an arbitrary sample belongs to class C_i. Let attribute A have v distinct values. Let d_{ij} be number of samples of class C_i in a subset D_j [9]. D_j contains those samples in D that have value a_j of A. The entropy based on partitioning into subsets by A as Eq. (5).

$$E(A) = -\sum_{i=1}^{n} I(D) \frac{(d_{1_i} + d_{2_i} + \ldots + d_{n_i})}{d} \tag{5}$$

The encoding information that would be gained by branching on A is $Gain(A)$.

$$Gain(A) = I(D) - E(A) \tag{6}$$

Decision tree applies a kind of normalization [9] to information gain using a "split information" value defined analogously with $Info(D)$.

$$SplitInfo_A(D) = -\sum_{i=1}^{v} \left(\frac{|D_j|}{|D|}\right) log_2 \left(\frac{|D_j|}{|D|}\right) \tag{7}$$

This value represents the information computed by splitting the dataset D, into v partitions, corresponding to the v outcomes of a test on attribute A. Finally, the gain ratio is defined as *GainRatio*(*A*).

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)} \tag{8}$$

The attribute with maximum gain ratio is selected as the split attribute. This paper uses GR for the reason that once it solves the drawback of information gain wills the performance of classification be significantly improved if we combine IG and GR together.

2.3 Symmetrical Uncertainty

Symmetrical uncertainty (SU) is used to calculate the degree of association between discrete features based on hypothesis - "Good feature subsets contain features highly correlate with the class, not uncorrelated to each other" [10]. Symmetrical uncertainty is derived from entropy [10]. It is a symmetric measure that is applied to measure feature-correlation.

$$SU = 2.0 * \frac{H(X) + H(Y) - (X, Y)}{H(Y) + H(X)} \tag{9}$$

From the equation, *SU* is symmetrical uncertainty, $H(X)$ and $H(Y)$ represent the entropy of features X and Y. The value of symmetrical uncertainty ranges between 0 and 1. The value of 1 indicates that one variable (either X or Y) completely predicts the other variable. The value of 0 indicates that both variables are completely independent. This paper uses SU for the reason that it finds a nonrandom relationship between discrete features and the class features for both high or low dimensional of data.

2.4 ReliefF

ReliefF (RFF) is simple, fast and accurate algorithm that was proposed by Kira and Rendell in 1994 [11]. The algorithm works by measuring the ability of an attribute in separate instances. ReliefF consists of three main procedures. First procedure calculates the nearest miss and nearest hit. Second procedure computes the weight of the feature. Third procedure returns a ranked list of features or the top-k features according to a given threshold. The basic idea of reliefF is to illustrate instances randomly then

calculate their nearest neighbors and modify a feature weighting vector to give more weight to features that differentiate instance from neighbors of different classes [11]. This paper uses RFF for the reason that it can deal with noisy data and can be used for regression and classification problems.

2.5 OneR Attribute Evaluation

OneR or One Rule is an algorithm that provides a way to generate compact or accurate rules by focusing on a particular class at a time or in another word, it approaches to find a classification rule as it generates one level decision tree by considering a classification as r = (a, c) where a is a precondition which performs a series of tests that can be evaluated as true or false and c is class that applies to instances covered by rule r [11]. OneR constructs rules and tests a single attribute at a time and branches for every values of that attribute. For every branches, the class with the best classification is the most often appear in the training data [11]. This paper uses OneR for the reason that it finds the best predictive feature by counting the smallest total error of features in dataset.

2.6 Ensemble Features Ranking Technique

Ensemble features ranking technique is a method that combines the results from multiple feature ranking techniques into single ranking list. The combination list may be aggregated by using weighted vote, linear aggregation function, or etc. Multiple feature ranker techniques generate individual ranking lists then all lists are combined to single ranking list by using ranking order of features [12]. The existing ensemble features ranking techniques are considered and investigated [12–19]. The enhanced version of ensemble algorithm is proposed in this paper by using the ranking score and frequency of features 'appearance technique from existing ensemble algorithm as described in the next section.

3 Proposed Algorithm

The proposed algorithm focuses on the ensemble approach. The algorithm combines the results of individual feature selection ranking techniques by using the fusion-based aggregation technique with priority value. Normally, the ensemble techniques use a combination of the ranked scores of multiple feature selection techniques in a traditional way such as the summation, mean, frequency or taking the highest or lowest ranking scores [19]. Only the ranking score or frequency cannot define the preference of a feature in all elements of the lists. From our experiment, the priority for each feature ranking techniques and the order of the feature in the ranked list is significant. So, the proposed ensemble algorithm considers order, frequency of individual feature

in all ranking lists and the priority for each feature ranking techniques to generate a final ranking list.

The proposed ensemble algorithm can be with any number of ranking lists. The feature ranking techniques that have been used in this research are Information Gain, Gain Ratio, Symmetrical Uncertainty, ReliefF and OneR Attribute Evaluation. All of them are from WEKA software library [20].

The proposed ensemble approach consists of three steps shown in Fig. 1. First step, the algorithm creates a set of five ranking lists that were generated by five feature ranking techniques (IG, GR, SU, RFF and OneR) from the N features of the dataset then selects top-k features in each ranking list. Second step, the algorithm calculates the frequency of individual feature by finding an order of the individual feature in each ranking list (Fr) and computes with priority value of the feature ranking techniques. The priority value of each feature ranking technique came from the experiment that using Information Gain, Gain Ratio, Symmetrical Uncertainty, ReliefF and OneR to find out how well each of these feature ranking techniques improve the performance of classification by using area under the curve (AUC) as metric. The classification techniques that were considered in this paper are k-Nearest Neighbor, Decision Tree, Naïve-Bayes, Random Forest, Logistic Regression and Support Vector Machine.

According to average AUC value from Tables 1 and 2, the preference order of 5 feature ranking techniques is (1) Symmetrical Uncertainty, (2) ReliefF, (3) Information Gain, (4) Gain Ratio and (5) OneR. Then the priority value will be assigned to the preference order (i.e., Symmetrical Uncertainty = 5, ReliefF = 4, Information Gain = 3, Gain Ratio = 2 and OneR = 1).

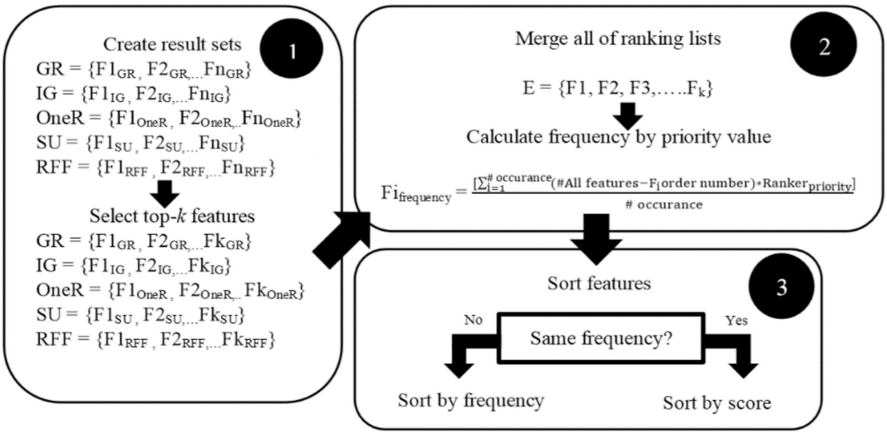


Fig. 1. Overview of proposed ensemble features selection algorithm

From our experiment, the AUC values obtained from 5 feature ranking techniques are calculated and showed in Tables 1 and 2. All ranker techniques are rearranged by AUC value in descending order.

After we obtained the priority value for each ranker. The algorithm counts the occurrence of an individual feature (j) in all ranking lists and merges all of ranking lists into single set then calculates the frequency. The frequency of each feature is calculated according to Eq. (10).

$$Feature_{frequency} = \frac{\left[\sum_{i=1}^{j}(N - F_r) * Ranker_{priority\ value}\right]}{j} \tag{10}$$

Third step, the proposed algorithm sorts all features based on frequency value then select top-k features. Two or more features may have the same frequency. This paper resolves this problem by using feature's score that is determined by the average of ranking scores in all ranking lists.

Table 1. Average AUC values of 6 classifiers from 5 ranker techniques of top 5, 10, 15 and 20 features for lung cancer dataset.

Number of top *k* features	Ranker techniques				
	Gain Ratio	ReliefF	Symmetrical	OneR	Information gain
Top 5 features	0.7407	0.8222	0.8197	0.7592	0.8097
Top 10 features	0.7567	0.7919	0.7818	0.7601	0.7705
Top 15 features	0.7727	0.7647	0.7773	0.6964	0.7773
Top 20 features	0.7690	0.7758	0.7798	0.7433	0.7759
Average	0.7598	0.7887	0.7900	0.7340	0.7834
Rank	4	2	1	5	3

Table 2. Average AUC values of 6 classifiers from 5 ranker techniques of top 25, 50, 100 and 500 features for lymphoma dataset.

Number of top *k* features	Ranker techniques				
	Gain ratio	ReliefF	Symmetrical	OneR	Information gain
Top 25 features	0.8955	0.9213	0.9414	0.8974	0.8372
Top 50 features	0.9431	0.9340	0.9510	0.9083	0.9857
Top 100 features	0.9495	0.9394	0.9588	0.9205	0.9579
Top 500 features	0.9467	0.9441	0.9553	0.9213	0.9550
Average	0.9337	0.9347	0.9516	0.9119	0.9340
Rank	4	2	1	5	3

The ensemble features selection algorithm by considering feature ranking priority is showed as follows.

```
Algorithm: Ensemble Features Selection Algorithm by Con-
sidering Features Ranking Priority
_____________________________________________________________
Input : Dataset with N features
1: Initialize E as output variable and F[i] for pro-
cessing
2: For each i ranker
3:   F[i] = each ranker result
4: EndFor
5: For each i in F //Iteration for each features
6:   For each feature in all ranking lists  from 0-(K-1)
7:    if(F[i][k] not in E) add to E and set count = 1
8:    else set count++
9:    set E[F[i][k]].rank = K , E[F[i][k]].score = score
10:  EndFor
11: EndFor
12: For each features in E for 0-(K-1)
13:      E[k]_frequency = [Σ_{i=1}^{E[k].count} (N−E[k],rank)∗priority{IG,GR,SU,OneR,RFF}] / E[k].count
14: EndFor
15: Sort the features in E based on frequency.
16: Check for same frequency, if any then sort by score.
17: Selects the top-k features.
```

4 Experimental Design and Results

4.1 Datasets

In this paper, the lung cancer, lymphoma, breast cancer, ovarian cancer and leukemia datasets are gathered. The datasets are from Kent Ridge bio-medical data [21] and Machine Learning data repository [22].

Lung cancer dataset consists of 57 features (attributes) and 32 instances. The class attribute for lung cancer dataset has 3 distinct values. Lymphoma dataset consists of 4,027 features and 96 instances. The class attribute for lymphoma dataset has 9 distinct values. Breast cancer dataset consists of 24,482 features and 78 instances. The class attribute has 2 distinct values. Ovarian cancer dataset consists of 15,155 features and 253 instances. The class attribute has 2 distinct values. Leukemia cancer dataset consists of 7,143 features and 49 instances. The class attribute has 8 distinct values.

For performance evaluation, this paper applies the area under the curve (AUC), precision and recall as indicators. AUC is calculated from receiver operating characteristic curve (ROC curve). The ROC curve is a graph of true positive rate (TPR) against

false positive rate (FPR). The classification models were built by dividing 70% training and 30% testing partitions randomly. A ROC graph is plotted using TPR against the FPR for evaluating the accuracy. Precision is value comes from ratio of the TPR and the summation of TPR and FPR. Recall value is calculated from TPR against the summation of TPR and false negative rate (FNR).

4.2 Experimental Result

The experiments were performed to evaluate the predictive performance of the proposed ensemble algorithm against the existing ensemble algorithm from previous research name "Ensemble of Feature Selection Techniques for High Dimensional Data" and individual rankers. In order to evaluate the performance of ranker, we selected the top-k features (represented by 20%, 40%, 60% and 80% of features in dataset), then the selected features will be applied to classification models that are k-nearest neighbor (KNN), naïve Bayes (NB), random forest (RF), logistic regression (LR), support vector machines (SVM) and decision trees. The classification models used in this paper tests with 10-fold cross validation. The performance of classification model is evaluated by using the average of AUC, precision and recall metrics. The results from the experiment are illustrated in Figs. 2, 3 and 4 for the average of AUC value, Figs. 5, 6 and 7 for average precision value and Figs. 8, 9 and 10 for average recall value.

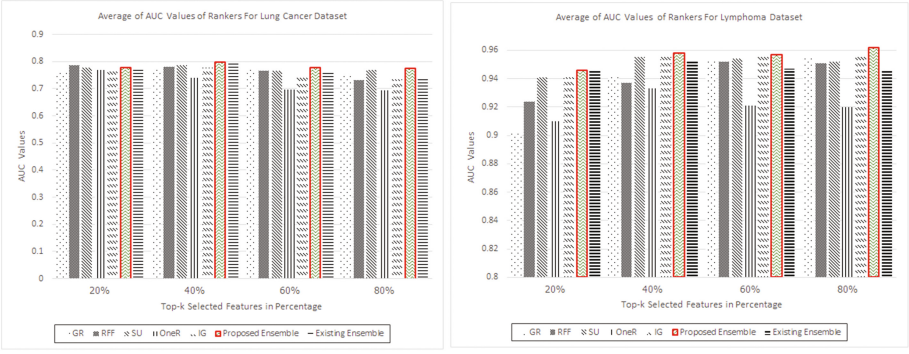


Fig. 2. The average of AUC values for lung cancer and lymphoma datasets.

From Fig. 2, the results of lung cancer dataset show that the average AUC value of our ensemble algorithm is higher than all individual and existing rankers for top 40%, 60% and 80% features. However for top 20% features, the average AUC value of reliefF (AUC = 0.787) outperforms than others due to the number of features are not much. Symmetrical uncertainty, the existing ensemble and our proposed ensemble have the same performance with 0.778 for top 20% features. The ranker that has the worst performance in all top-k features for lung cancer dataset is OneR. The proposed ensemble has the best average AUC value for top 40% features with 0.799. For

lymphoma dataset, the results show that the proposed ensemble algorithm tends to have more performance with high number of features. The performance of proposed ensemble is higher than all individual and the existing rankers for all top-k selected features (20%, 40%, 60% and 80%). Individual rankers that have performance closed to the proposed ensemble algorithm are information gain and symmetrical uncertainty. Similar with lung cancer dataset, the ranker that has the worst performance in all top-k features for lymphoma dataset is OneR. The proposed ensemble algorithm has the best performance for top 80% features with 0.962.

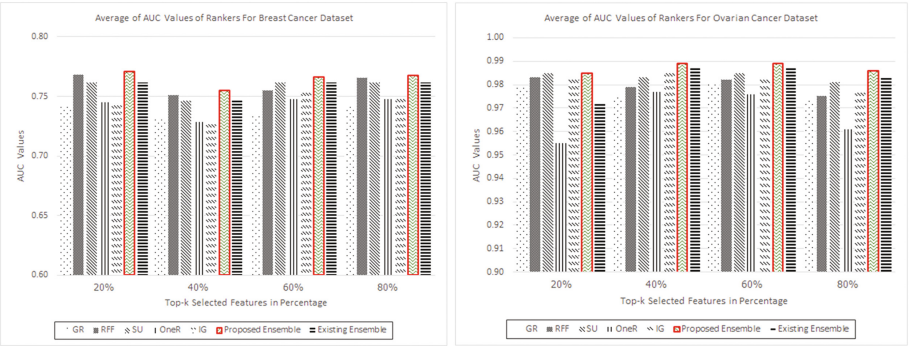


Fig. 3. The average of AUC values for breast cancer and ovarian cancer datasets.

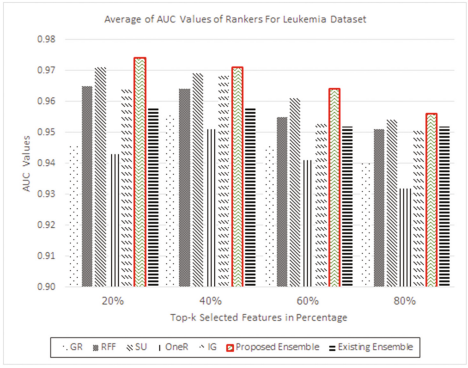


Fig. 4. The average of AUC values for leukemia dataset.

From Figs. 3 and 4, the bar charts show that the average AUC value of proposed ensemble algorithm is higher than all feature ranking techniques for all top-k selected features. The rankers that have performance closed to the proposed ensemble algorithm are symmetrical uncertainty for breast cancer and leukemia datasets, and the existing ensemble algorithm for ovarian cancer dataset. The ranker that has the worst performance in both ovarian cancer and leukemia datasets is OneR and for breast cancer

dataset is gain ratio. The proposed ensemble algorithm has the best performance for breast cancer dataset at top 20% features with 0.771, for ovarian cancer dataset is top 40% features with 0.989 and for leukemia dataset is top 20% with 0.974.

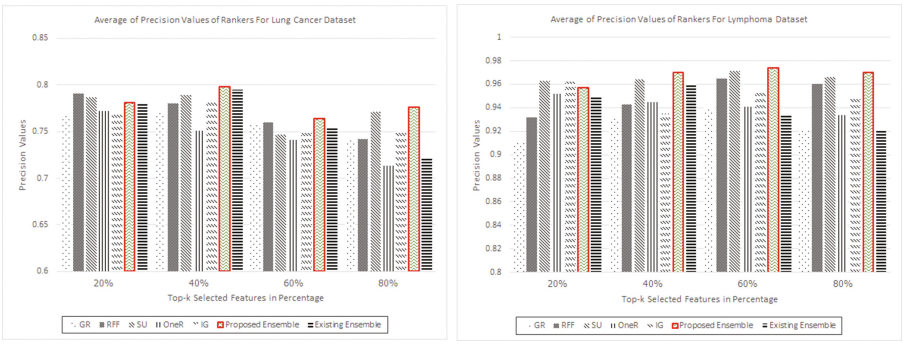


Fig. 5. The average of precision values for lung cancer and lymphoma datasets.

From Fig. 5, the results of lung cancer and lymphoma datasets show that the average precision value of the proposed ensemble algorithm is higher than all individual and existing rankers for top 40%, 60% and 80% features. But for top 20% features, the precision value of reliefF is better than others with average precision value = 0.791 for lung cancer dataset. For lymphoma dataset, symmetrical uncertainty is better than others with average of precision value = 0963. The ranker that has the worst performance for lung cancer dataset is OneR and for lymphoma dataset is gain ratio. The proposed ensemble algorithm has the best performance for lung cancer dataset at top 40% features with 0.798 and for lymphoma dataset is top 60% features with 0.972.

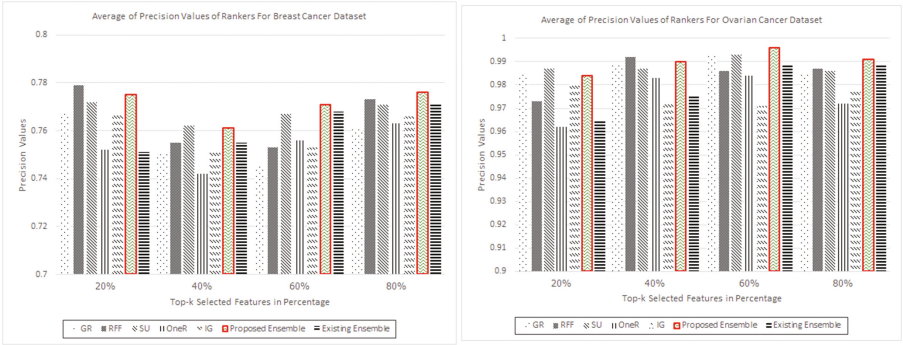


Fig. 6. The average of precision values for breast cancer and ovarian cancer datasets.

From Fig. 6, the results of breast cancer dataset show that the average precision value of our ensemble algorithm is higher than all individual and existing rankers for top 60% and 80% features. But for top 20% features, the precision value of reliefF outperforms than others with average precision value = 0.779. The precision value of symmetrical uncertainty is higher than others with average precision value = 0.762 for top 40% features. The ranker that has the worst performance for breast cancer dataset is OneR. The proposed ensemble has the best average precision value for top 80% features with 0.776. For ovarian cancer dataset, the results show that the average precision value of the proposed ensemble algorithm is higher than all individual and existing ranker for top 60% and 80% features. But for top 20% features, symmetrical uncertainty has the most average of precision value with 0.987. For top 40% of features, reliefF has the average precision value higher than others with 0.992. The ranker that has the worst performance for ovarian cancer dataset is OneR. The proposed ensemble has the best average of precision value for top 60% features with 0.996.

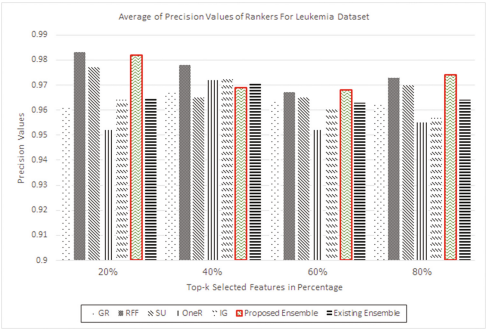


Fig. 7. The average of precision values for leukemia dataset.

From Fig. 7, the results of leukemia dataset show that the average precision value of proposed ensemble algorithm is higher than all individual and existing rankers for top 60% and 80% features. However for top 20% and 40% features, the precision value of reliefF outperforms than others with average precision value = 0.983 and 0.978. The ranker that has the worst performance for leukemia dataset is OneR. The proposed ensemble has the best average precision value for top 20% features with 0.981.

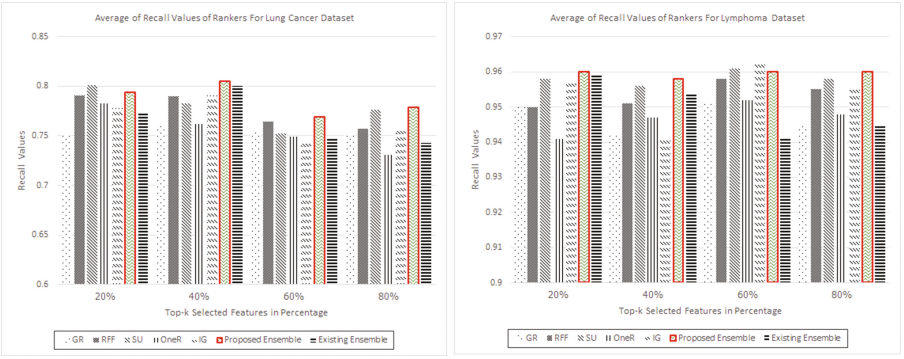


Fig. 8. The average of recall values for lung cancer and lymphoma datasets.

From Fig. 8, the results of lung cancer dataset show that the average recall value of our ensemble algorithm is higher than all individual and existing rankers for top 40%, 60% and 80% features. But for top 20% features, the recall value of symmetrical uncertainty is higher than others with average recall value = 0.801. The ranker that has the worst performance for lung cancer dataset is OneR. The proposed ensemble has the best average recall value for top 40% features with 0.805. For lymphoma dataset, the results show that the average of recall value of proposed ensemble algorithm is better than all individual and existing rankers for top 20%, 40% and 80% feature. But for top 40% features, the recall value of information gain is higher than others with average recall value = 0.963. The ranker that has the worst performance for lung cancer dataset is OneR. The proposed ensemble has the best average recall value for top 60% features with 0.961.

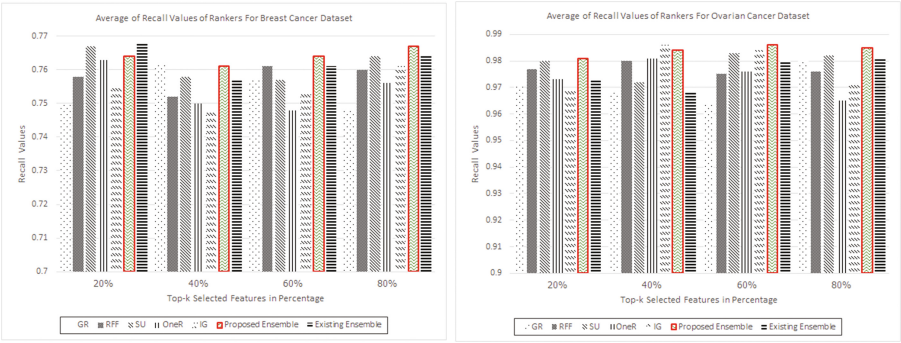


Fig. 9. The average of recall values for breast cancer and ovarian cancer datasets.

From Fig. 9, the results of breast cancer dataset show that the average recall value of proposed ensemble algorithm is higher than all individual and existing rankers for top 40%, 60% and 80% features. But for top 20% features, the recall value of existing ranker is higher than others with average recall value = 0.768. The ranker that has the worst performance for breast cancer dataset is gain ratio. The proposed ensemble has the best average recall value for top 80% features with 0.767. For ovarian cancer dataset, the results show that the average recall value of proposed ensemble algorithm is better than all individual and existing rankers for top 20%, 60% and 80% features. However for top 40% features, information gain has the most average recall value with 0.986. The ranker that has the worst performance for ovarian cancer dataset is gain ratio. The proposed ensemble has the best average recall value for top 60% features with 0.988.

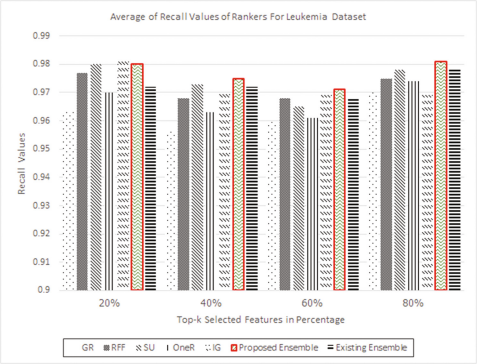


Fig. 10. The average of recall values for leukemia dataset.

From Fig. 10, the results of leukemia dataset show that the average recall value of proposed ensemble algorithm is higher than all individual and existing rankers for top 40%, 60% and 80% features. But for top 20% features, the recall value of information gain is higher than others with average recall value = 0.983. The ranker that has the worst performance for leukemia dataset is gain ratio. The proposed ensemble has the best average recall value for top 80% features with 0.981.

5 Conclusion and Future Work

In this paper, we have reviewed five feature ranking techniques consists of information gain, gain ratio, symmetrical uncertainty, relief and oneR attribute evaluation. Then we introduce an ensemble of multiple feature ranking techniques by considering ranker priority for feature selection. The proposed ensemble algorithm improves the accuracy performance of classification models. The proposed ensemble algorithm is performed by considering an order, frequency of single feature in all ranking lists and the priority for each feature ranking techniques to generate a final ranking list. For performance evaluation, AUC, precision and recall values from classification models are applied. The classification models that were considered in this paper are k-nearest neighbor, naïve bayes, random forest, logistic regression, support vector machines and decision trees. The datasets for the experiment are lung cancer, lymphoma, breast cancer, ovarian cancer and leukemia datasets from Kent Ridge bio-medical data and Machine Learning data repository.

The experiment shows that the proposed ensemble algorithm of multiple feature ranking techniques by considering ranker priority is tend to be more effective than an individual feature ranking techniques and the existing ensemble of multiple features ranking techniques for high dimensional dataset.

For future works, various sets of data with a large number of features and instances are considered. The other evaluation metrics will be explored.

References

1. Silwattananusarn, T., Kanarkard, W., Tuamsuk, K.: Enhanced classification accuracy for cardiotocogram data with ensemble feature selection and classifier ensemble. J. Comput. Commun. **4**, 20–35 (2016)
2. Zilin, Z., Hongjun, Z., Rui, Z., Youliang, Z.: Hybrid feature selection method based on rough conditional mutual information and naïve Bayesian classifier. ISRN Appl. Math. (2014)
3. Kashif, J., Haroon, A.B., Mehreen, S.: Feature selection based on class-dependent densities for high dimensional binary data. IEEE Trans. Knowl. Data Eng. **24**(3), 465–475 (2012)
4. Qinbao, S., Jingjie, N., Guangtao, W.: A fast clustering-based feature subset selection algorithm for high-dimensional data. IEEE Trans. Knowl. Data Eng. **25**(1), 1–14 (2013)
5. Aceto, G., Dainotti, A., Donato, W., Pescapé, A.: PortLoad: taking the best of two worlds in traffic classification. In: IEEE INFOCOM 2010 - WiP Track (2010)
6. Wang, H., Taghi, M., Gao, K.: High-dimensional software engineering data and feature selection. In: 21st IEEE International Conference on Tools with Artificial Intelligence, pp. 83–90 (2009)
7. Kexin, Z., Jian,Y.: A cluster-based sequential feature selection algorithm. In: 9th International Conference on Natural Computation, pp. 848–852 (2013)
8. Sutha, K., Temilselvi, J.: A review of feature selection algorithms for data mining techniques. Int. J. Comput. Sci. Eng. 10–13 (2016)
9. Koller, D., Sahami, M.: Toward optimal feature selection. In: International Conference on Machine Learning, pp. 284–292 (1996)
10. Fahad, A., Tari, Z., Khalil, I., Habib, I., Alnuweiri, H.: Toward an efficient and scalable feature selection approach for internet traffic classification. In: IEEE Computer Network Conference, vol. 57, pp. 2040–2057 (2013)
11. Osanaiye, O., Raymond, C., Dehghantanha, A., Zheng, X., Mqhele, D.: Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing. EURASIP J. Wirel. Commun. Netw. (2016)
12. Sujatha, M., Devi, L.: Feature selection techniques using for high dimensional data in machine learning. Int. J. Eng. Res. Technol. **2**(9), 4–9 (2013)
13. Zhao, Z., Liu, H.: On similarity preserving feature selection. IEEE Trans. Knowl. Data Eng. **25**(3), 619–632 (2013)
14. Castellanos, G., Delgado, E., Daza, G., Sanchez, L.G., Suarez, J.F.: Feature selection in pathology detection using hybrid multidimensional analysis. In: International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 5503–5506 (2006)
15. Vege, S.H.: Ensemble of feature selection techniques for high dimensional data. Master thesis and specialist projects, p. 1164 (2012)
16. Szabó, G., Veres, A., Malomsoky, S., Gódor, I., Molnár, S.: Traffic classification over Gbit speed with commodity hardware. J. Syst. Softw. (2010)
17. De Donato, W., Pescapé, A., Dainotti, A.: Traffic identification engine: an open platform for traffic classification. IEEE Netw. **28**(2), 56–64 (2014)
18. Wang, Y.: Fisher scoring: an interpolation family and its Monte Carlo implementations. Comput. Stat. Data Anal. 1744–1775 (2010)
19. Dahiya, S., Singh, N.P.: A rank aggregation algorithm for ensemble of multiple feature selection techniques in credit risk evaluation. Int. J. Adv. Res. Artif. Intell. **5**(3) (2016)
20. Weka Data Mining Software. http://www.cs.waikato.ac.nz/ml/weka/
21. Kent Ridge Bio-medical Data Repository. http://datam.i2r.a-star.edu.sg/datasets/krbd/
22. Machine Learning Data Repository. http://mldata.org/repository/data/

User Independency of SSVEP Based Brain Computer Interface Using ANN Classifier: Statistical Approach

Md. Kamrul Hasan(✉), Md. Samiul H. Sunny, Shifat Hossain, and Mohiuddin Ahmad

Department of Electrical and Electronic Engineering (EEE), Khulna University of Engineering & Technology (KUET), Khulna 9203, Bangladesh
{m.k.hasan, ahmad}@eee.kuet.ac.bd, mdshs31@gmail.com, shifataccount@gmail.com

Abstract. BCIs, which elaborated as Brain-computer Interface that use brain responses to control the BCI paradigms. These brain responses are measured using Electroencephalographic signal along the scalp of the subjects. However, the less variability of EEG signal from the subjects make the BCI paradigms user independent. In this research, we simply analyze the user independency of SSVEP based EEG signal that makes a conclusion inter subject's variability of BCI users. To accomplish the research goal, SSVEP based EEG signal extract from both different subjects and different stimulation conditions and a features vector is formed to compare each subject's variability. Artificial Neural Network classifier is used to determine the deviation and regression of deviation of each features vectors. From the heatmap and classifier, it is found that the used independency of the EEG signal is less that means that less variability of EEG. That ensures the user independent BCI paradigms with high transfer rate of the bits.

Keywords: Brain-computer Interface (BCI) · BCI paradigms · Electroencephalographic (EEG) · Steady-state Visual Evoked Potential (SSVEP) · Artificial Neural Network (ANN) · EEG classifier

1 Introduction

Brain-computer Interface is a technology in which a non-muscular channel can be created between a brain and a computer through which a human can control computer, peripheral or other electronics devices as well as advanced prosthetic devices by sending messages and commands to the external world with the extracted action potentials of EEG signal. It enables the user to send information or command to a communication system not through the brain's peripheral neuromuscular pathway as conventional output pathway [1]. To improve the quality of life of the severely disabled people BCIs are primarily developed [2–4]. Among various types of BCIs systems, the major categories are invasive system, partially invasive system and Non-invasive system [5]. In this system, arrays of electrodes implanted in the brain to record the action potentials. In partial invasive technique for BCIs, ECoG is used for feature in

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_6

BCIs applications. But in non-invasive technique the features for BCIs applications are extracted from the, MEG or fMRI signals which are taken from the scalp. All of these BCIs signal acquisition techniques are shown in Fig. 1(a).

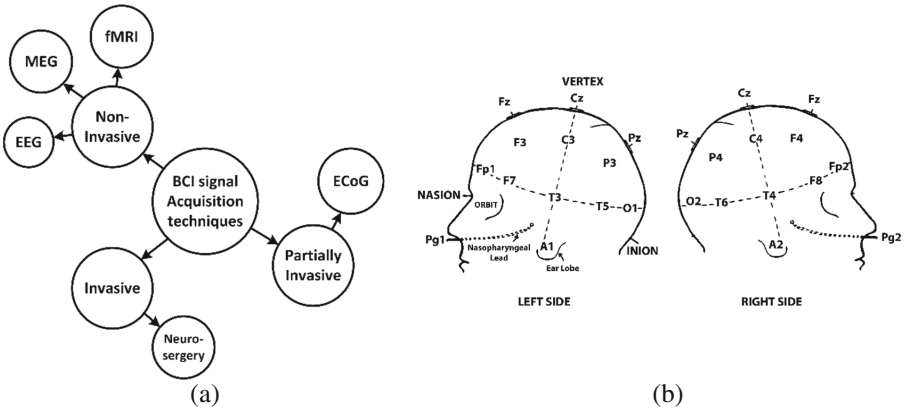


Fig. 1. (a) Pictorial presentations of BCI's signal acquisition techniques (b) Electrode (10–20) positioning system for EEG extraction using MP-36

Because of higher SNR in invasive BCI highly skilled operator is required to implement the sensors and avoid the other side effects in the result. Featuring low set-up cost and increased concerns for the personal healthcare non-invasive BCIs have become more popular today. Recently, the BCIs that are based on EEG signals have been used for controlling devices to help bring mobility back to some severely disabled people [6–8]. The recording magnitude of EEG is quite small and it is measured in microvolts with the help of electrodes from the scalp. In Fig. 1(b) shows the 10–20 system to describe and apply the location of scalp electrodes in the context of an EEG test of experiment. The brain's output channel is a certain signal in the EEG and the generation of this signal does not depend on the orientation of the eyes, but on the user's intent. Recorded brain activity from user, signal processing, commands administered by the BCI system and operating protocol, which determines the timing of operation are the main elements of BCI system [9–11]. Periodic evoked potentials induced by repetitive visual stimulation is known as SSVEP. Several SSVEP-based BCI systems have already been proposed [12–15]. Pre-processing the EEG signals using spectral and time domain filters a novel recognition method based on SSVEP is proposed in [13] in order to enhance the signal to noise ratio. To invoke SSVEP response a virtual reality head-mounted display is used for immersive brain computer interface in [13]. Navigation in a virtual environment combining the BCI and Convolutional NN is presented in [14] which relies on Steady-State Visually Evoked Potential. High-speed communication between human brain and external peripherals in hybrid frequency and phase coding methods for multi-target BCI is illustrated in [15]. Using a plain stimulus over a checkerboard stimulus results showed a statistically significant 9.26% average increase in SSVEP classification in [16].

2 Proposed Methodology

The overall proposed methodologies are described in several sub-section below:

2.1 Overall Frame Work of the Research

Basic Components for analysing SSVEP based user independency in BCIs starts from EEG signal acquisition, followed a signal pre-processing module includes Noise filtering. Next, a feature extraction module transforms the signals into useful features which are processed into a classification algorithm with optimized training for classification tasks. The outputs can be mapped for the BCI commands. For inspecting and analysing the user indecency, heatmap can be generated using feature matrix is used. The overall components of methodology of the research is shown in Fig. 2.

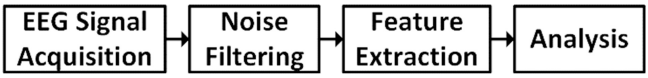


Fig. 2. Components of the methodology of frame work

2.2 EEG Data Acquisition

MP-36 EEG Acquisition System is used to collect raw EEG signal from the scalp of the subject. To extract SSVEP based EEG, a Repetitive Visual Stimulus (RVS) of different shapes, sizes and colors have been used is rendered on a computer screen by flashing different shapes, sizes and colors. This RVS stimulator is placed in front of the subjects at the visual distances. Figure 3 shows the environment of signal acquisition form the subjects.

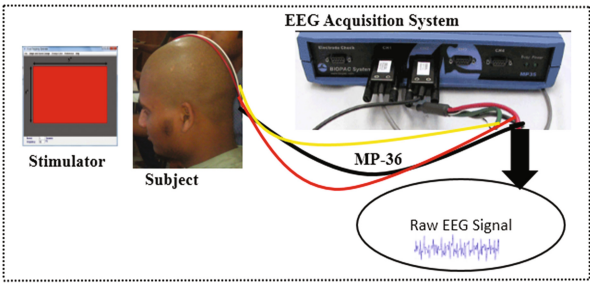


Fig. 3. Subjects placement for EEG signal extraction using MP-36

2.3 Noise Cancellations from Raw EEG Data

EEG signals are often mixed with artifacts which include EMG, EOG, ECG, and electrical line noise and other external and internal noises. These artifacts are being

added in the EEG signal by unknown bodily dynamics, which has a non-linear property. In our research, these noise reduction is accomplished using adaptive noise cancellation (ANC) based on ANFIS as shown in Fig. 4. In this system ANFIS is used to estimate the non-linear bodily dynamics. The noise is then estimated using the estimated function by ANFIS taking close to pure EMG, EOG signals and line noise as input. This signal is subtracted from the artifact affected signal to filter it from noise.

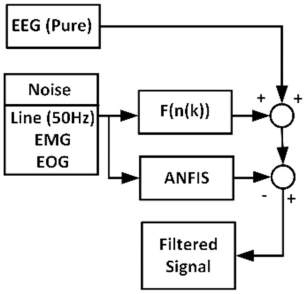


Fig. 4. Raw EEG Filtering and noise cancellation using ANC implemented in ANFIS

2.4 Features Extraction from Raw EEG Signal

Features of the EEG signal is the distinctive attribute by which that signal can be identified that convey information about that particular signal. In our research, raw EEG is collected form the different subjects using different shapes, sizes and colors of the RVS. The features mentioned in the Fig. 5 are extracted in MATLAB environments. After that, using these features, Neural Network is trained which is used to classify the EEG signal to determine the user independency.

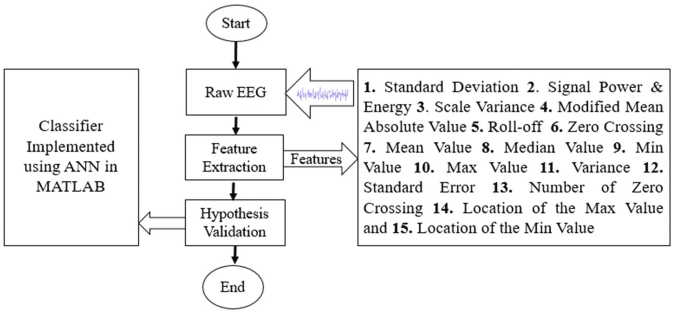


Fig. 5. Representations of features extraction and hypothesis validation

2.5 User Independency Analysis

For the user independency analysis, the work flow is shown in Fig. 6. A matrix is created from the features which are extracted from the EEG signals of different subjects. Normalization is applied to the feature matrix and the deviation matrix is calculated from the normalized feature matrix. In heatmap representation using color map it is observed that user to user feature deviation in almost all the features remains within 30%.

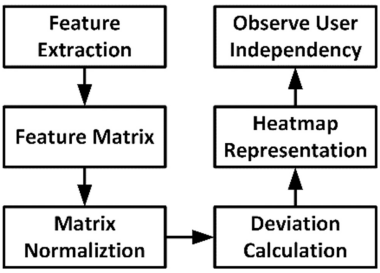


Fig. 6. Flow chart for user independency validation

3 Mathematical Background of Features

The raw EEG from the different subjects are analyzed for extracting various features as previously described. For feature extraction, various mathematical expression regarding signal processing is used. Energy of the signal is calculated at $-\infty$ to ∞ which indicates the strength of the EEG signal. Similarly, for period T, power is calculated when the signal is represented by x(t). The mathematical formulation is given below

$$Energy, E_{EEG} = \int_{-\infty}^{\infty} x^2(t)\, dt \tag{1}$$

$$Power, P_{EEG} = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} x^2(t)\, dt \tag{2}$$

Scale variance of EEG signal is a feature of object that do not change if scales of length, energy, or other variables, are multiplied by a common factor. The wavelet co-efficient scale variance which is also known as log variance is calculated by

$$Scale\ Variance, SV_{EEG} = \log(var(x_{EEG}))/\log 2 \tag{3}$$

The rate of loss or attenuation of a signal beyond a certain frequency is represented by the feature roll off and it is also a measure of spectral shape. Assuming that 85% of the magnitude distribution of the spectrum is intense roll off is derived from

$$Roll-off, R_{EEG} = 0.85 \times \sum_{n=1}^{n/2} |x_{EEG}| \tag{4}$$

The term standard deviation is used to determine the amount of quantity of the members of a group differ from the mean value for that group and commonly used as a feature of EEG signal. When the number of samples is N, it is obtained by

$$Standard\ Deviation, \sigma_{EEG} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} (x_{EEG,i} - \mu)^2} \tag{5}$$

For observing the strength of correlation between two or more random variables covariance is calculated. For two random variants X_{EEG} *and* Y_{EEG} covariance is determined from the following formula

$$Covariance = \sum_{i=1}^{N} \frac{\left(x_{EEG,i} - \bar{x}\right)\left(y_{EEG,i} - \bar{y}\right)}{N} \tag{6}$$

Using weighting function and taking the average of the absolute EEG signal modified mean absolute value is calculated. In other words, this feature is the moving average of full-wave rectified EEG signal. Here w_n is the weighting function.

$$MMAV_{EEG} = \frac{1}{N}\sum_{n=1}^{N} (w_n \times |x_{EEG}|), \quad \text{Where } w_n = \begin{cases} 1.0 & 0.25N \le n \le 0.75N \\ 0.5 & Otherwise \end{cases} \tag{7}$$

As a feature for estimating the frequency domain properties the number of zero crossing, which indicates the number of times the amplitudes of EEG signal crosses the zero y-axis is calculated as

$$\begin{aligned} ZC_{EEG} &= \sum_{n=1}^{N} sgn(x_n \times x_{n-1}) \cap |x_n - x_{n-1}| \ge Threshold, Where\ sgn(x) \\ &= \begin{cases} 1 & x \ge Threshold \\ 0 & Otherwise \end{cases} \end{aligned} \tag{8}$$

A features matrix is formed from the extracted features which like Eq. 1. Every element in this matrix denoted as a features of extracted raw EEG signal. This features matrix is used for the verifications of the user independency of the BCIs paradigms.

$$Matrix_{Features} = \begin{bmatrix} \sigma\_s1\sigma_{EEG} & P_{x,EEG} & Mode_{EEG} & Min\_Value_{EEG} \\ R\_s1R_{EEG} & ee & MMAV_{EEG} & Min_{Location_{EEG}} \\ \bar{X}_{EEG} & ZC_{EEG} & Median_{EEG} & Max\_Value_{EEG} \\ SV_{EEG} & COV_{EEG} & \sigma_{Error_{EEG}} & Max_{Location_{EEG}} \end{bmatrix} \tag{9}$$

4 Results and Discussion

Table 1 shows the numerical presentations of the values of different features that are extracted from the raw EEG signal for the analysis of user independency of BCIs paradigms. In table, each features are extracted from three different subjects which are stimulated visually by different shapes, sizes and colors of the RVS for SSVEP based EEG signal extraction. Deviation of one feature matrix from another feature matrix is the variation of matrix elements from other matrix elements.

Table 1. Statistical value of different features of EEG signal for three different subjects

SL #	Features of EEG	Sub-1	Sub-2	Sub-2
01	Standard deviation	05.4685	06.2104	06.8012
02	Covariance	29.9044	38.5670	23.0515
03	Mean value	40.7923	05.3709	04.1783
04	Median value	03.6560	03.8510	03.5522
05	Variance	29.9044	38.5670	23.0515
06	Mode	−03.7415	−04.1748	−02.7771
07	Roll off ($\times 10^5$)	09.8571	01.5494	01.1377
08	Signal power	41.2925	55.8691	27.9251
09	Signal energy ($\times 10^5$)	01.9341	04.4332	01.5778
10	Min value	−29.5593	−32.3853	−23.4436
11	Max value	16.9922	09.3018	26.8372
12	MMAV	03.3363	03.5976	03.1505
13	Scale variance	04.9023	05.2693	04.5268
14	Standard error	00.1566	00.1366	00.1252
15	# of zero crossing	313.000	479.000	424.000
16	Location of max value	161.000	6359.00	2421.00

The deviation matrixes are form for the justifications of the user independency of the BCIs paradigms. The lower the deviation the lower the variations which results the more user independency of BCIs paradigms.

$$Deviation\ Matrix_{1st\ Case} = \begin{bmatrix} 0.1356 & 0.3530 & 0.1158 & 0.0956 \\ 0.5719 & 1.2921 & 0.0783 & 67.6800 \\ 0.1207 & 0.5304 & 0.0534 & 0.4526 \\ 0.0749 & 0.2897 & 0.1275 & 38.4969 \end{bmatrix}$$

$$Deviation\ Matrix_{2nd\ Case} = \begin{bmatrix} 0.2269 & 0.5002 & 1.6652 & 0.2761 \\ 0.2658 & 0.6441 & 0.1243 & 0.9536 \\ 0.2221 & 0.1148 & 0.0777 & 1.8852 \\ 0.1409 & 0.4023 & 0.0838 & 0.6193 \end{bmatrix}$$

$$Deviation\ Matrix_{3rd\ Case} = \begin{bmatrix} 0.1390 & 0.4786 & 2.3473 & 0.2609 \\ 0.1336 & 0.2258 & 0.0590 & 0.6865 \\ 0.1470 & 0.2618 & 0.0292 & 0.3668 \\ 0.0830 & 0.2973 & 0.2509 & 0.9335 \end{bmatrix}$$

From the deviation matrixes heatmap is generated where the individual values contained in the deviation matrix are represented as colors for the individual subject in Fig. 7(a–c). Through these color heatmap an immediate visual summary of user independency is depicted except two or three features. Subject wise over all comparison of feature deviation Fig. 7(d) also proves the minimum deviation of the features.

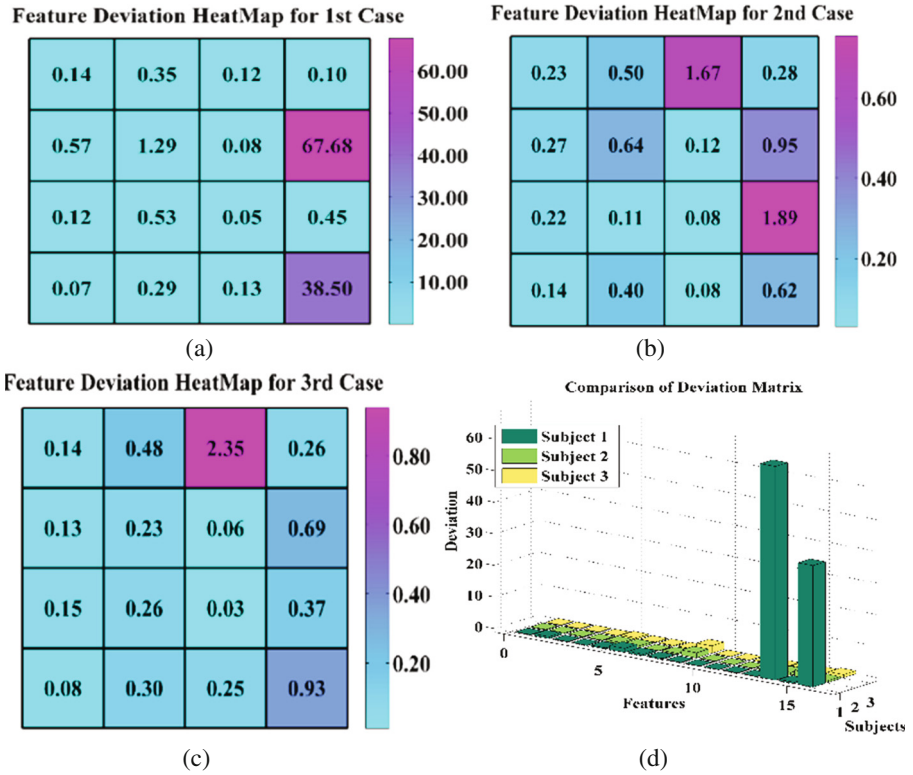


Fig. 7. (a, b, c) Features deviation heatmap for Subject 1, Subject 2 and Subject 3 respectively (d) Comparison of deviation matrixes.

Due to several capabilities such as off-line training, Nonlinear Mapping, less computational effort and compact solution for multi-variable problem ANN algorithm is chosen for the classification which can be high responsive and robust in operation. A feedforward NN is taken for training and tan sigmoid is used as a transfer function. Levenberg-Marquardt is used as training algorithm and MSE is set to check the

Performance. One hidden layer is used in ANN model with 33 neurons. In Fig. 8(a) the quality of the network is validation performance which is 6.8453×10^{-08} when the goal is zero and iteration number 1000. In Fig. 8(b), the training state of the network at the point of the best performance is expressed through the gradient, μ and the validation check. From the training state, it is found that the value of the gradient is 6.8139×10^{-10} which means it reaches the bottom of the local minimum of the goal function. The mu value is 10^{-13}. There is no value fail up to the best validation performance iteration means the perfect training of the network. Almost all the data in the zero-error line indicates the training accuracy. Figure 8(d) is the regression plot of the network training. The R value represents the relation between the outputs and the target data of the network training. From the plot, it is observed that the value of R is 0.94823 which implies that there is 94.82% linear relationship between the outputs and the target. So, the training is executed successfully. Dividing data before training, regularization and retraining, the possible overfitting is avoided. For the purpose of testing the trained network, the network is simulated with the sim command to justify whether it is working properly or not.

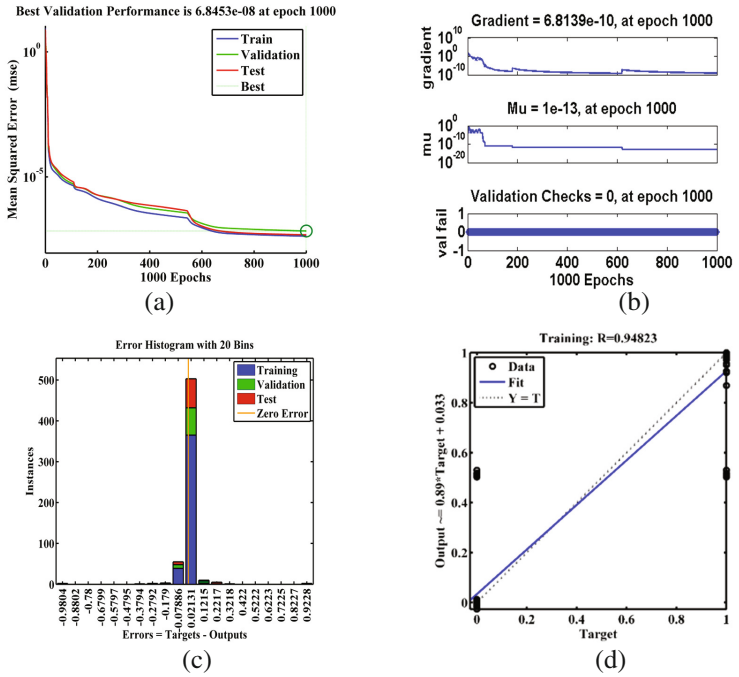


Fig. 8. (a) Training performance of the network (b) Training state during the training (c) Error histogram of the training (d) Regression curve of the training

5 Conclusion and Future Work

The user independency of the BCI paradigms is the primary concern of the EEG based HCIs. The less the variability of the EEG signal from the subjects for different stimulating conditions the more the acceptance of the paradigms because it makes the uniqueness of the BCI paradigms. In this research work, the user independency of SSVEP based BCI paradigms were analyzed. For that, brain was stimulated using different shapes, sizes and colors of the RVS to get diverse SSVEP based EEG signal. From the independency analysis, it was found that the variation of the EEG patterns and different features of the features vector is less and that is in acceptable limit. So, it can be concluded that EEG based BCI can be used as a suitable BCI paradigms for the physically disable persons. Advanced signal processing schemes may be used to investigate the EEG independency more sophistically.

References

1. Chiu, C.Y., Chen, C.Y., Lin, Y.Y., Chen, S.A., Lin, C.T.: Using a novel LDA-ensemble framework to classification of motor imagery tasks for brain-computer interface applications. Int. Comput. Symp. (ICS) **1**, 136–142 (2014)
2. Ortner, R., Lugo, Z., Noirhomme, Q., Laureys, S., Guger, C.: A tactile brain-computer interface for severely disabled patients. In: IEEE Haptics Symposium (HAPTICS), USA, pp. 235–237 (2014)
3. Wu, Y., Li, M., Wang, J.: Toward a hybrid brain-computer interface based on repetitive visual stimuli with missing events. J. Neuroeng. Rehabil. **13**, 66 (2016)
4. Kaufmann, T., Herweg, A., Kübler, A.: Toward brain computer interface based wheelchair control utilizing tactually-evoked event related potentials. J. Neuroeng. Rehabil. **11**, 7 (2014)
5. Kerous, B., Liarokapis, F.: Brain-computer interfaces—a survey on interactive virtual environments. In: 8th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES), Barcelona, pp. 1–4 (2016)
6. Li, Y., Yu, T.: EEG-based hybrid BCIs and their applications. In: 3rd International Winter Conference on Brain-Computer Interface (BCI), Sabuk, pp. 1–4 (2015)
7. Xiao, D., Zhang, W.: Electroencephalogram based brain concentration and its human computer interface application. In: IEEE International Conference on Computer and Communications (ICCC), Chengdu, pp. 21–24 (2015)
8. Robinson, N., Vinod, A.P.: Bi-directional imagined hand movement classification using low cost EEG-based BCI. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC), Kowloon, pp. 3134–3139 (2015)
9. Gaur, P., Pachori, R.B., Wang, H., Prasad, G.: A multivariate empirical mode decomposition based filtering for subject independent BCI. In: 27th Irish Signals and Systems Conference (ISSC), Londonderry, pp. 1–7 (2016)
10. Hasan, M.K., Rusho, R.Z., Ahmad, M.: A direct noninvasive brain interface with computer based on steady-state visual-evoked potential with high transfer rates. In: International Conference on Advances in Electrical Engineering (ICAEE), IUB, Bangladesh (2013)

11. Abbasi, M.A., Gaume, A., Francis, N., Dreyfus, G., Vialatte, F.B.: Fast calibration of a thirteen-command BCI by simulating SSVEPs from trains of transient VEPs—towards time-domain SSVEP BCI paradigms. In: 7th International IEEE/EMBS Conference on Neural Engineering (NER), Montpellier, pp. 186–189 (2015)
12. Yehia, A. G., Eldawlatly, S., Taher, M.: Principal component analysis-based spectral recognition for SSVEP-based brain-computer interfaces. In: 10th International Conference on Computer Engineering and Systems (ICCES), Cairo, pp. 410–415 (2015)
13. Koo, B., Lee, H.G., Nam, Y., Choi, S.: Immersive BCI with SSVEP in VR head-mounted display. In: 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, pp. 1103–1106 (2015)
14. Bevilacqua, V., Tattoli, G., Buongiorno, D., Loconsole, C., Leonardis, D., Barsotti, M., Frisoli, A., Bergamasco, M.: A novel BCI-SSVEP based approach for control of walking in virtual environment using a convolutional neural network. In: International Joint Conference on Neural Networks (IJCNN), Beijing, pp. 4121–4128 (2014)
15. Chen, X., Wang, Y., Nakanishi, M., Jung, T.P., Gao, X.: Hybrid frequency and phase coding for a high-speed SSVEP-based BCI speller. In: 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Chicago, IL, pp. 3993–3996 (2014)
16. Zerafa, R., Camilleri, T., Falzon, O., Camilleri, K.P.: Comparison of plain and checkerboard stimuli for brain computer interfaces based on steady state visual evoked potentials. In: 6th International IEEE/EMBS Conference on Neural Engineering (NER), San Diego, CA, pp. 33–36 (2013)

Analysis of Two-Missing-Observation 4 × 4 Latin Squares Using the Exact Approach

Kittiwat Sirikasemsuk[1(✉)] and Kanogkan Leerojanaprapa[2]

[1] Department of Industrial Engineering, Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
kittiwat.sirikasemsuk@gmail.com
[2] Statistics Department, Faculty of Science,
King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
kanogkan.le@kmitl.ac.th

Abstract. This research deals with the analysis of incomplete Latin square designs using the exact approach. Specifically, the study investigated the 4 × 4 Latin square designs with two missing observations without replication. In the research, the general regression significance test (i.e. the exact approach) was used to derive the estimation formulae of fitted parameters for the full and reduced linear statistical models, thereby simplifying the calculation process. In addition, the proposed exact approach-based formulae facilitate the determination of the treatment sum of squares and the error sum of squares, both of which are subsequently employed in the analysis of variance (ANOVA).

Keywords: Design of experiment · Latin square · Incomplete Latin square · Two missing observations · General regression significance test

1 Introduction

Classical experimental designs are commonly utilized in multidisciplinary and interdisciplinary fields of science and engineering, e.g. agriculture, biology, ecology, industry, pharmacology, and product designs. In addition, there exist several other experimental design strategies to acquire specific crucial factors and the derivatives of optimization and robustness. According to Montgomery [1] and Mead et al. [2], three basic elements of classical experimental designs are replication, blocking and randomization.

In the presence of controllable and known nuisance factors, the blocking technique is adopted to separate their effects from the experimental error. For instance, in the blocking-technique designs in which only one main or potential factor with multiple treatments is under consideration, a randomized complete block design (RCBD) would be used for one nuisance factor; a Latin square design or a Latin rectangle design (see [3]) for two nuisance factors; and a Graecco-Latin square design for three nuisance factors.

In the Latin square design, the Latin letters represent the levels of the potential factor and the number of rows and columns is identical to the number of blocks of all two nuisance factors. In addition, a crossed double-blocking system between two nuisance factors exists. Typically, the Latin letters in the same columns (or the same rows) must not be replicated, and the number of blocks in each of the two nuisance

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_7

factors must be equal to the number of treatments (or Latin letters) of the potential factor. In Youden [4], the author proposed the Youden square design, which is a distinct Latin rectangle design, in which the number of blocks on one side is greater than the other side's and the number of treatments (Latin letters) is equal to the number of blocks of the former.

Experimenters or statistical data analysts frequently come across problems of incomplete experimental data. For example, some irregular data are caused by an improper control over some variables, or test resources are missing or not enough for running experiments. Hence, some data cannot be correctly observed. In general, the incidence of missing observations could result in unfavorable analysis due to the unbalanced and non-orthogonal characteristics of these designs. Frequently, there was a lack of definite formula for the ANOVA table for the cases of the incomplete-data experiment designs. Greater efforts are required of the experimenters to carry out the analysis of variance. To address the missing-value issue, Montgomery [1] has proposed two comprehensive procedures: (1) estimating the missing values for subsequent determination of the ANOVA table, and (2) using the general regression significance test.

A number of existing research studies are concerned with an incomplete Latin square in which some units are missing or unobserved. Early discussion on estimation of the missing values in a model's dataset, the so-called missing-plot technique, was documented in Allan and Wishart [5] and Yates [6]. In [5], the derivative of the error-sum-of-square function with respect to the missing value was first calculated prior to obtaining the estimated missing value for the RCBD and the Latin square design by solving such a function. On the other hand, Yates [6] initially determined the fitting of a linear model using the least squares method and the missing value was subsequently estimated with the fitting model function, a procedure which is termed Yates's method. Nevertheless, both methods were criticized by later publications, e.g. Montgomery [1] and Little and Rubin [7], reasoning that the missing-value estimation procedure, which leads to the minimum mean square error to approximate the unobserved data, produces the biased mean squares for potential factors when the null hypothesis is actually correct, thereby resulting in many statistically significant factors.

Montgomery [1] seemingly encouraged the use of the general regression significance test (i.e. the exact approach) to solve the missing-value problems due to the estimate of the sum squares of error without bias. The exact approach requires a set of normal equations to determine the fitted values of the linear statistical model. Prior research on the incomplete Latin square by means of the exact approach is available, e.g. Yates [8], in 1936, focused on the Latin square design with either one treatment, one column or one row missing, while Yates and Hale [9], in 1939, increased the number of scenarios in which more than one treatment, column or row was missing. In particular, the work of Sirikasemsuk [10] seems to be the earliest paper to tackle the problems of incomplete Latin square design of any order consisting of only one unobserved data. His paper used the least square normal equations to find the estimated model parameters and to provide the explicit and mathematical formula for its regression sum of squares, thereby simplifying the calculation process. Nonetheless, neither considered the scenario where two values were missing. Readers are advised to refer to the work of Sirikasemsuk [11] for more details.

The aims of this research thus are to analyze the 4 × 4 Latin square designs with two missing observations using the exact approach and to propose the exact formulae for the estimates of the model parameters applicable to all two-missing-value 4 × 4 Latin square design cases without replication. This research partly resembles the works of Sirikasemsuk [10], De Lury [12], and Kramer and Glass [13]. In [12], the author investigated a number of scenarios of missing values for the 4 × 4 Latin square design with several replicates by fitting the model parameters and estimating the missing data, similar to Yates [6]. However, De Lury [12] failed to reflect the exact two missing values. Meanwhile, Kramer and Glass [13] tackled the two-missing-data Latin square design of any order by estimating the missing data using the same procedure as in Allan and Wishart [5] prior to determination of the bias using the two-way classification and re-computation to obtain the correct treatment sum of squares. Specifically, unlike in [12, 13] which solved the missing-value Latin square problems using the missing-plot technique, this current research has addressed the problem through the exact approach in which the fitted values of the full and reduced model parameters were derived and their explicit formulae were readily available.

This current paper has the same main assumptions as the classical experimental design based on the ANOVA (see Montgomery [1]), namely: (1) the error terms (or residuals) are independently, identically, and normally distributed with zero mean and a constant variance, (2) the mathematical model of y_{ijk} is presented in terms of a linear model, and (3) the potential variable and block variables satisfy a fixed effects model. In fact, if the set of data meets the assumptions above, the exact approach can be applied to solve all incomplete-data experimental designs consisting of many missing values in which the degree of freedom for the error term must be enough for the ANOVA.

The organization of the rest of this research is as follows: Sect. 2 introduces the general regression significance test and presents a set of normal equations for the 4 × 4 Latin square experiments in the event of no missing values. Section 3 describes certain patterns of two missing values for the 4 × 4 Latin square design. Moreover, the fitted values formulae for the full linear statistical model of each pattern are developed. In Sect. 4, the fitted values formulae for the reduced linear statistical model with two missing values are developed. An illustrative example is given in Sect. 5, while the concluding remarks are provided in Sect. 6.

Below are the notations used in this research:

i	index of rows (i = 1, 2, 3, or 4)
j	index of treatments (j = 1, 2, 3, or 4; or 'A', 'B', 'C', or 'D')
k	index of columns (k = 1, 2, 3, or 4)
y_{ijk}	the ijk^{th} observation taken under row i, treatment j, and column k
$y_{...}$	the grand total
$y_{..k}$	the total for the columns
$y_{.j.}$	the total for the treatments
$y_{i..}$	the total for the rows
r	index of a certain row in which the first observation is missing
r''	index of a certain row in which the second observation is missing (use r if two missing values occur in the same row)

m	index of a certain treatment (letter) in which the first observation is missing
m''	index of a certain treatment in which the second observation is missing (use m if two missing values are of the same letter)
c	index of a certain column in which the first observation is missing
c''	index of a certain column in which the second observation is missing (use c if two missing values occur in the same column)
μ	the overall mean
ω_i	the i^{th} row effect
τ_j	the j^{th} treatment effect
λ_k	the k^{th} column effect
ε_{ijk}	random error due to other sources of variability
X	the model parameters ($X = \mu, \omega_i, \tau_j$, or λ_k)
$\hat{X}$	the estimated fitted values of X for the full model ($\hat{X} = \hat{\mu}, \hat{\omega}_i, \hat{\tau}_j$ or $\hat{\lambda}_k$)
$\hat{X}'$	the estimated fitted values of X for the reduced model in which the effect of treatment is ignored ($\hat{X}' = \hat{\mu}', \hat{\omega}_i'$, or $\hat{\lambda}_k'$)
df	the degree of freedom
$R(\mu, \omega, \tau, \lambda)$	the regression sum of squares from fitting the full model y_{ijk}
$R(\mu, \omega, \lambda)$	the regression sum of squares from fitting the reduced model y_{ijk}

It should be noted that the r (or r''), m (or m''), and c (or c'') indices are part of the i, j, and k indices, respectively. For instance, when the first value with the letter '*C*' is not observed at row number 1 and column number 3, the following values of $r = 1$, $c = 3$ and $m = 3$ (or $m =$ '*C*') are returned; and when the second value with the same letter '*C*' is not observed at row number 3 and column number 1, $r'' = 3$, $c'' = 1$ and $m = 3$ with the same value of m are subsequently returned.

$\hat{\omega}_r$, $\hat{\omega}_{r''}$, $\hat{\tau}_m$, $\hat{\tau}_{m''}$, $\hat{\lambda}_c$ and $\hat{\lambda}_{c''}$ are also the fitted values estimates of the full model parameters at the missing-value coordinates. Likewise, $\hat{\omega}_r'$, $\hat{\omega}_{r''}'$, $\hat{\lambda}_c'$ and $\hat{\lambda}_{c''}'$ are the estimates of the fitted values of the reduced model parameters at the missing-value coordinates without consideration of the effects of τ_j.

2 The General Regression Significance Test for the 4 × 4 Latin Square Design

For a typical Latin square design, the linear statistical model can be expressed as

$$y_{ijk} = \mu + \omega_i + \tau_j + \lambda_k + \varepsilon_{ijk} \tag{1}$$

In order to analyze the variances in the experiments, the treatment sum of squares and the error sum of squares are derived. To start, the regression sum of squares for the full linear statistical model is calculated using Eq. (2):

$$R(\mu, \omega, \tau, \lambda) = \hat{\mu} y_{...} + \sum_{i=1}^{4} \hat{\omega}_i y_{i..} + \sum_{j=1}^{4} \hat{\tau}_j y_{.j.} + \sum_{k=1}^{4} \hat{\lambda}_k y_{..k} \tag{2}$$

Likewise, the regression sum of squares from the reduced linear statistical model $y_{ijk} = \mu + \omega_i + \lambda_k + \varepsilon_{ijk}$ without consideration of the effects of treatments (τ_j) can be expressed as

$$R(\mu, \omega, \lambda) = \hat{\mu}' y_{...} + \sum_{i=1}^{4} \hat{\omega}_i' y_{i..} + \sum_{k=1}^{4} \hat{\lambda}_k' y_{..k} \tag{3}$$

It is of importance to note that it is not imperative that the estimated fitted values for the full linear statistical model and for the reduced model be identical, particularly in the event of missing values.

The treatment sum of squares and the error sum of squares can be calculated by

$$SS_{treatment} = R(\tau|\mu, \omega, \lambda) = R(\mu, \omega, \tau, \lambda) - R(\mu, \omega, \lambda) \tag{4}$$

with $df = 3$, and

$$SS_E = \sum_{i=1}^{4} \sum_{j=1}^{4} \sum_{k=1}^{4} y_{ijk}^2 - R(\mu, \omega, \tau, \lambda). \tag{5}$$

If every value could be observed, the degree of freedom (*df*) for the error sum of squares would become 6. Nevertheless, with two observations missing, the degree of freedom (*df*) is thus 4.

In the case of the 4 × 4 Latin square design without missing values, a set of normal equations can be written as

$$\mu \ : \ 16\hat{\mu} + 4(\hat{\omega}_1 + \hat{\omega}_2 + \hat{\omega}_3 + \hat{\omega}_4) + 4(\hat{\tau}_1 + \hat{\tau}_2 + \hat{\tau}_3 + \hat{\tau}_4) + 4(\hat{\lambda}_1 + \hat{\lambda}_2 + \hat{\lambda}_3 + \hat{\lambda}_4) = y_{...} \tag{6}$$

$$\omega_i \ : \ 4\hat{\mu} + 4\hat{\omega}_i + (\hat{\tau}_1 + \hat{\tau}_2 + \hat{\tau}_3 + \hat{\tau}_4) + (\hat{\lambda}_1 + \hat{\lambda}_2 + \hat{\lambda}_3 + \hat{\lambda}_4) = y_{i..} \tag{7}$$

$$\tau_j \ : \ 4\hat{\mu} + 4\hat{\tau}_j + (\hat{\omega}_1 + \hat{\omega}_2 + \hat{\omega}_3 + \hat{\omega}_4) + (\hat{\lambda}_1 + \hat{\lambda}_2 + \hat{\lambda}_3 + \hat{\lambda}_4) = y_{.j.} \tag{8}$$

$$\lambda_k \ : \ 4\hat{\mu} + 4\hat{\lambda}_k + (\hat{\tau}_1 + \hat{\tau}_2 + \hat{\tau}_3 + \hat{\tau}_4) + (\hat{\omega}_1 + \hat{\omega}_2 + \hat{\omega}_3 + \hat{\omega}_4) = y_{..k}. \tag{9}$$

However, in the Latin square system, there are following restrictions:

$$\sum_{i=1}^{4} \hat{\omega}_i = 0, \tag{10}$$

$$\sum_{j=1}^{4} \hat{\tau}_j = 0 \tag{11}$$

$$\sum_{k=1}^{4} \hat{\lambda}_k = 0. \tag{12}$$

Thus, Eqs. (6), (7), (8), (9), (10), (11) and (12) can be solved and rewritten as

$$\hat{\mu} = \frac{y_{...}}{16}, \tag{13}$$

$$\hat{\omega}_i = \frac{y_{i..}}{4} - \hat{\mu}, \tag{14}$$

$$\hat{\tau}_j = \frac{y_{.j.}}{4} - \hat{\mu}, \tag{15}$$

$$\hat{\lambda}_k = \frac{y_{..k}}{4} - \hat{\mu}. \tag{16}$$

In the case of two missing observations, the estimated fitted values (i.e. $\hat{\mu}, \hat{\omega}_i, \hat{\tau}_j, \hat{\lambda}_k, \hat{\mu}', \hat{\omega}_i', \hat{\lambda}_k$) of the parameters belonging to the full and reduced linear statistical models are derived from the normal equations whose details regarding the two-missing-value experiments are respectively presented in Sects. 3 and 4. The estimated fitted values are subsequently substituted into Eqs. (2) and (3) to determine the treatment and the error sums of squares based on Eqs. (4) and (5), respectively.

3 The Fitted Parameters for the Full Model with Two Missing Observations

For a Latin square of 4×4 order with two missing observations, four particular patterns are of interest:

Pattern 1: Two missing values; not in the same row, column or treatment
Pattern 2: Two missing values; not in the same row nor column but with the same treatment
Pattern 3: Two missing values in the same row
Pattern 4: Two missing values in the same column

Figure 1 illustrates four sample patterns of the two-missing-value 4×4 Latin square design. Specifically, in Patterns 3 and 4, the two missing treatments must be of

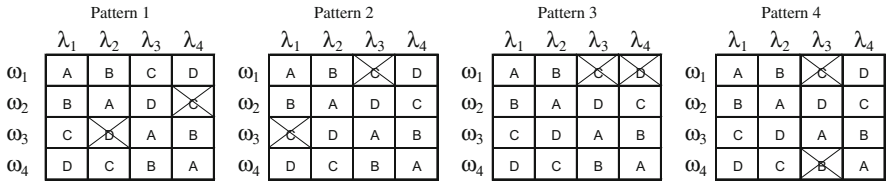


Fig. 1. Four sample patterns of a 4×4 Latin square design with two missing values

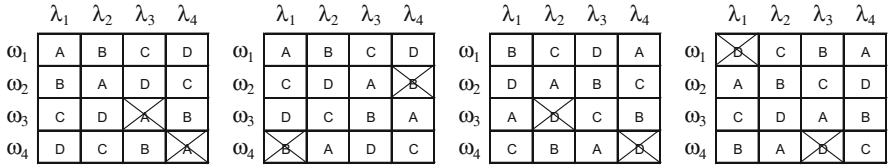


Fig. 2. Other examples of Pattern 2

Table 1. The normal equations for *the full model parameters* in the case of two missing values

Pattern	Normal quation (full model)
Pattern 1	$\mu: \quad 14\hat{\mu} - (\hat{\omega}_r + \hat{\omega}_{r''}) - (\hat{\tau}_m + \hat{\tau}_{m''}) - (\hat{\lambda}_c + \hat{\lambda}_{c''}) = y_{...}$
	$\omega_r: 3\hat{\mu} + 3\hat{\omega}_r - \hat{\tau}_m - \hat{\lambda}_c = y_{r..}$
	$\omega_{r''}: 3\hat{\mu} + 3\hat{\omega}_{r''} - \hat{\tau}_{m''} - \hat{\lambda}_{c''} = y_{r''..}$
	$\tau_m: 3\hat{\mu} - \hat{\omega}_r + 3\hat{\tau}_m - \hat{\lambda}_c = y_{.m.}$
	$\tau_{m''}: 3\hat{\mu} - \hat{\omega}_{r''} + 3\hat{\tau}_{m''} - \hat{\lambda}_{c''} = y_{.m''.}$
	$\lambda_c: 3\hat{\mu} - \hat{\omega}_r - \hat{\tau}_m + 3\hat{\lambda}_c = y_{..c}$
	$\lambda_{c''}: 3\hat{\mu} - \hat{\omega}_{r''} - \hat{\tau}_{m''} + 3\hat{\lambda}_{c''} = y_{..c''}$
Pattern 2	$\mu: \ 14\hat{\mu} - (\hat{\omega}_r + \hat{\omega}_{r''}) - 2\hat{\tau}_m - (\hat{\lambda}_c + \hat{\lambda}_{c''}) = y_{...}$
	$\omega_r: 3\hat{\mu} + 3\hat{\omega}_r - \hat{\tau}_m - \hat{\lambda}_c = y_{r..}$
	$\omega_{r''}: 3\hat{\mu} + 3\hat{\omega}_{r''} - \hat{\tau}_m - \hat{\lambda}_{c''} = y_{r''..}$
	$\tau_m: 2\hat{\mu} - (\hat{\omega}_r + \hat{\omega}_{r''}) + 2\hat{\tau}_m - (\hat{\lambda}_c + \hat{\lambda}_{c''}) = y_{.m.}$
	$\lambda_c: 3\hat{\mu} - \hat{\omega}_r - \hat{\tau}_m + 3\hat{\lambda}_c = y_{..c}$
	$\lambda_{c''}: 3\hat{\mu} - \hat{\omega}_{r''} - \hat{\tau}_m + 3\hat{\lambda}_{c''} = y_{..c''}$
Pattern 3	$\mu: \ 14\hat{\mu} - 2\hat{\omega}_r - (\hat{\tau}_m + \hat{\tau}_{m''}) - (\hat{\lambda}_c + \hat{\lambda}_{c''}) = y_{...}$
	$\omega_r: 2\hat{\mu} + 2\hat{\omega}_r - (\hat{\tau}_m + \hat{\tau}_{m''}) - (\hat{\lambda}_c + \hat{\lambda}_{c''}) = y_{r..}$
	$\tau_m: 3\hat{\mu} - \hat{\omega}_r + 3\hat{\tau}_m - \hat{\lambda}_c = y_{.m.}$
	$\tau_{m''}: 3\hat{\mu} - \hat{\omega}_r + 3\hat{\tau}_{m''} - \hat{\lambda}_{c''} = y_{.m''.}$
	$\lambda_c: 3\hat{\mu} - \hat{\omega}_r - \hat{\tau}_m + 3\hat{\lambda}_c = y_{..c}$
	$\lambda_{c''}: 3\hat{\mu} - \hat{\omega}_r - \hat{\tau}_{m''} + 3\hat{\lambda}_{c''} = y_{..c''}$
Pattern 4	$\mu: \ 14\hat{\mu} - (\hat{\omega}_r + \hat{\omega}_{r''}) - (\hat{\tau}_m + \hat{\tau}_{m''}) - 2\hat{\lambda}_c = y_{...}$
	$\omega_r: 3\hat{\mu} + 3\hat{\omega}_r - \hat{\tau}_m - \hat{\lambda}_c = y_{r..}$
	$\omega_{r''}: 3\hat{\mu} + 3\hat{\omega}_{r''} - \hat{\tau}_{m''} - \hat{\lambda}_c = y_{r''..}$
	$\tau_m: 3\hat{\mu} - \hat{\omega}_r + 3\hat{\tau}_m - \hat{\lambda}_c = y_{.m.}$
	$\tau_{m''}: 3\hat{\mu} - \hat{\omega}_{r''} + 3\hat{\tau}_{m''} - \hat{\lambda}_c = y_{.m''.}$
	$\lambda_c: 2\hat{\mu} - (\hat{\omega}_r + \hat{\omega}_{r''}) - (\hat{\tau}_m + \hat{\tau}_{m''}) + 2\hat{\lambda}_c = y_{..c}$

different values due to the Latin square design characteristic. In fact, there are several possibilities to configure the values/treatments in the four patterns. For instance, the depiction of Pattern 2 in Fig. 1 is one possible configuration among a myriad of

arrangements, as illustrated in Fig. 2. Thus, the proposed formulae are applicable to a two-missing-value 4 × 4 Latin square design of any configuration.

Table 1 presents the normal equations for the model parameters, i.e. ω_r, $\omega_{r''}$, τ_m, $\tau_{m''}$, λ_c, $\lambda_{c''}$, pertaining to the four patterns in Fig. 1. Since Eqs. (10), (11) and (12) are true, they are thus utilized in the two-missing-value Latin square analysis. In addition, each set of the normal equations in Table 1 can be solved for the estimates of the fitted values of the full model parameters, as presented in Tables 2 and 3.

Table 2. The estimated overall means for *the full model* for the two-missing-value problem

Pattern	The estimated overall mean (full model)
Pattern 1	$\hat{\mu} = (y_{...} + y_{r..} + y_{r''..} + y_{.m.} + y_{.m''.} + y_{..c} + y_{..c''})/32$
Pattern 2	$\hat{\mu} = (y_{r..} + y_{r''..} + 2y_{.m.} + y_{..c} + y_{..c''})/16$
Pattern 3	$\hat{\mu} = (2y_{r..} + y_{.m.} + y_{.m''.} + y_{..c} + y_{..c''})/16$
Pattern 4	$\hat{\mu} = (y_{r..} + y_{r''..} + y_{.m.} + y_{.m''.} + 2y_{..c})/16$

Table 3. The fitted values for *the full model* in the case of two missing values

The estimate of fitted values (full model)
Pattern 1
$\omega_r : \hat{\omega}_r = -3\hat{\mu} + (2y_{r..} + y_{.m.} + y_{..c})/4 = (13y_{r..} - 3y_{r''..} + 5y_{.m.} - 3y_{.m''.} + 5y_{..c} - 3y_{..c''} - 3y_{...})/32$
$\omega_{r''} : \hat{\omega}_{r''} = -3\hat{\mu} + (2y_{r''..} + y_{.m''.} + y_{..c''})/4 = (-3y_{r..} + 13y_{r''..} - 3y_{.m.} + 5y_{.m''.} - 3y_{..c} + 5y_{..c''} - 3y_{...})/32$
$\tau_m : \tau_m = -3\hat{\mu} + (y_{r..} + 2y_{.m.} + y_{..c})/4 = (5y_{r..} - 3y_{r''..} + 13y_{.m.} - 3y_{.m''.} + 5y_{..c} - 3y_{..c''} - 3y_{...})/32$
$\tau_{m''} : \tau_{m''} = -3\hat{\mu} + (y_{r''..} + 2y_{.m''.} + y_{..c''})/4 = (-3y_{r..} + 5y_{r''..} - 3y_{.m.} + 13y_{.m''.} - 3y_{..c} + 5y_{..c''} - 3y_{...})/32$
$\lambda_c : \hat{\lambda}_c = -3\hat{\mu} + (y_{r..} + y_{.m.} + 2y_{..c})/4 = (5y_{r..} - 3y_{r''..} + 5y_{.m.} - 3y_{.m''.} + 13y_{..c} - 3y_{..c''} - 3y_{...})/32$
$\lambda_{c''} : \hat{\lambda}_{c''} = -3\hat{\mu} + (y_{r''..} + y_{.m''.} + 2y_{..c''})/4 = (-3y_{r..} + 5y_{r''..} - 3y_{.m.} + 5y_{.m''.} - 3y_{..c} + 13y_{..c''} - 3y_{...})/32$
Pattern 2
$\omega_r : \hat{\omega}_r = (3y_{r..} + y_{.m.} + y_{..c} - y_{...})/8$
$\omega_{r''} : \hat{\omega}_{r''} = (3y_{r''..} + y_{.m.} + y_{..c''} - y_{...})/8$
$\tau_m : \hat{\tau}_m = 3\hat{\mu} - [(y_{...} - y_{.m.})/4] = [3(y_{r..} + y_{r''..} + y_{..c} + y_{..c''})/16] + (5y_{.m.}/8) - (y_{...}/4)$
$\lambda_c : \hat{\lambda}_c = (y_{r..} + y_{.m.} + 3y_{..c} - y_{...})/8$
$\lambda_{c''} : \hat{\lambda}_{c''} = (y_{r''..} + y_{.m.} + 3y_{..c''} - y_{...})/8$
Pattern 3
$\omega_r : \hat{\omega}_r = 3\hat{\mu} - [(y_{...} - y_{r..})/4] = [3(y_{.m.} + y_{.m''.} + y_{..c} + y_{..c''})/16] + (5y_{r..}/8) - (y_{...}/4)$
$\tau_m : \hat{\tau}_m = (y_{r..} + 3y_{.m.} + y_{..c} - y_{...})/8$
$\tau_{m''} : \hat{\tau}_{m''} = (y_{r..} + 3y_{.m''.} + y_{..c''} - y_{...})/8$
$\lambda_c : \hat{\lambda}_c = (y_{r..} + y_{.m.} + 3y_{..c} - y_{...})/8$
$\lambda_{c''} : \hat{\lambda}_{c''} = (y_{r..} + y_{.m''.} + 3y_{..c''} - y_{...})/8$
Pattern 4
$\omega_r : \hat{\omega}_r = (3y_{r..} + y_{.m.} + y_{..c} - y_{...})/8$
$\omega_{r''} : \hat{\omega}_{r''} = (3y_{r''..} + y_{.m''.} + y_{..c} - y_{...})/8$
$\tau_m : \hat{\tau}_m = (y_{r..} + 3y_{.m.} + y_{..c} - y_{...})/8$
$\tau_{m''} : \hat{\tau}_{m''} = (y_{r''..} + 3y_{.m''.} + y_{..c} - y_{...})/8$
$\lambda_c : \hat{\lambda}_c = 3\hat{\mu} - [(y_{...} - y_{..c})/4] = [3(y_{r..} + y_{r''..} + y_{.m.} + y_{.m''.})/16] + (5y_{..c}/8) - (y_{...}/4)$

In the incomplete Latin square design for the estimates of the full model parameters whose positions are not concerned with the missing values, the normal equations for the effects of rows, treatments, and columns have still satisfied Eqs. (7), (8), and (9), respectively. In addition, the fitted values of the parameters can be derived by Eqs. (14), (15) and (16). Nevertheless, the fitted overall means ($\hat{\mu}$) of the four patterns of the two-missing-value 4 × 4 Latin square design would not be equal to Eq. (13) but identical to the estimates in Table 2. Thus, the estimates of the fitted values of the parameters in the case of no missing value could be computed by substituting the fitted overall means in Table 2 into Eqs. (14), (15) and (16).

4 The Fitted Parameters for the Reduced Model with Two Missing Observations

In this research, the reduced model is the linear statistical model that ignores the effect of the main or potential factor consisting of several treatments; but that takes into consideration the effects of the row and column. The reduced model is utilized to calculate the remaining fitted values, i.e. $\hat{\mu}', \hat{\omega}_i', \hat{\lambda}_k'$. The regression sum of squares and the treatment sum of squares from the reduced model are subsequently determined using Eqs. (3) and (4), respectively.

By following the same procedure as in Sect. 3, the fitted values are determined and tabulated in Tables 4 and 5. The normal equations for the reduced model under the two-missing-value condition are similar to those for the full model (Table 1) except for the effect of treatments (τ_j) which is ignored in the reduced model. The estimated overall means and the fitted values for the reduced model nevertheless can be determined and are respectively presented in Tables 4 and 5, unlike Tables 2 and 3, respectively.

Tables 4 and 5 show the fitted values of the parameters concerned with the positions of the missing values. Based on Eqs. (14) and (16), the estimates of the remaining parameters can be derived by

$$\hat{\omega}_i' = \frac{y_{i\cdot\cdot}}{4} - \hat{\mu}', \tag{17}$$

$$\hat{\lambda}_k' = \frac{y_{\cdot\cdot k}}{4} - \hat{\mu}'. \tag{18}$$

Table 4. The estimated overall means for *the reduced model* for the two-missing-value problem

Pattern	The estimated overall mean (reduced model)
Patterns 1 and 2	$\hat{\mu}' = (2y_{\cdots} + y_{r\cdot\cdot} + y_{r''\cdot\cdot} + y_{\cdot\cdot c} + y_{\cdot\cdot c''})/40$
Pattern 3	$\hat{\mu}' = (y_{\cdots} + 2y_{r\cdot\cdot} + y_{\cdot\cdot c} + y_{\cdot\cdot c''})/24$
Pattern 4	$\hat{\mu}' = (y_{\cdots} + y_{r\cdot\cdot} + y_{r''\cdot\cdot} + 2y_{\cdot\cdot c})/24$

Interestingly, the estimated overall means ($\hat{\mu}'$) are the values from Table 4.

The estimates of the fitted parameters are provided in Tables 2, 3, 4 and 5, which would contribute to the shorter calculation time and less complex procedure.

Table 5. The fitted values for *the reduced model* in the case of two missing values

Pattern	The estimate of fitted values (reduced model)
Patterns 1 and 2	ω_r : $\hat{\omega}'_r = (3y_{r..} + y_{..c} - 12\hat{\mu}')/8 = (27y_{r..} - 3y_{r''..} + 7y_{..c} - 3y_{..c''} - 6y_{...})/80$
	$\omega_{r''}$: $\hat{\omega}'_{r''} = (3y_{r''..} + y_{..c''} - 12\hat{\mu}')/8 = (-3y_{r..} + 27y_{r''..} - 3y_{..c} + 7y_{..c''} - 6y_{...})/80$
	$\lambda_c : \hat{\lambda}'_c = (y_{r..} + 3y_{..c} - 12\hat{\mu}')/8 = (7y_{r..} - 3y_{r''..} + 27y_{..c} - 3y_{..c''} - 6y_{...})/80$
	$\lambda_{c''}$: $\hat{\lambda}'_{c''} = (y_{r''..} + 3y_{..c''} - 12\hat{\mu}')/8 = (-3y_{r..} + 7y_{r''..} - 3y_{..c} + 27y_{..c''} - 6y_{...})/80$
Pattern 3	$\omega_r : \hat{\omega}'_r = (4y_{r..} + y_{..c} + y_{..c''} - y_{...})/8$
	$\lambda_c : \hat{\lambda}'_c = (y_{r..} + 4y_{..c} - y_{...})/12$
	$\lambda_{c''} : \hat{\lambda}'_{c''} = (y_{r..} + 4y_{..c''} - y_{...})/12$
Pattern 4	$\omega_r : \hat{\omega}'_r = (4y_{r..} + y_{..c} - y_{...})/12$
	$\omega_{r''} : \hat{\omega}'_{r''} = (4y_{r''..} + y_{..c} - y_{...})/12$
	$\lambda_c : \hat{\lambda}'_c = (y_{r..} + y_{r''..} + 4y_{c..} - y_{...})/8$

5 Example and Comparison

This section provides an example to illustrate the approach previously described. As illustrated in Table 6, which satisfies Pattern 1, two observations of a dataset are missing. The formulae for the fitted model parameters of Pattern 1 as presented in Table 7 are thus employed, and the treatment and error sums of squares are determined.

Table 6. Data collected during the experiments

Row blocking variable	Column blocking variable				
	1	2	3	4	
1	C = 12	D = 15	A = 7	B = 8	$y_{1..} = 42$
2	B = 8	C = 19	D = —	A = 11	$y_{2..} = 38$
3	A = 6	B = —	C = 12	D = 10	$y_{3..} = 28$
4	D = 11	A = 11	B = 13	C = 15	$y_{4..} = 50$
	$y_{..1} = 37$	$y_{..2} = 45$	$y_{..3} = 32$	$y_{..4} = 44$	$y_{...} = 158$
	$y_{.1.} = 35$	$y_{.2.} = 29$	$y_{.3.} = 58$	$y_{.4.} = 36$	

For the full model, $\hat{\mu}$, $\hat{\omega}_2$, $\hat{\omega}_3$, $\hat{\tau}_2$, $\hat{\tau}_4$, $\hat{\lambda}_2$, and $\hat{\lambda}_3$ can be promptly calculated from Tables 2 and 3; while $\hat{\omega}_1$, $\hat{\omega}_4$, $\hat{\tau}_1$, $\hat{\tau}_3$, $\hat{\lambda}_1$, and $\hat{\lambda}_4$ are computed by Eqs. (14), (15), and (16). For the reduced model, $\hat{\mu}'$, $\hat{\omega}'_2$, $\hat{\omega}'_3$, $\hat{\lambda}'_2$, and $\hat{\lambda}'_3$ can be promptly calculated from

Table 7. The fitted model parameters

Model	Estimated fitted parameters
Full model	$\hat{\mu} = 11.4375$
	$\hat{\omega}_1 = -0.9375$, $\hat{\omega}_2 = 1.6875$, $\hat{\omega}_3 = -1.8125$, $\hat{\omega}_4 = 1.0625$
	$\hat{\tau}_1 = -2.6875$, $\hat{\tau}_2 = -1.5625$, $\hat{\tau}_3 = 3.0625$, $\hat{\tau}_4 = 1.1875$
	$\hat{\lambda}_1 = -2.1875$, $\hat{\lambda}_2 = 2.4375$, $\hat{\lambda}_3 = 0.1875$, $\hat{\lambda}_4 = -0.4375$
Reduced model	$\hat{\mu} = 11.4750$
	$\hat{\omega}'_1 = -0.9750$, $\hat{\omega}'_2 = 1.0375$, $\hat{\omega}'_3 = -1.0875$, $\hat{\omega}'_4 = 1.0250$
	$\hat{\lambda}'_1 = -2.2250$, $\hat{\lambda}'_2 = 3.1625$, $\hat{\lambda}'_3 = -0.4625$, $\hat{\lambda}'_4 = -0.4750$

Table 8. The analysis of the variance

$R(\mu, \omega, \tau, \lambda) = 1930.7500$ $R(\mu, \omega, \lambda) = 1856.6125$
$SS_{treatment} = 74.1375$ (with $df = 3$) $SS_E = 13.2500$ (with $df = 4$)
Mean Square of Treatment = 24.7125 *Mean Square of Error* = 3.3125
F test = 7.46037736

Tables 4 and 5; while $\hat{\omega}'_1$, $\hat{\omega}'_4$, $\hat{\lambda}'_1$, and $\hat{\lambda}'_4$ are computed by Eqs. (17) and (18). Finally, the important results can be easily calculated as presented in Table 8.

In this paper, comparisons of the errors in terms of the mean absolute deviation (MAD) and the mean square of treatment by means of the exact approach and the missing-plot technique are made. The errors and MAD can be expressed as

$$e_{ijk} = y_{ijk} - \hat{y}_{ijk}, \tag{19}$$

$$MAD = \frac{\sum_{All\ i} \sum_{All\ j} \sum_{All\ k} \left| e_{ijk} \right|}{n} \tag{20}$$

where $\hat{y}_{ijk} = \hat{\mu} + \hat{\omega}_i + \hat{\tau}_j + \hat{\lambda}_k$ and $n = 14$ in the case when Table 6 is considered.

For comparison purposes, the missing-plot technique based on the works of Allan and Wishart [5] and Yates [6] is considered here. Hence, the missing values in Table 6, i.e., y_{322} and y_{243}, are approximated in order to bring about the complete information design. By following the same procedures as in the work of Yates [6], y_{322} and y_{243} are 10.5 and 14.5, respectively. After using the missing-plot technique, the fitted parameters can be determined by Eqs. (13), (14), (15), and (16); while the mean square of treatment can be determined based on classical experimental design (see Montgomery [1]). The detailed comparison results are presented in Table 9 below. From Table 9, it can be

Table 9. Comparisons of the mean absolute deviation (MAD) and the mean square of treatment by means of the exact approach and the missing-plot technique

	Exact approach	Missing-plot technique
Mean absolute deviation	75.50	75.50
$SS_{treatment}$	74.1375	81.8130
Mean square of treatment	**24.7125**	**27.2710**

concluded the mean absolute deviations of both two methods are equal. However, the mean square of treatment using the exact approach is better than that using the missing-plot technique, – that is, the mean square of treatment using the exact approach is unbiased.

6 Conclusion

Meticulous investigation of the related literature revealed that there was no ready-made formula to analyze the Latin square design with the two missing values through the exact approach in which no estimate of the missing values was made. This research is concerned with the two-missing-value 4×4 Latin square designs without replication. In the data analysis, the general regression significance test (i.e. the exact approach) was utilized to realize more reasonable treatment sum of squares relative to Allan and Wishart [5] and Yates [6], because the calculation of the mean square of treatment using the exact approach was more accurate than that using the missing-plot technique. In addition, the extent of the mean absolute deviation of residuals under the two approaches was exactly the same. With the missing-plot technique, the estimated values were calculated based on the minimize error sum of squares in which the errors were defined as the all differences between the actual and predicting values from fitting the linear model's parameters. Hence, both two techniques similarly tried to solve the incomplete Latin square design to affect the analysis of variance at least. In this paper, every possible scenario associated with two missing values was taken into account. Also, this research helps to enrich a clear understanding of education in the engineering statistics. Furthermore, the observations showed that Patterns 2, 3 and 4 were structurally identical. Moreover, future research should attempt to apply the exact approach to solving the Latin square designs of higher order and with more missing values.

References

1. Montgomery, D.C.: Design and Analysis of Experiments. Wiley, New York (2008)
2. Mead, R., Gilmour, S.G., Mead, A.: Statistical Principles for the Design of Experiments: Applications to Real Experiments, vol. 36. Cambridge University Press, Cambridge (2012)
3. Stones, D.S.: The many formulae for the number of Latin rectangles. Electron. J. Comb. **17**(1), 46 (2010)

4. Youden, W.J.: Use of incomplete block replications in estimating tobacco-mosaic virus. Contrib. Boyce Thompson Inst. **9**(1), 41–48 (1937)
5. Allan, F.E., Wishart J.: A method of estimating the yield of a missing plot in field - experimental work. J. Agric. Sci. **20**(3), 399–406 (1930)
6. Yates, F.: The analysis of replicated experiments when the field results are incomplete. Emp. J. Exp. Agric. **1**(2), 129–142 (1933)
7. Little, R.J.A., Rubin D.B.: Statistical Analysis with Missing Data, 2nd edn. Wiley, New York (2002)
8. Yates, F.: Incomplete Latin squares. J. Agric. Sci. **26**(2), 301–315 (1936)
9. Yates, F., Hale, R.W.: The analysis of Latin squares when two or more rows, columns, or treatments are missing. J. Roy. Stat. Soc. **6**(1), 67–79 (1939)
10. Sirikasemsuk, K.: One missing value problem in Latin square design of any order: regression sum of squares. In: 2016 Joint 8th International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems, pp. 142–147. IEEE Press, Japan (2016)
11. Sirikasemsuk, K.: A Review on incomplete Latin square design of any order. In: Rusli, N., Zaimi, W.M., Khazali, K.A., Masnan, M.J., Daud, W.S., Abdullah, N., Yahya, Z., Amin, N.A., Aziz, N.H., Yusuf, Y.N. (eds.) AIP Conference Proceedings 2016, vol. 1775, p. 030022. AIP Publishing (2016)
12. De Lury, D.B.: The analysis of Latin squares when some observations are missing. J. Am. Stat. Assoc. **41**(235), 370–389 (1946)
13. Kramer, C.Y., Glass, S.: Analysis of variance of a Latin square design with missing observations. Appl. Stat. **9**(1), 43–50 (1960)

Software Size Estimation in Design Phase Based on MLP Neural Network

Benjamas Panyangam(✉) and Matinee Kiewkanya

Faculty of Science, Department of Computer Science,
Chiang Mai University, Chiang Mai, Thailand
{benjamas.p,matinee.k}@cmu.ac.th

Abstract. Size estimation is one of important processes related to success of software project management. This paper presents novel software size estimation model by using Multilayer Perceptron approach. Software size in terms of Lines of code is used as criterion variable. Structural complexity metrics are used as predictors. The metrics can be captured from a software design model named UML Class diagram. A high predictive ability of the model is shown with correlation coefficient measure. Moreover, four training algorithms; Levenberg-Marquardt, Scaled Conjugate Gradient, Broyden-Fletcher-Golfarb-Shanno and Bayesian Regularization, have been applied on the network for better estimation. The obtained results indicate the highest accuracy on the model with Bayesian Regularization algorithm.

Keywords: MLP neural network · Software size estimation · Software metrics · Training algorithm

1 Introduction

Software size estimation is an activity in software engineering. Software size can be measured in various aspects such as length (physical size), effort, functionality and complexity. Software size measurements reflect efforts, costs and productivities for software development. Estimating software size in early phases can help the project manager for planning about human resource, cost, schedule and activities in later phases. Lines of code (LOC) is one of well-known and popular size measurements because it is easy to understand and collect. It also provides the ability for automated estimation and the ease of historical data usage because the data related to estimation in the past are collected by LOC approach. The relationships between LOC and other software aspects have been explored in many research works. Bhatia and Malhotra [1] surveyed impact of LOC on software complexity from various research works. They concluded that LOC had strong relationship with software complexity and bugs. Tashtoush and his colleagues [2] investigated the correlation between complexity measurements including LOC, Cyclomatic Complexity and Hallstead Complexity form the data sets provided by Metrics Data Program repository and NASA IV&V Facility. The result showed the evidence of high correlation between Cyclometric Complexity and LOC. Zhang [3] explored the relationship between LOC and software defects by using case study of NASA and Eclipse datasets. The result showed that LOC could be a

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_8

good indicator for predicting the number of defects. The constructed prediction model by using 5 techniques consisting Multilayer Perceptron (MLP), K-Star, Logistic Regression, Decision Tree and Naïve Bayes was also compared and discussed in the paper. Research topics related to software size estimation are still attractive for researchers. Bhati and Pooja [4] summarized guidelines for defining and counting the popular software size metrics including LOC, Function Points and Token Points. The interesting conclusion is that among many estimation techniques, no technique provides the best accuracy. Thus, organizations should apply more than one technique to get high estimation accuracy. Jain et al. [5] introduced a methodology for estimating object-oriented software size by using Predictive Object Point (POP) metrics. The software size in terms of Kilo Lines of Code (KLOC) was predicted from POP by using their proposed linear regression model. Harizi discussed about the role and potency of UML Class diagram, a well-known software design model, in software size estimation [6]. He presented the parameter list which can be captured from Class diagram and also provided the weight values for each parameter.

2 Research Objectives

As already mentioned in the previous section, software size estimation should be done in early phases in order to gain the benefit for project management planning. This work aims to construct software size estimation model which can be utilized in the software design phase. There are four objectives for this research.

- To explore the software metrics which can be captured in design phase and utilized as predictors for software size in terms of LOC.
- To construct software size estimation model by applying technique of MLP neural network.
- To gain the best fit MLP neural network model for software size estimation by comparing the performance accuracy of estimation model obtained from different neuron numbers.
- To evaluate performance of various training algorithms by comparing the model accuracy of the estimation models obtained from 4 types of training algorithms: Levenberg-Marquardt, Scaled Conjugate Gradient, Broyden-Fletcher-Golfarb-Shanno and Bayesian Regularization.

3 Multilayer Percepton

Artificial Neural Network (ANN) is a data modeling technique inspired by human brain process. ANN consists of a large number of interconnected processing elements as neurons in brain. There are many ANN types introduced to model for different properties of systems. Multilayer Perceptron with feed-forward network topology is a popular and successful ANN architecture. Typically, MLP network structure consists of several layers of neuron nodes, i.e. one input layer, one or more hidden layers and one output layer. Each layer is built with a set of neurons and each neuron is fully

connected to the neurons in the next layer. A simple structure of MLP neural network having three layers with only single output neuron is presented in Fig. 1(a). The first layer is called input layer. It consists of a set of neuron nodes corresponding to input parameters of the considered problem. The second layer is called hidden layer and used to capture the relationship among the parameters. The last layer is output layer composing of neurons corresponding to a predicted result.

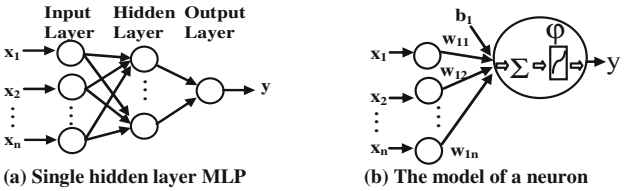


Fig. 1. A simple structure of MLP neural network

Mathematically, the input signals are multiplied by connection weights and summed before a transfer function gains the output of each neuron as shown in Fig. 1(b). Thus, the relationship between the output and a set of input variables in MLP with single hidden layer can be expressed by Eq. (1).

$$y = \varphi_2\left(\sum_{j=1}^{m} w_{2,j}\left(\varphi_1\left(\sum_{i=1}^{n} w_{1,i}x_i + b_1\right)\right) + b_2\right) \tag{1}$$

where

n is the number of input parameters of the model, i.e. number of input neurons.
M is the number of hidden neurons.
Y is the output of the model.
$\mathbf{x_i}$ is the input variable of the model (i = 1…n).
$w_{1,i}$ are connected weights between input and hidden neurons.
$w_{2,j}$ are connected weights between hidden and output neurons(j = 1…m).
φ_1 and b_1 are activation function and bias term used in hidden layer.
φ_2 and b_2 are activation function and bias term used in output layer.

In this study, two commonly used activation functions are applied to the MLP model as displayed in Eq. (2). φ_1 is hyperbolic tangent function utilized to make the outputs for neurons in hidden layer. It is a symmetric shaped function and its output lies in the range (−1, 1). φ_2 is linear function applied to neurons in output layer to generate the outputs. The function produces the same outputs as its input.

$$y = \varphi_2\left(\sum_{j=1}^{m} w_{2,j}\left(\varphi_1\left(\sum_{i=1}^{n} w_{1,i}x_i + b_1\right)\right) + b_2\right) \tag{2}$$

In MLP model, the back propagation learning method based on supervised learning is utilized to train the network. The errors at the output layer are propagated backward to input layer through hidden layer in the network to obtain the final desired outputs. The training process will use this error information to adjust network parameters in order to reduce the error and obtain close values to targets as outputs.

MLP network with back propagation learning has been investigated to construct the estimation models in many researches such as forecasting weather [7], and stock movements [8]. The results showed high performance of MLP models for capability of yielding good results. In [8–10], the ANN models were suggested for predicting data and their performance of the models with NN method showed more significant than linear regression method. Moreover, Aggarwal et al. [11] applied ANN approach for software quality prediction.ANN model with Bayesian Regularization training algorithm provided a sufficient model for predicting maintenance effort from Object-Oriented (OO) metrics. The different methods such as MLP, Radial Basis Function Neural Network and Support Vector Machines had been also studied for software effort estimation [12].

4 Methodology

The main objective of this research is to study on application of MLP neuron network for constructing software size estimation model. There are 5 methodology steps as follows:

(1) Data selection by choosing the suitable data variables for constructing the software size estimation model.
(2) Data preparation by normalizing data into suitable form before feeding into the network.
(3) MLP neuron network model setup for software size estimation.
(4) Network learning process with training algorithm to get the minimum error between actual and predicted outputs.
(5) Performance evaluation for the accuracy of the obtained model.

4.1 Data Selection

UML Class diagram is an important design model for object-oriented software. It represents software structure including classes, attributes, methods and relationships among classes. Software design can defect the quality and the quantity of source code implemented in later phase. Since the strong correlation between software size and software complexity measures has been reported in [13], this work selects 8 structural complexity metrics as predictors for software size. They are shown in Table 1. All of them can be measured from UML Class diagram. They were proposed in the previous work of our research team [14]. This work considers software size in terms of LOC which is defined as the definition of Source lines of code by counting every lines of code excepting blank lines and comment lines.

Table 1. Structural complexity metrics.

Metric	Description
NC	The total number of classes
ANA	The average number of attributes
ANM	The average number of methods (excluding constructor methods, get methods and set methods)
ANSM	The average number of set methods
ANGM	The average number of get methods
ANCM	The average number of constructor methods
NGen	The total number of generalization relationships
NAssoc	The total number of association relationships (including composition and aggregation relationships)

Therefore, the input is a set of structural complexity metrics and the output is LOC. The data used in this work are collected from 60 C++ software with various sizes and domains. Each sample has a number of lines of code between 1,737–465,458 lines and contain between 10–469 classes.

4.2 Data Preparation

In this study, for each experiment as described in the next section, data set of 60 C++ software sample is randomly separated into 40 training data points and 20 test data points. Before ANN training, the data are preprocessed by transforming the input data into the suitable form for training the network. The goal of this data preprocessing is to improve network performance [15, 16]. In this study, because of using tan-sigmoid activation function for training the network, the input data are normalized in the range of −1 and 1 by using the Min-Max normalization method [16, 17] as expressed in Eq. (3), where $\boldsymbol{x_i}$ is the original data, $\boldsymbol{x_{inorm}}$ is the normalized data and $\boldsymbol{x_{max}}$ and $\boldsymbol{x_{min}}$ are the maximum and minimum values.

$$x_{inorm} = \frac{2 * x_i - x_{max} - x_{min}}{x_{max} - x_{min}} \tag{3}$$

4.3 MLP Network Model Setup

This research aims to develop neural network model with 8 selected variables of structural complexity metrics as inputs and one software size metric, LOC, as the output. Generally, the accuracy of the network model depends on several important parameters such as number of hidden layers, number of hidden neurons in hidden layer, activation function, and training algorithm used to adjust the weights in the network. They are determined using trial and error methods. However, many previous researches have revealed that neural networks with one hidden layer are suitable for the most applications. To accomplish the objective of this work, the different network structures

with varying hidden neurons are constructed to select the best model. All the designed networks contain 8 input units and 1 output neuron in output layer. The sigmoid function and linear transfer function are performed as activation functions in hidden layer and output layer, respectively.

4.4 Neural Network Training

In this research, 4 different types of algorithms are utilized to minimize the output error during training process. They are described as follows:

- Levenberg-Marquardt (LM) algorithm uses the second order derivatives of the mean squared error between predicted outputs and actual outputs. The techniques based on Gauss Newton and Gradient Descent methods are combined to minimize the error function. It provides the fast convergence to accomplish the learning of the network.
- Scaled Conjugate Gradient (SCG) algorithm has been introduced by Moller to reduce time-consuming line search.
- Broyden-Fletcher-Golfarb-Shanno (BFGS) algorithm is one of the most popular of Quasi-Newton algorithms. The method often converges faster than other conjugate gradient methods.
- Bayesian Regularization (BR) algorithm is considered as one of the best methods to prevent over fitting tendency and improve the prediction accuracy.

4.5 Performance Evaluation

The performance measures based on statistical metrics, namely, Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and coefficient of determination (R^2) are used for evaluating the prediction accuracy. These error measures are calculated by Eqs. (4)–(6) defined as follows:

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(y_k - t_k)^2} \tag{4}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{k=1}^{n}|y_k - t_k| \tag{5}$$

$$R^2 = \frac{\sum_{k=1}^{n}(y_k - \bar{y})(t_k - \bar{t})}{\sqrt{\sum_{k=1}^{n}(y_k - \bar{y})^2 \sum_{k=1}^{n}(t_k - \bar{t})^2}} \tag{6}$$

where

$\boldsymbol{n}$ is the number of inputs feed into the model.
$\boldsymbol{y_k}$ is predicted value and $\bar{y}$ is average of all predicted values.
$\boldsymbol{t_k}$ is actual value and $\bar{t}$ is average of all actual values.

RMSE and MAE indicate average magnitude of the errors. Except, MAE function is computed without considering their signs. Both measures show the errors in the same unit and scale of input parameters. Their values vary between 0 and 1. Lower values are better and zero means no error. The coefficient of the determination (R^2) indicates the degree of the correlation between actual value and predicted value. The higher R^2 value (maximum is 1) means the better performance.

5 Experimental Results

In order to construct software size estimation model based on MLP neural network approach, two experiments are setup as follows:

5.1 Exploring MLP Model Performance with Different Neuron Numbers

First experiment setup aims to assess the MLP model performance for software size estimation. A MLP model is constructed for predicting LOC using 8 selected software metrics. The MLP network with single hidden layer is tested with LM algorithm. The experiment is tested on the network with different number of neurons (nH) between 4 and 16, i.e. 4, 6, 7, 8, 9, 10, 12, 14 and 16 in the hidden layer. Ten-cross validation is used for each training process to predict LOC. Then, the performance measures, R^2 and RMSE, are averaged to evaluate difference between predicted value and actual value of LOC as shown in Table 2. According to our experimental results, 8-8-1 MLP model is selected as it achieves very satisfying statistical results. The model gives the smallest RMSE of 0.2448 among all tests. This neuron network presents the highest R^2 with 0.8314 on the training set and 0.8631 on the test set. However, all training gives R^2 higher than 0.8 on the test data set. Thus, the introduced MLP model shows the ability for software size estimation.

Table 2. Performance evaluation of training with different neuron numbers.

MLP model	Performance measure				
	R^2		RMSE		
	Train	Test	Train	Test	All
8-4-1	0.7952	0.8313	0.2706	0.2241	0.2559
8-6-1	0.7938	0.8426	0.2961	0.2394	0.2786
8-7-1	0.7936	0.8024	0.2907	0.2516	0.2784
8-8-1	**0.8314**	**0.8631**	0.2563	0.2203	**0.2448**
8-9-1	0.7872	0.8390	0.2640	0.2047	0.2458
8-10-1	0.7770	0.8131	0.2857	0.2074	0.2621
8-12-1	0.8003	0.8625	0.3013	0.2328	0.2804
8-14-1	0.8118	0.8387	0.2646	0.2482	0.2592
8-16-1	0.8282	0.8149	0.2668	0.2107	0.2496

5.2 Exploring MLP Model Performance with Different Training Algorithms

In the second experiment, the selected 8-8-1MLP model is trained in twenty times with four different algorithms, i.e. LM, SCG, BFGS and BR. The training results for comparing the performance of each algorithm are showed in Table 3. The network model using BR algorithm shows the best prediction. It gives the highest R^2 of 0.9152 on the test set. However, the network trained by LM algorithm gives higher R^2 with 0.8290 and 0.860 on train and test data sets, respectively. It also shows the minimum MAE (0.1638). The BFGS network trained by Quasi Newton algorithm gives the smallest RMSE (0.2334) and closes to the results from SCG and BR algorithms.

Table 3. Performance evaluation of MLP with different training algorithms.

Training algorithm	Performance measure							
	R^2		RMSE			MAE		
	Train	Test	Train	Test	All	Train	Test	All
LM	**0.8290**	**0.8618**	0.2550	0.2138	0.2421	0.1694	0.1527	**0.1638**
SCG	0.7938	0.8518	0.2563	0.1986	0.2386	0.2004	0.1435	0.1814
BFGS	0.8016	0.8575	0.2532	0.1877	0.2334	0.1910	0.1408	0.1742
BR	0.7913	**0.9152**	0.2581	0.1820	0.2355	0.1936	0.1446	0.1773

As seen in Table 3, results of all training algorithms show the similar acceptable level with higher R^2 (≥ 0.85) on the test set. So, the MLP model is suitable for the estimation of LOC. Moreover, network model is also trained with different number of neurons in the hidden layer again. It aims to explore stability of each MLP model. Performance result is measured with RMSE as shown in Fig. 2. Almost RMSE results of MLP with BR algorithm give smaller values. This indicates that the model acts more stable on the prediction performance compared to other networks. MLP with BR algorithm also shows as the best model for the software size estimation.

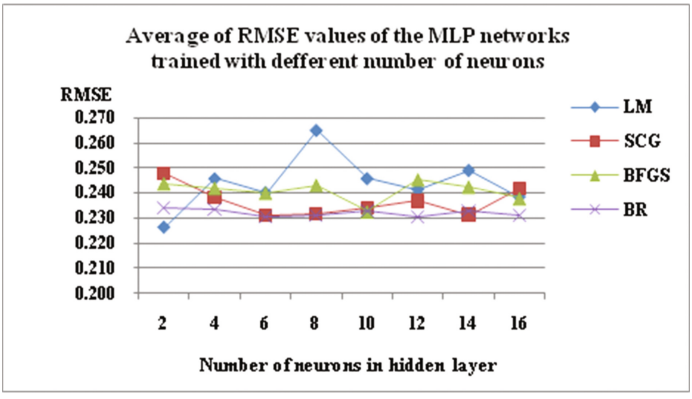


Fig. 2. Performance evaluation of MLP using different algorithms

6 Applying the Proposed Model

The proposed model can be applied to automatically estimate LOC of any software as Fig. 3. The steps for estimating LOC can be described as follows:

(1) Prepare the design model in UML Class diagram of given software by using an existing tool named ArgoUML.
(2) Convert the designed model to an XML document.
(3) Use the XML document as the input for the automated tool.
(4) The automated tool will count the values of structural complexity metrics and use the values for predicting LOC by utilizing the obtained MLP model implemented in the tool.

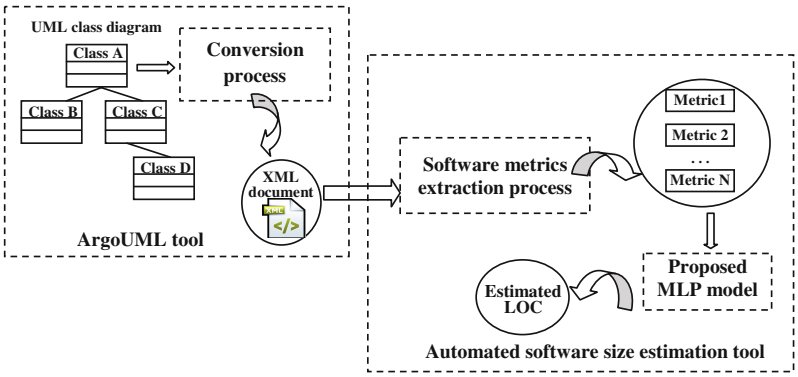


Fig. 3. A framework for applying the proposed estimation model

7 Conclusions and Future Work

This study presents an approach based on MLP neuron network to fit the software size estimation model for C++ software. A feed-forward back propagation neuron network model is constructed from 8 structural complexity matrices to predict LOC. Four learning algorithms, namely LM, SCG, BFCG and BR are adopted to train the developed ANN model. According to results of the study, they provide appropriate performance evaluation with R^2 measurement. All R^2 values are greater than 0.85 on the test set. Consequently, the constructed MLP model is played as a good prediction system for LOC estimation. Especially, BR neuron network model gives the highest quality on the prediction. It shows the stable performance even if the network is trained in the different number of neurons in the hidden layer. The given average R^2 result for BR on tested data is greater than 0.9. The comparison study on other types of neural network such as Generalized Regression Neural Network, Radial Network Function network and Adaptive Neuro-Fuzzy Inference System (ANFIS) should be also studied in the future in order to find the higher fitting for software size estimation model.

References

1. Bhatia, S., Malhotra, J.: A survey on impact of lines of code on software complexity. In: 2014 International Conference on Advances in Engineering and Technology Research (ICAETR - 2014), pp. 1–4, Unnao (2014)
2. Tashtoush, Y., Al-Maolegi, M., Arkok, B.: The correlation among software complexity metrics with case study. Int. J. Adv. Comput. Res. **4**(2), 414–419 (2014)
3. Zhang, H.: An investigation of the relationships between lines of code and defects. In: IEEE International Conference on Software Maintenance (ICSM 2009), pp. 274–283, Edmonton (2009)
4. Bhati, G.K., Pooja: An approach for software size estimation. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **4**(2), 19–25 (2014)
5. Jain, S., Yadav, V., Singh, R.: An approach for OO software size estimation using predictive object point metrics. In: 2014 International Conference on Computing for Sustainable Global Development (INDIACom), pp. 421–424 (2014)
6. Harizi, M.: The role of class diagram in estimating software size. Int. J. Comput. Appl. **44**(5), 31–33 (2012)
7. Baboo, S.S., Shereef, I.K.: An efficient weather forecasting system using artificial neural network. Int. J. Environ. Sci. Dev. **1**(4), 321–326 (2010)
8. Gurusen, E., Kayakutlu, G., Daim, T.U.: Using artificial neural network models in stock market index prediction. Expert Syst. Appl. **38**(8), 10389–11039 (2011)
9. Paswan, R.P., Begum, S.A.: MLP for prediction of area and rice production of upper Brahmaputra valley zone of Assam. In: The 4th International Conference on Computing, Communications and Networking Technologies (ICCCNT), pp. 1–9 (2013)
10. Capilla, C.: Prediction of hourly Ozone concentrations with multiple regression and multilayer perceptron models. Int. J. Sustain. Dev. Plann. **11**(4), 558–565 (2016)
11. Aggarwal, K.K., Singh, Y., Kaur, A., Malhotra, R.: Application of artificial neural network for predicting maintainability using object-oriented metrics. Int. J. Comput. Electr. Autom. Control Inf. Eng. **2**(10), 3552–3556 (2008)
12. Subitsha, P., Rajan, J.K.: Artificial neural network models for software effort estimation. Int. J. Technol. Enhancements Emerg. Eng. Res. **2**(4), 76–80 (2014)
13. Li, H.F., Cheung, W.K.: An empirical study of software metrics. IEEE Trans. Softw. Eng. **13**(6), 697–708 (1987)
14. Kiewkanya, M., Surak, S.: Constructing C++ software size estimation model from class diagram. In: 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE), pp. 1–6, Khon Kaen (2016)
15. Sola, J., Sevilla, J.: Importance of input data normalization for the application of neural networks to complex industrial problems. IEEE Trans. Nucl. Sci. **44**(3), 1464–1468 (1997)
16. Jayalakshmi, T., Santhakumaran, A.: Statistical normalization and back propagation for classification. Int. J. Comput. Theor. Eng. **3**(1), 89–93 (2011)
17. Nayak, S.C., Misra, B.B., Behera, H.S.: Impact of data normalization on stock index forecasting. Int. J. Comput. Inf. Syst. Ind. Manage. Appl. **6**, 257–269 (2014)

Data Mining Applications

Data Driven Prediction of Dengue Incidence in Thailand

Nirosha Sumanasinghe[1(✉)], Armin R. Mikler[1], Jayantha Muthukudage[2], Chetan Tiwari[3], and Reynaldo Quiroz[1]

[1] Department of Computer Science, College of Engineering, University of North Texas, Denton, TX, USA
Nirosha.lk@gmail.com, armikler@gmail.com
eynaldoquirozheredia@my.unt.edu
[2] Department of Statistics and Computer Science, University of Peradeniya, Peradeniya, Sri Lanka
jayantha.lk@gmail.com
[3] Department of Geography and the Environment, University of North Texas, Denton, TX, USA
chetan.tiwari@gmail.com

Abstract. Communicable diseases such as dengue pose a significant threat on public health across the world. Modeling an accurate and efficient prediction of dengue disease will improve public health response planning to outbreaks. However, despite the fact that many researches has focused on dengue prediction, it has been lacking geographical variation of dengue fever taken into account. Dengue is a mosquito-borne virus that annually infects over 400 million people worldwide. The infection pattern is different from region to region. We developed a model for predicting dengue fever for four provinces of Thailand with geographical variation taken into account. These predictions show slightly varying outcomes across provinces. Support Vector Regression (SVR) was used as the modeling tool. Additionally, we introduced a novel method of assessing regression model in terms of accuracies over Mean Square Error (MSE) which does not capture the behavior of data pattern spatially. This novel method resulted in 71% accuracy of prediction for Kamphaeng Phet province. The proposed model of prediction facilitates administrative bodies to make informed decisions in the context of public health of Thailand.

Keywords: Dengue prediction · Regression analysis of dengue · Prediction accuracy of regression

1 Introduction

Dengue is a mosquito born disease caused by dengue virus. The principle vector is *Aedes Agypti. Aedes Albopictus* is also responsible for spreading the dengue virus [1, 2]. These mosquitoes live around tropical and subtropical area around the world [3]. Dengue has become a major global threat. Within last 50 years it has been increased 30 fold. Almost half of the population is at risk for Dengue fewer (DF) and Dengue hemorrhagic fever (DHF). Each year 50 to 500 million people are infected and

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_9

10000 to 20000 people are dead due to dengue fever. The first dengue case in Thailand was reported in 1958 [4]. Until now dengue cases are increased and several outbreaks of dengue occurred. At present dengue is a major public health threat to Thailand. Dengue fever is mostly asymptomatic. But dengue virus can cause DF or more severe DHF or Dengue Sick Syndrome (DSS) [4].

Special programs have been implemented at national level to control dengue epidemic situation as the mortality rate is high. Currently, there are many research programs conducted to address the nationally important issue. Some programs take interdisciplinary approach to look into the problem in many different perspectives.

There is no successful vaccination for the disease. Vector control and human mosquito contact avoidance are the main controlling strategies use at present. Risk area identification is very important in controlling vectors [5–7]. Dengue prediction is carrying an equal importance as controlling dengue disease. This study aims to predict dengue cases based on rainfall, temperature and population densities of each province of Thailand. Initially, four provinces of Thailand (Kamphaeng Phet, Nakhon Sawan, Uthai Thani and Phichit) have been selected to conduct this study. A regression model with slight variant (Support Vector Regression - SVR) was used as the modeling tool of the dengue epidemic.

2 Related Work

The main goal of researchers working in the field of dengue epidemic prediction is to achieve a higher accuracy of prediction of dengue cases. Prediction accuracy depends on the selection of factors that are influencing dengue vector dynamics. There are several paths that research works progress on. Finding main influencing factors, prediction and model evaluation for the best fit are some of them to name. Machine learning techniques are primarily used in the modeling and the prediction of dengue epidemics. Variation in training gives different accuracies of the prediction. We investigate major studies of dengue epidemic analysis and prediction in recent years in the following section.

A comparison of DHF cases prediction methods of SVM, Neural network, Decision tree, and K-nearest neighbor is reported in [8]. Infection rate in the *Aedes aegypti* female mosquito with climate factors were used in model training. Authors reported that SVM with Radial Basis Function (RBF) outperform Support Vector Machine with linear kernel and polynomial kernel. They also compare the model accuracy of Decision Tree, K-Nearest Neighbor (KNN) and Neural Network Model (NNM). They concluded that SVM-R has a higher prediction power as measured by accuracy (96.26%), sensitivity (0.8747), and specificity (0.8747). Climate and mosquito data for three provinces (Nakhon Pathom, Ratchaburi, and Samut Sakhon) in Thailand from 2007 to 2013 were used in model training.

A Least Square Support Vector Machine model is used to predict dengue outbreak in the work reported in [9]. The study was conducted for five districts of Selangor for a 5 year period. The performance of SVM model and ANN model compared in the study. Authors reported that SVM were outperformed ANN in prediction accuracy. SVM generated 86.84% accuracy and ANN generated 65.58% of accuracy. They also reported that the prediction speed is also a winner in SVM. Only the rainfall data was

used as model training data which we consider as a weak point in the study. They could have increase the accuracy by employing multivariable classifiers with additional factors such as population, temperature, etc.

Authors of study reported in [10] used ANN to predict dengue confirmed cases in Singapore. Data set consist of weekly data from 2001 to 2007. Mean temperature, mean relative humidity and total rainfall were used as influencing factors to predict dengue confirmed cases. Authors claimed that the usage of previous month dengue case data as an input variable increased the prediction accuracy. The correlation of the model reported as 0.91 and the root mean square error was reported as 50.7. They also claimed that the model is independent of time and space factors of dengue cases.

In [11], A C-Support Vector Machine was used to predict dengue cases. The authors used epidemic location, air temperature and daily precipitation data as input variables. Dengue fever weekly cases and weather data from 2014 to 2015 were taken for the experiment. They used linear optimization and RBF kernel function of C-SVM. RBF kernel, after parameter optimization gives a better accuracy than linear model. They could achieve a higher accuracy by changing the split ratio of training data set and test data set (ratio 6:4 gives over 99% accuracy while ratio 7:3 gives 92% for the RBF). The dataset contained 52 data points which was divided into five categories. This results in a few numbers of data points for the SVM. A few number of data points forces SVM to settle in an over fit model.

Authors in [12] have used SVR for dengue case prediction. They have used wavelet transformation for data preprocessing and SVM along with GA for feature selection. The prediction model was tested with dengue weekly data from 2001 to 2006 in Singapore. They compared SVR results with simple linear regression. To assess the model, they used MSE, Coefficient of Determination (R2). Even though they claimed SVR out performs LR, the lowest LSE (0.083) and the highest R2 (0.96) came from LR. This is somewhat controversial.

Author in [13] approached dengue prediction in a different way from conventional classification techniques. They utilized online search query terms to forecast the dengue cases. Online queries those have dengue related terms taken into consideration as machine learning inputs. In addition, weather data related terms were also taken as input as they have a strong correlation to dengue cases.

None of the studies mentioned above have considered geographical variations in each study region. Geographical variation plays a major role in dengue cases reported in that region. We take into consideration that the geographical weight in each region in predicting dengue cases.

The remainder of the paper is organized as follows: Sect. 3 explains our approach of predicting dengue cases. Section 4 presents results of the study and conclude the paper with our discussion on dengue prediction.

3 Materials and Methods

3.1 Data Gathering

The number of dengue cases reported depends on various factors such as rainfall, temperature, population density, waste management efficiency, land use, and water

body management etc. In this study, only rainfall, temperature, and population densities are considered. These factors are gathered from various sources based on the availability. The following sections explain our strategies of obtaining each factor.

Rainfall Data. Rainfall data was obtained from Global Rainfall Map in Near Real Time (GSMaP_NRT) distributed from JAXA Global Rainfall Watch, which was developed based on activities of the GSMaP (Global Satellite Mapping of Precipitation) project. The GSMaP project is promoted for the study "Production of a high-precision, high-resolution global precipitation map using satellite data," sponsored by Core Research for Evolutional Science and Technology (CREST) of the Japan Science and Technology Agency (JST) [14]. GSMaP_NRT repository provides hourly rain rate data in 0.1° resolution (10 km at the equator). Repository divides the globe into 15 distinct regions as shown in Fig. 1 and provides rainfall data separately for each region as Comma Separated Values (CSV) files. Registered users get free access to the repository. The users can get data using an FTP client that is connected to the repository using credential provided by the repository management. Thailand is included in 02_AsiaSE area. Table 1 lists location specific information for each Asian region. For the model training, data from five consecutive years was used.

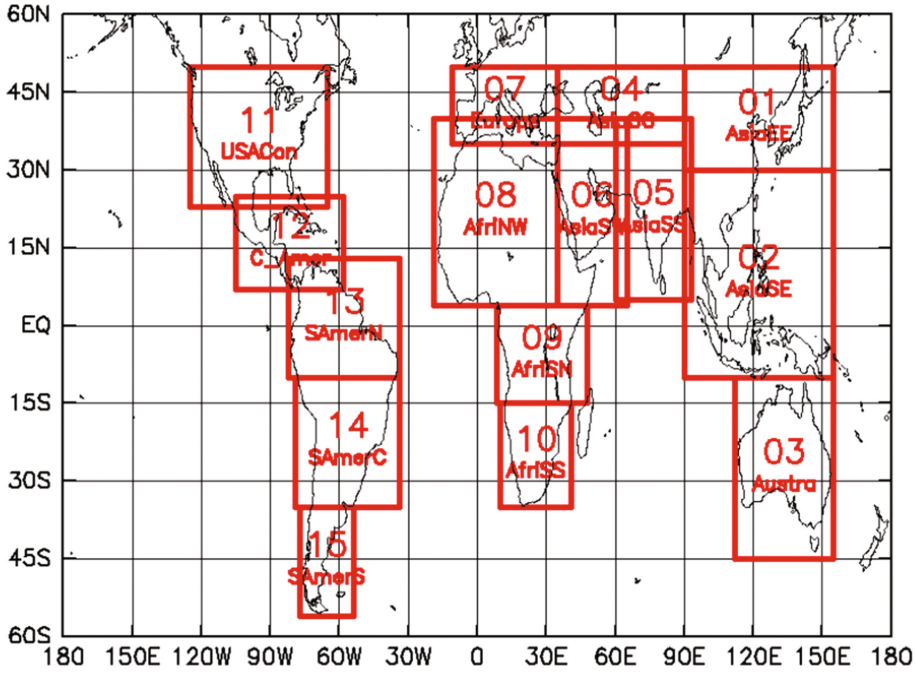


Fig. 1. Definition of text areas of JAXA data repository for text data [14]

Table 1. GSMap text area declaration for Asian region [14]

Area name	Lon (W)	Lon (E)	Lat (S)	Lat (N)	Description
01_AsiaEE	90	155	30	50	East Asia
02_AsiaSE	90	155	−10	30	South East Asia
04_AsiaCC	35	90	35	50	Central Asia
05_AsiaSS	60	93	5	40	South Asia
06_AsiaSW	35	65	4	40	Arabian Peninsula and East Africa

Temperature Data. Temperature data was obtained from Thai Meteorological Department (TMD). Average temperature value for each month for each district was used in the training. Time span of temperature data is five consecutive years.

Population Data and Dengue Case Data. Population and dengue case data for each district for five consecutive years were obtained. Dengue case data is given in three groups which are Dengue Hemorrhagic Fever (DHF), Dengue Fever (DF) and Dengue Shock Syndrome (DSS). We combine all three categories to form a single entity and used in model training as dengue cases. Dengue case data was obtained from Department of Disease Control, Ministry of Health.

3.2 Data Processing

Extracting Relevant Data and Alignment of Time Resolution. GSMaP_NRT region 02_AsiaSE covers a larger area than Thailand geographical region (Fig. 1). This results in large amount of non-related data being loaded into the spatial database making it heavy for fast computations. To reduce the data load overhead, only the rainfall data that falls inside Thailand geographical area was obtained by cropping the dataset using longitude and latitude. Non-relevant data was discarded. As the time resolution of rainfall data is one reading per hour, it is required to compute the monthly rainfall data from hourly data. This matches the time resolution of each factor before use in training process as temperature and population data recorded monthly basis. Further, there are multiple observation points fallen in a single district as shown in Fig. 2. The average accumulated value of all the points that fall in a district was taken as the monthly rainfall of that district. The unit of recording is mm per hour (mm/h). Sample data file format for rainfall from GSMaP_NRT is given in Table 2.

No pre-processing was conducted on temperature and population data. Each factor has outliers. A careful decision has to be made on outliers as these may be a result of an outbreak rather than just an outlier. An outbreak can be a result of many other factors and rainfall may or may not contribute to the outbreak. As a model generalization step, we do not consider extreme values as outliers in this study. We keep further investigations on this scenario to be covered in future works of this study.

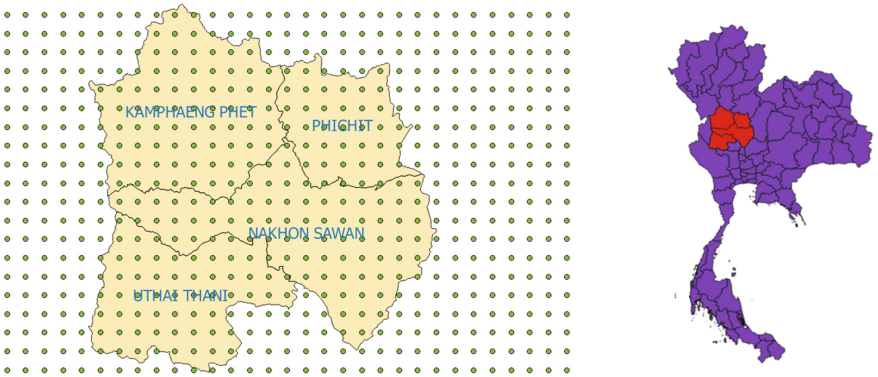


Fig. 2. Rain fall data observation points and geographical boundaries of all four provinces

Table 2. Fragment of rainfall data text file from GSMap_NRT

Lat	Lon	Rain rate
20.95	97.05	0.1
20.85	97.05	0.06
20.75	97.05	0.04
20.65	97.05	0.06

3.3 Pre-analysis of Data

The proposed model used in this study is the SVR [15]. SVR is built on regression analysis. To get a better result from a regression analysis, there must be a positive correlation between explanatory variables (factors) and dependent variable (dengue cases). As the primary model of prediction is SVR, this study needs a data analysis before moving forward with SVR. A separate correlation analysis was conducted for each factor (rain, temperature and population) to determine the suitability of the regression analysis of the proposed factors. Correlation is a statistical relationship between those two sets of data which describes the strength of the relationship between those two data sets in consideration. If the correlation is low there is a weak interdependency between those two sets. If the correlation is high (normally greater than 0.5 negative or positive), there is a considerable relationship between those two sets. Correlation of two data sets is computed as given in the equation below.

$$\rho_{X,Y} = \frac{E[(X - \mu X)(Y - \mu Y)]}{\sigma X \sigma Y} \tag{1}$$

where $\rho_{X,Y}$ is the correlation between datasets X and Y. E is the expected value operator. μX is the mean of data set X, μY is the mean of data set Y. σX and σY are standard deviation of data sets X and Y respectively. The correlation value is generally interpreted as shown in Table 3.

Table 3. Correlation values and their meanings

Correlation value	Interpretation
−1	A perfect downhill (negative) linear relationship
−0.7	A strong downhill (negative) linear relationship
−0.5	A moderate downhill (negative) relationship
−0.3	A weak downhill (negative) linear relationship
0	No linear relationship
+0.3	A weak uphill (positive) linear relationship
+0.5	A moderate uphill (positive) relationship
+0.7	A strong uphill (positive) linear relationship
+1	A perfect uphill (positive) linear relationship

3.4 Model Generation

Behavior of each factor (rainfall, temperature and population density) on dengue cases is spatially dependent. That is the effect of rainfall on dengue for each district is different from district to district as presented in [5]. Hence, a separate analysis for each district was conducted and a separate model for each province was generated. Data for 5 years were combined together for each district and fed into the model for training. The proposed arrangement can capture the spatial heterogeneity of each province and hence improve the performance of prediction model.

The SVR model is based on the regression analysis. A regression analysis can estimate the relationship between two data sets (random variable) and fit a cure to the data sets (explanatory variable and dependent variable). This curve can then be used in prediction of unknown cases. The regression curve for this study has three explanatory variable, Rain R, Population P, and Temperature T. The regression model for this study is given in the equation below.

$$C_{di} = \beta_0 + \beta_1 P_i + \beta_2 R_i + \beta_3 T_i + \varepsilon \tag{2}$$

where Pi is the population in *i*th region, Ri is the rainfall for *i*th region and T_i is the temperature for *i*th region. The error term is ε. C_{di} is the dengue cases for region *i*. Intercept is β_0, a constant.

SVR improves the detection speed as it keeps only a subset of training data as support vectors in the model. The SVR uses the same principles as the Support Vector Machine (SVM) for classification, with only a few minor differences. SVR's output is a real number which makes it difficult to match target output on test dataset. A margin of tolerance (epsilon) is set in approximation to the SVM to address the problem associated with real numbers output. General construction of SVR is given in the following equations.

SVM regression is constructed by first mapping the input vector X into an m-dimensional feature space using a non-linear mapping function. The linear regression model is then constructed in this feature space. The linear model $f(x, \omega)$ is given by Eq. 3.

$$f(x,\omega) = \sum_{j=1}^{m} w_j g_j(x) + b \tag{3}$$

where $g_j(x), j = 1, \ldots m$ denotes a set of nonlinear transformations and b is the "bias" term. The bias term can be dropped with the assumption of zero mean data set. ω is the normal vector.

The quality of estimation is measured by the loss function $L_\varepsilon(y, f(x,\omega))$ given in Eq. 4. The loss function is computed as proposed in [16].

$$L_\varepsilon(y, f(x,\omega)) = \begin{cases} 0 & if \quad |y - f(x,\omega)| \le \varepsilon \\ |y - f(x,\omega)| - \varepsilon & otherwise \end{cases} \tag{4}$$

Then the empirical risk function is given in Eq. 5.

$$R_{emp}(\omega) = \frac{1}{n} \sum_{i=1}^{n} L_\varepsilon(y, f(x,\omega)) \tag{5}$$

The model generated by minimizing the ω^2. This can be achieved by introducing (non-negative) slack variables $\xi_i, \xi_i^*\ i = 1, \ldots n$ to measure the deviation of training samples outside ε-insensitive zone. Thus the SVM regression is formulated by minimization of the function given in Eq. 6.

$$\arg\min_{(\omega)} = \frac{1}{2}\|\omega\|^2 + C \sum_{i=1}^{n} \left(\xi_i + \xi_i^*\right) \tag{6}$$

Such that

$$\begin{cases} y_i - f(x_i,\omega) \le \varepsilon + \xi_i^* \\ f(x_i,\omega) - y_i \le \varepsilon + \xi_i \\ \xi_i, \xi_i^* \ge 0, \quad i = 1, \ldots, n \end{cases}$$

This optimization problem can transformed into the dual problem and its solution is given by the equation in 7.

$$f(x) = \sum_{i=1}^{n_{sv}} \left(\alpha_i - \alpha_i^*\right) K(x_i, x) \tag{7}$$

Such that: $0 \le \alpha_i^* \le C$, $0 \le \alpha_i \le C$.

Where n_{sv} is the number of Support Vectors (SVs) and the kernel function is given by the equation in 8.

$$K(x_i, x) = \sum_{j=1}^{m} g_j(x) g_j(x_i) \tag{8}$$

The RBF was used as the kernel function and epsilon was set to 0.001. The cost parameter was kept at 100.

3.5 Prediction

A vector of unseen data for rainfall, temperature and population data is fed into the trained model and estimated output is obtained from the SVR. This output is not a label as in SVM. Rather it is a real number approximating the number of dengue cases pertaining to the given scenario. The number of cases predicted show cases the severity of the condition that may occur if the given scenario appears in the future.

3.6 Model Validation

Conventional regression models are evaluated based on the MSE of the cross validation (mostly 10-fold cross validation). MSE cannot capture the total picture of the behavior of the data set. Several outliers can affect the final outcome of the validation. Another problem of regression analysis is there is no way of computing the accuracy of the prediction with cross validation. Regression gives real values as estimates and there is theoretically infinite number of possibilities with a real number. This fact makes it impossible to compare against target value. Accuracy is computer as per the following equation in SVM like classifiers. SVM like classifiers are based on class labels and hence makes it possible to compute accuracies conveniently.

$$\mathrm{ACC} = \frac{(\mathrm{TP} + \mathrm{TN})}{(TP + TN + FP + FN)} \tag{9}$$

where ACC is accuracy, TF is true positives, TN is true negative, FP is false positive and FN is false negative. In this study a novel yet simple accuracy calculation method was introduced. A positive confidence boundary parameter α was included in cross validation. If $|actual\ value - estimated\ value| > \alpha$, we label the estimated value as a correct prediction and incorrect prediction otherwise.

Determination of the Degree of Fit of the Regression Model to the Dataset with Parameter Alpha (α). The value of α is inversely proportional to the model accuracy. If the model generate a higher accuracy value for a lower value of α the regression model fits relatively well to the dataset. If the model accuracy is high only for a large value of α the dataset is loosely correlated to the influencing factors. Higher accuracy for a smaller alpha value indicates that the dataset and the fitted regression is a best fit for the problem domain.

4 Results

4.1 Data Pre-analysis

Analysis of data to find the degree of relationship between influencing factors and the dengue cases revealed strong relationship between them. This enabled this study to move forward with regression analysis. Results of the correlation analysis are given in the Table 4.

Table 4. Correlation analysis results for four provinces for dengue cases and rainfall data

Year	District			
	Kamphaeng Phet	Nakhon Sawan	Pitchit	Uthai Thani
2007	0.62	0.54	0.61	0.61
2008	0.71	0.75	0.84	0.50
2009	0.81	0.59	0.46	0.26
2010	0.91	0.82	0.64	0.87
2011	0.62	0.61	0.70	0.58

We also conducted a correlation analysis for all 5 years together. The result was less correlated than individual year. Nevertheless there is a good correlation between factors for all 5 years as well. The correlation results for full data set for each province are given in the Table 5.

Table 5. Correlation analysis results for four provinces for dengue cases and rainfall data for all 5 years together

	Kamphaeng Phet	Nakhon Sawan	Pitchit	Uthai Thani
All 5 years	0.60	0.53	0.57	0.40

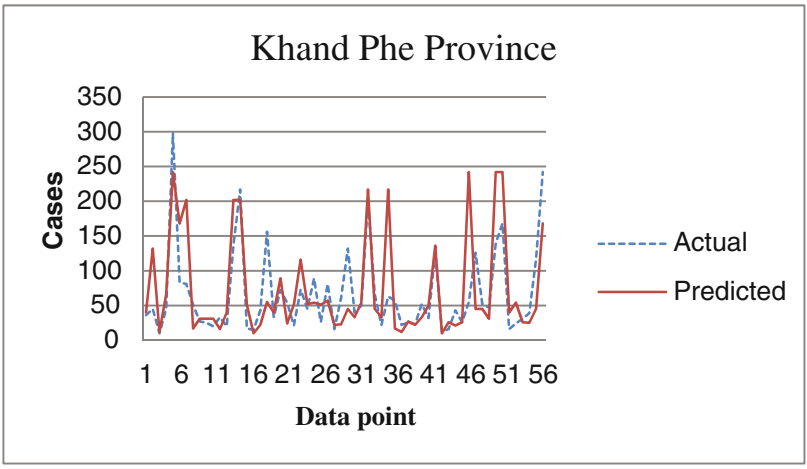
4.2 Cross Validation

We conducted 10-fold cross validation to assess the performance of the model. A novel method of summing the results of all ten folds was introduced in the methodology Sect. 3.6. We present the results of our novel method computing accuracies of each fold below. Two trials were run for provinces Kamphaeng Phet and Uthai Thani those have the highest and the lowest correlation values. Other two provinces generated a medium accuracy that is between Kamphaeng Phet and Uthai Thani accuracies. Therefore we present only the highest (Kamphaeng Phet) and the lowest (Uthai Thani). The corresponding accuracies for each trial and for each province are listed in Table 6.

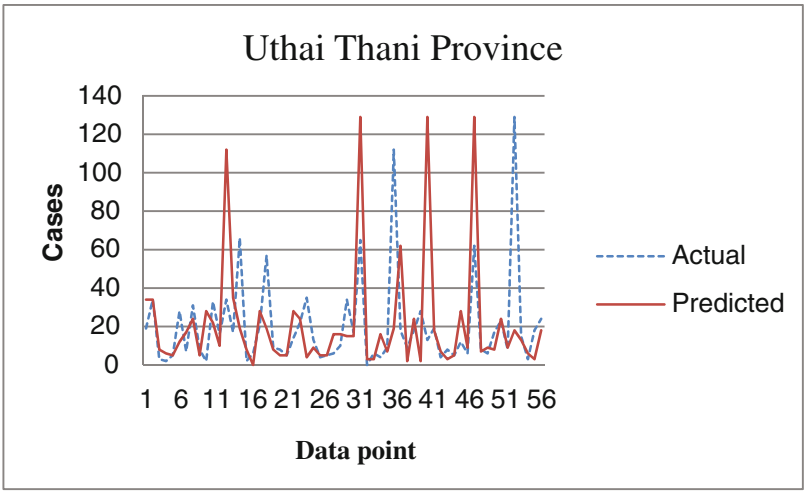
Table 6. Accuracies for 10-fold cross validation and average accuracy

Fold	Kamphaeng Phet		Uthai Thani	
	Alpha = 50		Alpha = 50	
	Trial 1	Trial 2	Trial 1	Trial 2
1	100	66.6	50.0	83.0
2	100	83.3	50.0	83.0
3	66.6	66.6	16.0	66.0
4	66.6	83.3	66.0	50.0
5	66.6	50.0	66.0	50.0
6	66.6	66.6	50.0	50.0
7	33.3	50.0	83.0	83.0
8	83.3	66.6	83.0	50.0
9	50.0	100	66.0	50.0
10	83.3	83.3	66.0	83.0
Average	71.63	71.63	59.6	64.8

In addition, we visualize the actual case data against the estimated value of the model during the cross validation. The results of the visualization are given in Fig. 3.



(a)



(b)

Fig. 3. Actual values and predicted values during 10-fold cross validation process for (a) Kamphaeng Phet and (b) Uthai Thani provinces.

5 Discussions

In this study, we proved that rainfall along with temperature data and population densities can predict the dengue cases with a good accuracy level (71%). A novel method of accuracy calculation for regression analysis was also introduced. This

method enabled us to measure the performance of the model in terms of accuracy over MSE for regressions. A comparison of novel accuracy and MSE is out of context as two methods measure the performance in two different angles. MSE is largely affected by great outliers and novel method of calculating accuracy is not affected in large by such outliers as it treated them as incorrect predictions. We also revealed that the geographical variation of the study regions influences the final outcome of the prediction model. Hence a local models for each geographically distinct region is needed. Although global models generated higher accuracies for prediction that incorrectly interpret the actual behavior of the disease. We strongly suggest using local analysis over global analysis to better understand the behavior of the disease. A higher accuracy should be obtained (greater than 71%) through thorough investigation of the scenario in different point of views. We kept it as a future development for this study.

There is no strong correlation among temperature, population data and the dengue cases. Temperature or population density along cannot predict dengue behavior according to our data sets. According to the test results and visualization, it is clear that there are other factors beside factors considered in this study influence the number of cases reported. It is clear from later part of the result section that rainfall cannot explain the sudden hikes on dengue cases as seen in Fig. 3. It is inappropriate to consider those hikes as outliers as these data are reported after careful investigation and laboratory confirmation. There must be a mechanism to identify the causes of these hikes as it may be a sign of dengue outbreak. We keep it too as a future work of this study. And also we proposed to conduct the same study for the entire country to better understand the situation of dengue disease in Thailand.

Acknowledgements. The work described was partly supported by Grant Number NIH 1R01LM011647-01 from the National Institutes of Health (PI: Mikler, AR).

References

1. Hay, S.I., Myers, M.F., Burke, D.S., Vaughn, D.W., Endy, T., Ananda, N., et al.: Etiology of interepidemic periods of mosquito-borne disease. In: Proceedings of National Academy of Sciences, USA, pp. 9335–9339 (2000)
2. WHO Dengue: Guidelines for Diagnosis, Treatment, Prevention and Control. WHO and TDR Publication, France (2009)
3. Gubler, D.J.: Dengue and dengue hemorrhagic fever. Clin. Microbiol. **11**(3), 480–496 (1998)
4. UNDP/World Bank/WHO Special Programme for Research and Training in Tropical Diseases. Report of the Scientific Working Group meeting on dengue Geneva (2006)
5. Sumanasinghe, N., Mikler, A., Tiwari, C., Muthukudage, J.: Geo-statistical dengue risk model using GIS techniques to identify the risk prone areas by linking rainfall and population density factors in Sri Lanka. Ceylon J. Sci. **45**(3), 39–46 (2016)
6. Mammen Jr., M.P., Pimgate, C., Koenraadt, C.J.M., Rothman, A.L., Aldstadt, J., Nisalak, A., et al.: Spatial and temporal clustering of dengue virus transmission in Thai villages. PLoS Med. **5**(11), e205 (2008)

7. Jeefoo, P., Tripathi, N.K., Souris, M.: Spatio-temporal diffusion pattern and hotspot detection of dengue in Chachoengsao Province, Thailand. Int. J. Environ. Res. Public Health **8**(1), 51–74 (2011)
8. Kesorn, K., Ongruk, P., Chompoosri, J., Phumee, A., Thavara, U., Tawatsin, A., et al.: Morbidity rate prediction of dengue hemorrhagic fever (DHF) using the support vector machine and the *Aedes aegypti* infection rate in similar climates and geographical areas. PLoS ONE **10**(5), e0125049 (2015)
9. Yusof, Y., Mustaffa, Z.: Dengue outbreak prediction: a least squares support vector machines approach. Int. J. Comput. Theory Eng. **3**(4), 489–493 (2011)
10. Hani, M., Aburas, B., Gultekin, C., Murat, S.: Dengue confirmed-cases prediction: a neural network model. Experts Syst. Appl. **37**(6), 4256–4260 (2010)
11. Rahmawati, D., Huang, Y.P.: Using C-support vector classification to forecast dengue fever epidemics in Taiwan. In: International Conference on System Science and Engineering (ICSSE), pp. 1–4 (2016)
12. Wu, Y., Lee, G., Fu, X., Hung, T.: Detect climatic factors contributing to dengue outbreak based on wavelet, support vector machines and genetic algorithm. In: World Congress on Engineering, London, UK (2008)
13. Chartree, J., Angel, B., Jimenez, T., et al.: Predicting dengue incidence in Thailand from online search queries that include weather and climatic variables. In: Text Mining of Web-Based Medical Content, pp. 77–106. De Gruyter, Berlin (2014)
14. Kubota, T., Kachi, M., Oki, R., Ushio, T., Shige, S., Aonashi, K., Okamoto, K.: Near-real-time global rainfall map using multi-satellite data by JAXA and its validation. In: American Geophysical Union. Fall Meeting (2010)
15. Alex, J.S., Bernhard, S.: A tutorial on support vector regression. Stat. Comput. Arch. **14**(3), 199–222 (2004)
16. Cortes, C., Vapnik, V.N.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)

A Comparative Analysis of Bayesian Network and ARIMA Approaches to Malaria Outbreak Prediction

A.H.M. Imrul Hasan[1(✉)], Peter Haddawy[1], and Saranath Lawpoolsri[2]

[1] Faculty of Information and Communcation Technology (ICT), Mahidol University, Salaya, Thailand
ahmimrul.has@student.mahidol.ac.th, peter.had@mahidol.ac.th
[2] Faculty of Tropical Medicine, Mahidol University, Bangkok, Thailand
saranath.law@mahidol.ac.th

Abstract. Disease outbreaks are important to predict since they indicate hot spots of transmission with high risk of spread to neighboring regions and can thus guide the allocation of resources. While numeric prediction models can be easily used for outbreak prediction by setting thresholds, an alternative is to build a model that specifically classifies situations into outbreak or none. In this paper we compare Bayesian network models built for the outbreak classification problem with Bayesian network, ARIMA and ARIMAX models built for numeric prediction and used for outbreak prediction by thresholding. We show that in most cases the classification models outperform the other models. We then investigate the reasons underlying the differences in performance among the models in order to shed light on their strengths and weaknesses. The models are developed and evaluated using two years of malaria and environmental data from northern Thailand.

Keywords: Bayesian networks · ARIMA · Outbreak prediction · Malaria

1 Introduction

Malaria is one of the most common and serious vector-borne infectious diseases. According to the WHO, the number of estimated malaria cases was 214 million globally in 2015, with 438,000 malaria deaths [19]. In Thailand, 31,121 and 15,446 confirmed cases were reported in 2014 and 2015, respectively [13]. Since malaria typically occurs in underdeveloped and remote areas where public health resources are often scarce, targeted intervention and resource allocation are considered essential for effective malaria control. Prediction models are an important supporting technology in order to help anticipate resource allocation needs. A variety of modeling approaches (ARIMA, regression, neural

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_10

nets, SIR models) have been used to build malaria prediction models [21]. While numeric case prediction is one common approach, another is to predict outbreaks, which are instances in which the number of cases is well above the norm. Outbreaks are important to predict since they indicate hot spots of transmission with high risk of spread to neighboring regions.

In previous work [8] we investigated the use of Bayesian networks to build models for numeric case prediction of malaria and showed the Bayes net models to be superior to traditional ARIMA models due to their ability to model non-linear effects of environmental variables. While numeric prediction models can be easily used for outbreak prediction by setting thresholds, an alternative is to build a model that specifically classifies situations into outbreak or none. In this paper we compare Bayesian network models built for the outbreak classification problem with Bayesian network and ARIMA models built for numeric prediction and used for outbreak prediction by thresholding. We show that in most cases the classification model outperforms the other models. Our previous work relied exclusively on judgment of domain experts to engineer the Bayes nets. In this paper we improve upon the methodology by using a data driven approach to determining optimal time lags and optimal sets of environmental variables to include in the model.

2 Related Work

2.1 Bayesian Epidemic Models

A Bayes net is developed by Cooper et al. [4] with 20 million nodes for anthrax outbreak detection. The entire population is modeled by a single network with each person in the network connected to the rest of the network via a node called disease status. The task is to calculate the posterior probability of the Alarm node given the inferred disease statuses of all people in the population. The work does not consider spatial or temporal aspects.

Jiang and Wallstrom [10] explore the use of Bayesian networks for Cytosporidium outbreak detection and prediction. This research focuses mainly on the shortcomings of classical methods of time-series analysis and improving them by using Bayesian networks. The model is tested on a simulated outbreak data set.

Sebastian et al. [17] produce a dynamic Bayesian network for predicting influenza like illness and the number of pneumonia and influenza deaths based on previous pediatric and adult cases of respiratory syndrome. No environmental variables are used.

A predictive dynamic graphical model is developed by Mubangizi et al. [15] for malaria case prediction. This work mainly focuses on developing a probabilistic model that combines a predictive and a diagnostic model. The diagnosis is done on images of blood sample by a computer vision technique. Their state space prediction model performs only slightly better than the baseline model.

2.2 Malaria Prediction Models

Among the numerous techniques that have been used to create predictive models, ARIMA [1] is the most popular because of it's ability to accurately model characteristics of the time series. In addition, ARIMA has a few variations incorporating seasonality (SARIMA) [9] and external explanatory variables (ARIMAX), which is also known as multivariate ARIMA.

One of the most recent works on malaria prediction [20] shows that high frequency variation is best predicted in short-term horizons (1–4 weeks). They developed ARIMAX models for six different catchment areas in Uganda. Clinical data includes previous cases, number treated and test individuals. Environmental data includes rainfall, temperature and enhanced vegetation index. Lags are determined using pre-whitening. The prediction accuracy is tested by symmetric mean absolute percentage error (SMAPE).

Another weekly model was produced by Teklehaimanot et al. [18] to predict malaria cases in 10 districts of Ethiopia using Poisson regression. The predictor includes lagged maximum and minimum temperature, rainfall and an autoregressive term based on a moving average of the number of cases 4th, 5th and 6th weeks. The accuracy was measured by calculating potentially prevented cases. The system managed to generate prediction which follows the overall pattern but undershoots the higher peaks.

A 10-day Poisson regression model was developed by Haghdoost et al. [6] on 8-years case data to predict malaria in Kahnooj district of Iran. Environmental data includes mean daily temperature, relative humidity and rainfall. The best fitted model had 1-month time lag between the meteorological variables and predicted cases. The data was divided into six year for training and 2 years for testing. The prediction accuracy was tested using mean absolute percent error (MAPE).

Buczak et al. [2] developed a malaria prediction model using fuzzy association rule mining for 64 regions of South Korea. The predictor variables were weekly malaria cases, mosquito net distribution, malaria control financing, land surface temperature, NDVI, EVI, southern oscillation index and sea surface temperature. The fuzzy Association rules produced 7 to 8 weeks ahead predictions for three incidence levels: high, medium and low. The produced rules perform significantly better than random forest, decision tree and support vector machine.

3 Methodology

3.1 Study Site and Data

The district Tha Song Yang is chosen as the study area for this research. It is the northwestern most district of Tak province in northern Thailand near to the Myanmar border. Two years of weekly clinically confirmed case (*plasmodium vivax* & *plasmodium falciparum*) data of Tha Song Yang were obtained from Thailands national E-Malaria Information System (EMIS) [11]. The data covers each of the 66 villages for the years 2012 and 2013, providing a total of

6,138 records with 12,800 total cases. The numbers of cases per village per week ranged from 0 to 82 with a mean of 2.1. For the purpose of outbreak prediction, we focus on 13 villages of high incidence which have a minimum weekly incidence of 0, maximum of 82, and mean of 7.43 cases. Outbreaks are defined in terms of mean and standard deviation. We define two thresholds in terms of severity: mean plus one and two standard deviations [18], resulting in thresholds of 16.82 and 26.18, respectively.

In addition to the case data, our model makes use of a number of environmental factors associated with malaria. The factors considered for inclusion in our model and the source for each are:

- Normalized Difference Vegetation Index (NDVI): monthly satellite data from MOD11A3,
- Land Surface Temperature (LST): monthly satellite data at 5 km resolution from MOD11C3,
- Rainfall: daily satellite data at 10 km resolution from JAXA Global Rainfall Watch,
- Slope: Average in 1 km buffer around each village, computed from elevation data, Distance to nearest stream: Euclidean distance from village center to closest point on the stream,
- Stream density: total stream length in 4 km buffer around each village, and
- Distance to border: Euclidean distance from village center to the closest point on the border with Myanmar.

The variables Distance to nearest stream, and Stream density are thought to positively impact malaria incidence. NDVI is generally correlated with malaria transmission, with some studies showing positive correlation [5] and others negative correlation [7]. LST has a nonlinear effect on malaria with malaria incidence low for low temperatures, increasing over some range, and then dropping off for high temperatures [14]. Rainfall also has a nonlinear effect, with malaria incidence increasing with rainfall until the point where the flushing effect is reached, at which point it decreases [12]. Slope is included because it interacts with rainfall, with rain draining off more quickly the higher the slope. Distance to border is a proxy for the number of imported cases and is thought to have a positive effect on incidence. Some satellite data were missing due to cloud cover. Missing values were filled in using temporal and spatial interpolation as appropriate.

3.2 Development of Bayesian Network Models

The first step in model construction is to determine the appropriate time lags for the temporal covariates LST, Rainfall, and NDVI. For that, the pre-whitening [3] process is used. Pre-whitening removes spurious correlations due to auto-regression before cross-correlation analysis. The process consists of fitting an auto-regressive model to covariate time series (X), using this to filter the dependent variable time series (Y), and calculating the cross correlation between the residuals for X and the filtered Y. This results in cross-correlation graphs.

Since the graphs can indicate several potential time lags for a given covariate, each is tested using regression to determine the one with the most predictive power. The analysis resulted in identification of optimal time lags of 6 weeks for LST, 7 weeks for Rainfall, and 8 weeks for NDVI. The continuous variables were then discretized. Initial discretizations were produced by using unsupervised binning in the Weka package with approximately equal data counts in each bin. The discretizations were then fine-tuned experimentally. Using these lagged variables we produced an initial Bayes net model using Netica. We then exhaustively examined the prediction accuracy using all combinations of the environmental variables. This resulted in the removal of the variables Rainfall and NDVI, yielding the optimal Bayesian network prediction model shown in Fig. 1. The model includes two latent variables: Stream_Effect summarizes the effect of stream distance and stream density; and Mosquito_pop_density_wi represents the effect of various environmental factors on the vector density. Inclusion of these variables increases the explanatory power of the network and, importantly, reduces the size of some of the conditional probability tables.

The Bayes net model is used for prediction by entering the known value for cases at week zero (cases_w0), LST at weeks minus 5 and minus 4 (LST_wm5, LST_wm4), and Stream Distance (STRM_DIS), Stream Density (StreamDensity), Slope (Slope), and Border Distance (BOR_DIS), and computing the posterior

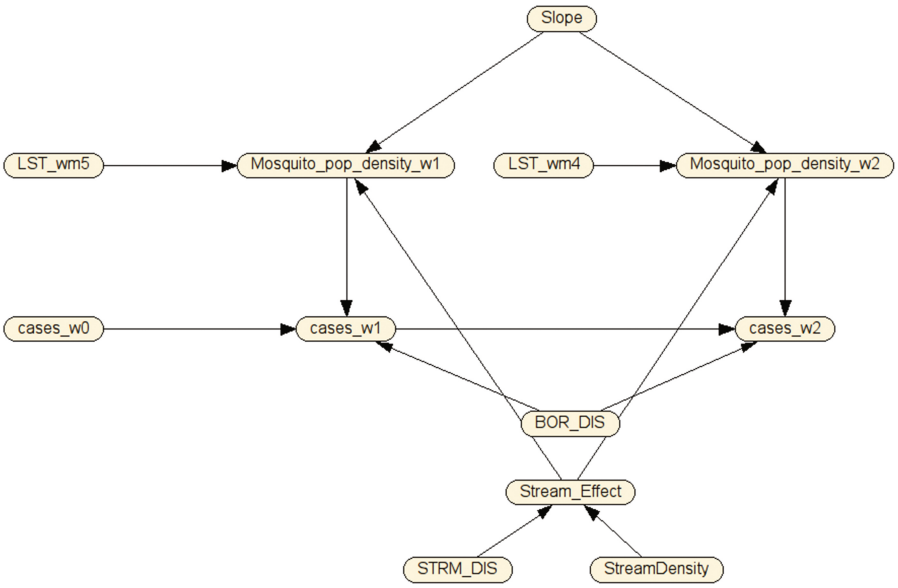


Fig. 1. Single Village Dynamic Bayesian Network (shown in Netica), with two time slice: 1 and 2 week ahead prediction. Temporal nodes represent the state of a random variable at a point in time, such as LST at week minus 5 (LST_wm5), and non-temporal nodes represent random variables whose state does not change, such as Border Distance.

probability of cases at week 1 and week 2 (cases_w1, cases_w2). The predicted number of cases is then the expected value of the cases random variable:

$$E(cases\_w_1) = \sum_{i=1}^{14} \{P(range_i) * mean(range_i)\} \quad (1)$$

where $i = 1, 2, ..., 14$ are the ranges, $P(range_i)$ is the probability of ith range and $mean(range_i)$ is the mean of the distribution of the data over the ith range.

The model in Fig. 1 is used for outbreak prediction by determining whether the predicted value is above a given threshold. For the binary outbreak classification model, one Bayes net was produced for each threshold: mean plus one and plus two standard deviations. This was done by replacing the cases nodes with outbreak nodes with states yes/no depending on whether the number of cases was above or below the particular threshold. All Bayes nets were trained on 70% (65 weeks) of consecutive data for all 66 villages.

3.3 Development of ARIMA Model

For building the ARIMA model 70% contiguous cases were collected from each village separately and concatenated by padding with null values between the village time series. The null values are needed to prevent treating different villages time series as one. The exact number of null values is chosen so as to keep the seasonality intact. In this way, the time series of the 66 villages are combined into a single time series. Then the auto.arima() function of the forecast R software package v3.2.3 was used to obtain the optimal ARIMA model based on the available data. The auto.arima() function executes an optimal model choosing algorithm based on Akaike information criterion (AIC) and Bayesian information criterion (BIC). The notation of the found optimal model is ARIMA (0, 1, 0)(1, 0, 0) with a frequency of 52, which means the data series required first order differencing (d = 1) to make the series stationary and no autoregressive term (p = 0) to determine the influence of the previous week's cases on the current weeks cases. There is no influence (q = 0) of the previous week residuals on the current week. In addition the model shows the seasonality (P = 1) with a single seasonal autoregressive term. An ARIMAX model was also developed by following a similar process. The model fitting algorithm chose LST with a lag of six weeks as the only external variable to include in the model to obtain the best fit.

4 Results and Discussion

The Bayes net, ARIMA, and ARIMAX models were tested on the 13 high incidence villages with the remaining 30% (28 weeks) of the data. For this binary classification problem, ROC analysis is the commonly used evaluation technique. Tables 1 and 2 show the ROC AUC values for the Bayes net, ARIMA, and binary Bayes net models for the thresholds mean + std. dev. and mean + 2std. dev.,

Table 1. AUC values of ROC for Outbreak threshold: mean + Std. dev.

Models	Week1	Week2	Week3	Week4	Week5	Week6
Bayes net	0.945	**0.956**	0.952	0.923	**0.911**	0.894
ARIMA	**0.951**	0.94	0.935	0.921	0.896	0.863
Binary Bayes net	0.932	0.929	**0.958**	**0.943**	0.91	**0.933**

Table 2. AUC values of ROC for Outbreak threshold: mean + 2 * Std. dev.

Models	Week1	Week2	Week3	Week4	Week5	Week6
Bayes net	0.983	0.977	0.974	0.964	0.969	0.952
ARIMA	**0.992**	0.976	0.966	0.959	0.939	0.893
Binary Bayes net	0.981	**0.983**	**0.984**	**0.966**	**0.977**	**0.979**

respectively. The best accuracy for each week is shown in bold font. The ARIMAX model did not outperform the other models for any prediction horizon, so its values are not shown.

For both the thresholds ARIMA performs best for one week prediction. This is expected since ARIMA considers only lagged cases which have a strong influence on future cases for short horizons. But at longer prediction horizons the Bayes net models (both continuous and binary) perform better. At the lower outbreak threshold, the binary Bayes net performs best for 3, 4, and 6 week predictions. At the higher threshold, it performs best for all horizons greater than one week.

The dataset for this experiment is highly skewed with only 7.7% of the weeks in the test set containing outbreaks at the lower threshold and 3.8% of the weeks containing outbreaks at the higher threshold. This is common in outbreak prediction problems. Since the number of negative instances for this problem is far greater than the number of positive instances, precision recall (PR) curves can provide a more informative measure of performance than ROC curves [16]. Tables 3 and 4 show the AUC values of the PR curves for the two thresholds. The results in terms of best performing model using PR analysis for the higher outbreak threshold are identical to those using ROC analysis. For the lower threshold the results are similar but not identical. In particular, ARIMA now

Table 3. AUC values of PR for Outbreak threshold: mean + Std. dev.

Models	Week1	Week2	Week3	Week4	Week5	Week6
Bayes net	0.651	0.684	0.602	**0.602**	**0.535**	0.509
ARIMA	**0.761**	**0.714**	0.617	0.512	0.353	0.251
Binary Bayes net	0.614	0.614	**0.642**	0.576	0.481	**0.531**

Table 4. AUC values of PR for Outbreak threshold: mean + 2 * Std. dev.

Models	Week1	Week2	Week3	Week4	Week5	Week6
Bayes net	0.634	0.648	0.544	**0.7**	0.583	0.589
ARIMA	**0.869**	0.772	0.646	0.496	0.305	0.229
Binary Bayes net	0.828	**0.811**	**0.796**	**0.7**	**0.703**	**0.629**

performs best for one and two week prediction, the Bayes net best for 4 and 5 week prediction, and the binary Bayes net best for 3 and 6 week prediction.

For the lower outbreak threshold there is not clear winner between the binary and numeric Bayes nets, which for the higher outbreak threshold the binary Bayes net clearly out performs the other except in week 1. This is likely due to the fact that the numeric Bayes net discretizes the range of the number of cases and due to the small numbers of high incidence weeks must use a relatively wide bin size at the upper end. This can cause inaccuracies in numeric prediction at the upper end. The binary Bayes net does not suffer from this problem.

In both the ROC and PR analyses, the Bayes nets outperform the ARIMA model for longer time horizons, with the difference in performance increasing with the time horizon. This is likely due to the fact that the influence of the environmental variables in the Bayes nets relative to the influence of cases in week zero increases with increasing time horizon. To test this we carried out a sensitivity analysis on the two types of Bayes nets. In the case of the numeric prediction Bayes net, sensitivity analysis shows that variance reduction of cases_w1 through w6 as a function of cases_w0 decreases from 15.31 for week one to 0.7276 for week 6, a decrease by a factor of more than 20. At the same time the variance reduction as a function of Border Distance is 0.228 at week 1 but increases to 1.981 at week 6 to become more significant than cases_w0. While the other environmental variables have significantly lower influence than Border Distance and always less than cases_w0, their influence also increases by roughly a factor of 20 from week 1 to week 6 prediction. Sensitivity analysis for classifiers like the binary Bayes is done in terms of entropy reduction. We again see a similar pattern. The entropy reduction as a function of Outbreak_w0 decreases from 0.01282 for week 1 predictions to 0.0001 for week 6, while the influence of Border Distance increases from 0.01625 for week 1 to 0.05631 for week 6. In the case of this model, all the environmental variables have a stronger influence on the prediction than the Outbreak_w0 variable by 4 week prediction horizon already.

5 Conclusion

This paper has explored the use of Bayes nets to predict malaria outbreaks. Following a data-driven methodology, we compared Bayes nets for numeric prediction with those specifically designed for outbreak prediction and compared both with ARIMA and ARIMAX models. Our results based on ROC and PR analysis show that the Bayes nets outperform the ARIMA model for all examined

prediction horizons except the shortest (1 week). In addition, the binary Bayes net does particularly well for the higher outbreak threshold, outperforming the other models for 2–6 week predictions.

Studies show a nonlinear relation between malaria cases and LST [14]. As a linear model ARIMAX could not capture the dynamics of this relation, hence the prediction accuracy was not high. In addition to LST, the Bayes net models were also able to take slope, distance to border, distance to stream and stream density into account. ARIMAX does not support such non-temporal variables.

Depending on the availability, inclusion of some other variables like human mobility, humidity, wind speed, wind direction, and even social media data might help improve the prediction accuracy.

Spatial auto-correlation is one of the characteristics of malaria transmission which is not addressed here. It is a measure of correlation between spatial features and outcome. By incorporating spatial auto-correlation (if there is any) the accuracy of the Bayes net might be improved.

Acknowledgments. This research project was supported by Faculty of Information and Communication Technology, Mahidol University. This paper is based upon work supported by the US Army International Technology Center Pacific (ITC-PAC) under contract FA5209-15-P-0183.

References

1. Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control. Holden-Day, Francisco (1970)
2. Buczak, A.L., Baugher, B., Guven, E., Ramac-Thomas, L.C., Elbert, Y., Babin, S.M., Lewis, S.H.: Fuzzy association rule mining and classification for the prediction of malaria in South Korea. BMC Med. Inform. Decis. Mak. **15**(1), 47 (2015)
3. Chatfield, C.: The Analysis of Time Series: An Introduction. Chapman & Hall, London (2004)
4. Cooper, G.F., Dash, D.H., Levander J.D., Wong, W., Hogan, W.R., Wagner, M.M.: Bayesian biosurveillance of disease outbreaks. In: 20th International Conference on Uncertainty in Artificial Intelligence, pp. 94–103. AUAI Press, Arlington (2004)
5. Gomez-Elipe, A., Otero, A., Herp, M.V., Aguirre-Jaime, A.: Forecasting malaria incidence based on monthly case reports and environmental factors in Karuzi Burundi, 1997–2003. Malar. J. **6**(1) (2007). Article no. 129
6. Haghdoost, A., Alexander, N., Cox, J.: Modelling of malaria temporal variations in Iran. Trop. Med. Int. Health **13**(12), 1501–1508 (2008)
7. Haque, U., Hashizume, M., Glass, G.E., Dewan, A.M., Overgaard, H.J., Yamamoto, T.: The role of climate variability in the spread of malaria in Bangladeshi highlands. PloS ONE **5**(12), e14341 (2010)
8. Hasan, A.H.M., Haddawy, P.: Integrating ARIMA and spatiotemporal Bayesian networks for high resolution malaria prediction. In: ECAI 2016, pp. 1783–1790. IOS Press (2016)
9. Hipel, K.W., McLeod, A.I.: Time Series Modelling of Water Resources and Environmental Systems. Elsevier, Amsterdam (1992)
10. Jiang, X., Wallstrom, G.L.: A Bayesian network for outbreak detection and prediction. In: 21st AAAI Conference, pp. 1155–1160. AAAI Press (2006)

11. Khamsiriwatchara, A., Sudathip, P., Vijakadge, S.S., Potithavoranan, T., Sangvichean, A., Satimai, W., Delacollette, C., Singhasivanon, P., Lawpoolsri, S., Kaewkungwal, J.: Artemisinin resistance containment project in Thailand. (I): implementation of electronic-based malaria information system for early case detection and individual case management in provinces along the Thai-Cambodian border. Malar. J. **11**, 247 (2012)
12. Koenraadt, C.J.M., Harrington, L.C.: Flushing effect of rain on container-inhabiting mosquitoes aedes aegypti and culex pipiens (Diptera: Culicidae). J. Med. Entomol. **45**(1), 28–35 (2008)
13. Malaria Report, Ministry of Public Health, Thailand. http://www.thaivbd.org
14. Mordecai, E.A., Paaijmans, K.P., Johnson, L.R., Balzer, C., BenHorin, T., Moor, E.: Optimal temperature for malaria transmission is dramatically lower than previously predicted. Ecol. Lett. **16**(1), 22–30 (2013)
15. Mubangizi, M., Ikae, C., Spiliopoulou, A., Quinn, J.A.: Coupling spatiotemporal disease modeling with diagnosis. In: 26th AAAI Conference, pp. 342–348. AAAI Press (2012)
16. 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**(3), e0118432 (2015)
17. Sebastiani, P., Mandl, K.D., Szolovits, P., Kohane, I., Ramoni, M.: A Bayesian dynamic model for influenza surveillance. J. Am. Stat. Assoc. **25**(11), 1803–1825 (2006)
18. Teklehaimanot, H.D., Schwartz, J., Teklehaimanot, A., Lipsitch, M.: Weather-based prediction of plasmodium falciparum malaria in epidemic-prone regions of Ethiopia II. Weather-based prediction systems perform comparably to early detection systems in identifying times for interventions. Malar. J. **3**, 44 (2004)
19. World Malaria Report 2015, WHO. http://apps.who.int/iris/bitstream/10665/200018/1/9789241565158_eng.pdf?ua=1
20. Zinszer, K., Kigozi, R., Charland, K., Dorsey, G., Brewer, T.F., Brownstein, J.S, Kamya, M.R, Buckeridge, D.L.: Forecasting malaria in a highly endemic country using environmental and clinical predictors. Malar. J. **14**, 245 (2015)
21. Zinszer, K., Verma, A.D., Charland, K., Brewer, T.F., Brownstein, J.S., Sun, Z., Buckeridge, D.L.: A scoping review of malaria forecasting: past works and future directions. BMJ Open **2**(6), e001992 (2012)

A Multiple-stage Classification of Fall Motions Using Kinect Camera

Orasa Patsadu[1(✉)], Bunthit Watanapa[2], and Chakarida Nukoolkit[2]

[1] Faculty of Science and Technology,
Rajamangala University of Technology Krungthep, Bangkok, Thailand
Orasa.p@mail.rmutk.ac.th
[2] School of Information Technology,
King Mongkut's University of Technology Thonburi, Bangkok, Thailand
{bunthit, chakarida}@sit.kmutt.ac.th

Abstract. This paper proposes a model of fall detection using hybrid classification methods in video streaming. In particular, we are interested in a stream of data representing time sequential frames of fifteen body joint positions capturable by Kinect camera. A set of features is then extracted and fed into the designated multiple-stage classification. The first stage classifies a fall as a different event from normal activities of daily living (ADLs). The second stage is to classify types of fall once the fall was detected in the first stage, for aiding the diagnosis and treatment of a fall by a physician. We selected a number of reliable machine learning algorithms (MLP, SVM, and decision tree) in forming a hybrid model. Experimental results show that the first stage classifier can differentiate falls and ADLs with 99.98% accuracy and the second stage classifier can identify type of fall with 99.35% accuracy.

Keywords: Fall detection · Hybrid classification methods · Kinect camera · Multiple-stage classifier · Smart home system

1 Introduction

Falls are a major problem especially for the elderly people. Statistics shows that approximately 33% of the elderly world-wide experience fell a year [1]. Falls may occur from several causes such as congenital diseases, environment, or ageing-related issues [2, 3]. Falls are a leading cause of injury, disability, and accidental death [4]. Our previous work [5] proposed a fall classification system to classify a fall as a different event from non-fall (a single-stage classifier model). However, that work has a limitation because there is no incident information for doctors to make fall diagnosis.

In this paper, we propose a two-stage classification where the first-stage classifier detects a fall. Later, the second-stage classifier identifies type of the fall detected by the first stage. Type of fall is important since it gives detail of fall incident (i.e. fall direction, velocity on impact, kinetic energy on impact, and sequence of body joint fall) for supporting decision of doctor in diagnosis. The input data are a set of features extracted from a time sequential frames of fifteen body joint positions (see in Fig. 2(b)) obtained from Kinect camera. This study uses Kinect camera, because it is affordable,

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_11

having low-level of image processing step, preserving human subject privacy, and highly practical to set up. The classification methods selected for our work were based on previous research literature [5]. There are multilayer perceptron (MLP), support vector machine (SVM), and decision tree (DT).

This paper is organized as follows: Sect. 2 presents related works; Sect. 3 describes the methodology of our proposed system; Sect. 4 presents the experimental results and discussions; finally, Sect. 5 presents the conclusion and future work directions.

2 Literature Review

Fall detection system has been developed using various approaches, for examples, acoustic and ambient sensor-based, kinematic sensor-based, and computer vision and NUI sensor-based [6]. Existing approaches share common limitations, i.e. complexity, intrusion, lack of privacy, or are expensive to be practically deployed for home use. Therefore, further research used Kinect camera to solve above problems is promising. Vitoantonio et al. [7] proposed a real-time system for fall detection based on RGB. This work uses contraction and the expansion speed of the width, height and depth of the 3D human bounding box to detect a fall. The result shows that this method can reduce false positive in fall detection.

In addition, Ma et al. [8] presented fall detection via shape features. The extreme learning machine is used to classify a fall using curvature scale space. The result shows that this method can classify with 86.83% accuracy, 91.15% sensitivity, and 77.14% specificity, respectively.

Wagner et al. [9] presented a method for fall detection. Decision Trees and SVM are used to classify falls using three-dimensional position, velocity and acceleration obtained from depth image. Likewise, Yang et al. [10] used thresholds obtained from 3D depth images. This method can detect a fall with high accuracy.

Existing hybrid classification methods used to detect fall can be called by several names such as hybrid data mining model and hybrid classification model [11]. These approaches are the integration of supervised classification or combination of both supervised learning and unsupervised learning. Dash et al. [11] presented a hybrid feature selection scheme and evaluated performance of four classifiers (i.e. radial basis function network, MLP, SVM using polynomial kernel, and SVM using RBF kernel function). The result shows that SVM (polynomial kernel) and MLP achieved higher accuracy than other classifiers.

Özdemir and Barshan [12] developed an automated fall detection system with wearable motion sensor based on three tri-axial devices (accelerometer, gyroscope, and magnetometer). There are six machine learning techniques: the k-nearest neighbor, least squares method, SVM, Bayesian decision making, dynamic time warping, and ANN. The result shows that the k-nearest neighbor and least squares method share the best result with sensitivity, specificity, and accuracy all above 99%. Likewise, Sukreep et al. [13] used decision tree, naïve Bayes, SVM, and k-nearest neighbor to detect a fall from the daily activities and in-house locations using a Smartphone. This method can detect a fall and in-house location with accuracy of 97.48% and 94.11%, respectively.

3 Methodology

In this section, we describe the model to detect and classify a fall of a considering subject. The system is divided into four sections: data collection, data preprocessing, real-time segmentation, and fall detection/classification as shown in Fig. 1.

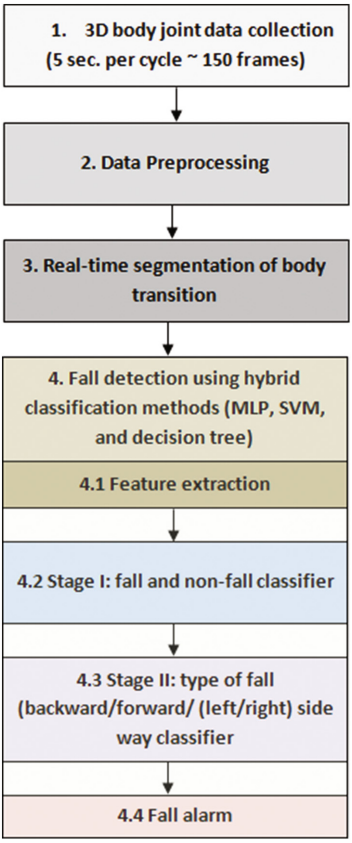
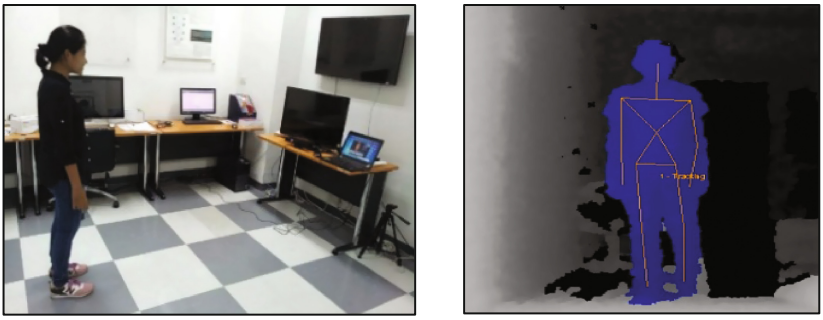


Fig. 1. Overview of the proposed system

According to Fig. 1, the holistic function of the proposed system is to continuously acquire and transform body-joint data of a subject from Kinect for performing fall detection (stage I). Once a fall is detected, a set of classifications work in ensemble style to identify type of the fall (stage II). The details of each component shown in Fig. 1 can be seen in various part of Sect. 3 as follows.

3.1 Experiment Setup and Dataset

We establish an indoor environment setting with a single Kinect camera [14] to track movement of the subject. The Kinect camera was set up at approximately 1 m above the floor as shown in Fig. 2(a). Kinect captures a video stream which has 640 * 480 resolution at the rate of 30 frames per second (FPS). We use OpenNI [15] to extract the vectors (X, Y, Z) of fifteen body joint positions as shown in Fig. 2(b).



(a.) Environment setup of Kinect camera (b.) Skeleton (body joint position)

Fig. 2. Experiment setup for fall detection system

In the experiment, dataset was collected, with approval of the Institutional Review Board of King Mongkut's University of Technology Thonburi. There are ten subjects (age 30 ± 8 years, body mass 75 ± 35 kg, and height 165 ± 15 cm with an equal number of males and females of various weights and heights). We simulated various falls and activities of daily living (ADLs) from the positions of standing, walking and sitting on a variety of seat types such as sofas, chairs with a backrest, and stools as shown in Table 1. Each scenario was repeated 15 times per subject, there are a total of 4,350 (10 × 29 × 15) video clips. To control the bias in model evaluation, our model was evaluated using fivefold cross-validation, where the whole dataset was spilt into a

Table 1. Various activities for motion detection

Type of activities	Situations → direction
Falls (11 types)	Sit on chair with backrest/stool/sofa → backward/forward/left/right fall
	Walking → forward fall
	Stand → backward/forward/left/right fall
ADLs (18 types)	Sit on chair with backrest/stool/sofa and lie on the floor/bend down/stand up → forward movement/left side movement
	Lie on the sofa and sit down on sofa/stool/chair with backrest/stand up → left side movement
	Stand on the floor and lie down on the floor/sit down on the sofa/stool/chair with backrest/bend down → left side movement/forward movement

training dataset for 1,320 falls and 2,160 ADLs from eight subjects, while data of the remaining subjects was used to test our model.

3.2 Data Preprocessing

The proposed classification method requires a certain preprocessing. A clip (5 s. per cycle ~150 frames) of a time sequential vector (X, Y, Z) data of selected body joints extracted from Kinect as explained in 3.1 will be normalized. Then, Euclidean distance of each two consecutive frames is calculated for being input in segmenting the body transition. Detail is presented next.

3.2.1 Preliminary Study

To optimize accuracy and processing time of the fall detection, we evaluated three alternatives: (1) all body joint positions, (2) head, shoulder, torso, hip and knee position, and (3) torso only position. The result shows that only the torso position provides high accuracy and fast run time when compared to the other alternatives. So, this work uses only the torso position.

3.2.2 Data Normalization

To adjust varied body sizes of subject, we use min-max normalization [16]. A series of torso position data are transformed data into normalized values in the range of [0, 1], where 0 represents minimum normalized value of that dimension in the clip and 1 represents the maximum.

3.2.3 Data Transformation

Once we performed min-max normalization, we map the normalized torso position into a time series of Euclidean distance [17] between two consecutive Kinect video frames (vector of torso position at time t and time t + 1, respectively).

3.3 Real-Time Segmentation of Body Transition [5]

Figure 3 shows example time series of Euclidean distance of transition phase between ADLs and falls. We can clearly see that generally the duration of transition from falls are half span of ADLs. Therefore, we use the transition phase to extract features for fall motion detection using our algorithm as shown in Fig. 4.

Based on the obtained boundaries, we can derive three features ($f_{max\ peak}$, $f_{no\ of\ frame}$, $f_{Avg.\ time}$) for being inputs of fall motion detection (feature extraction or step 4.1 in Fig. 1) as shown in Fig. 5.

$f_{max\ peak}$ denotes maximum peak of a time series of transition phase, denoted as P in Fig. 5, as depicted in Eq. 1.

$$f_{max\,peak} = \max;\ \forall \in [t_s, \ldots, t_f] \tag{1}$$

Where t is the number of frame within the transition phase.

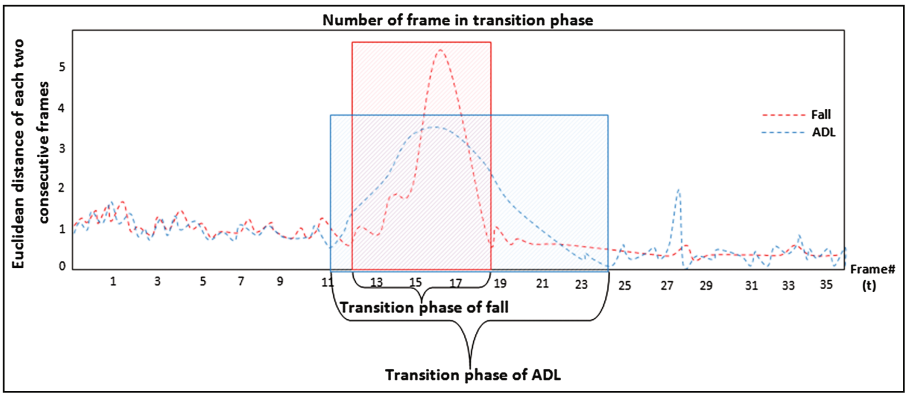


Fig. 3. An example of transition phase (ADLs and falls)

Algorithm of Real-time segmentation of body transition
Input: a time series of Euclidean distance between two consecutive Kinect video frames of torso position
Output: the boundaries of the transition phase, denoted as $[t_s, \ldots, t_f]$. t_s states the possible start frame of transition and t_f represents the finish frame of transition as shown in Figure 5.
1: Sliding window of time series data of the torso's position (5 seconds = 150 frames)
2: finding max peak of time series data of the torso's position
3: While loop (left side of max peak) to detect start frame of transition (the time series data of the torso's position ≤ a predefined threshold (t_l) (0.3))
4: While loop (right side of max peak) to detect finish frame of transition (time series data of the torso's position ≤ a predefined threshold (t_r) (0.3))

Fig. 4. Real-time segmentation of body transition algorithm

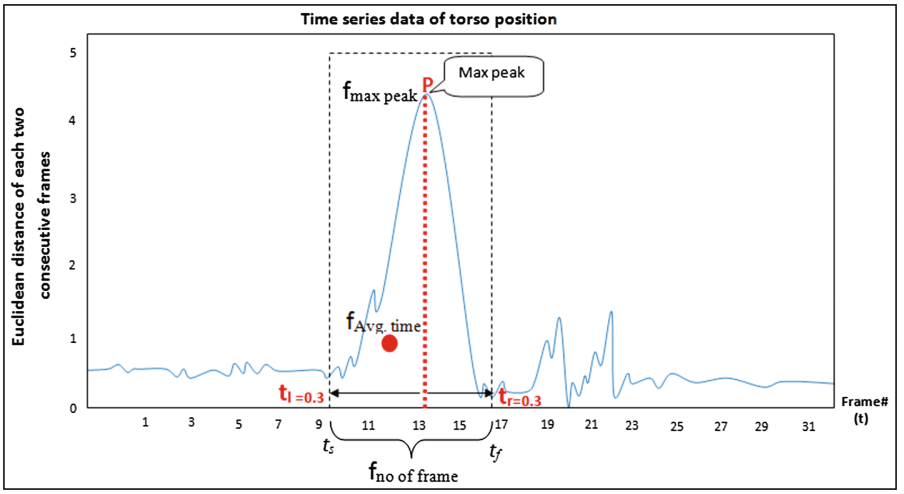


Fig. 5. Three extracted features based on the boundaries of the transition phase

$f_{no\ of\ frame}$ denotes the duration of the transition phase, which is distance from the start frame of transition (t_s) to the finish frame of transition (t_f). We can count number of frames within the boundary of the transition phase as depicted in Eq. 2.

$$f_{no\ of\ frame} =; \forall \in [\mathrm{t_s}, \ldots, \mathrm{t_f}] \tag{2}$$

Lastly, $f_{Avg.\ of\ time}$ denotes the average of start frame of transition phase to max peak as depicted in Eq. 3.

$$f_{Avg\ of\ time} = \text{average}\,(\text{start frame of transition} - \text{max peak}) \tag{3}$$

3.4 Fall Detection Using Hybrid Classification Methods

Once we obtained the set of extracted features, we use this feature set to build model for fall motion detection/classification based on Multiple-stage classifier using hybrid classification methods. We evaluated three well known classifiers (MLP, SVM, and decision tree) in performing fall detection using majority vote for each stage of determination as shown in Fig. 6.

MLP is a feedforward artificial neural network model that maps a set of input data onto a set of appropriate outputs. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. MLP utilizes a supervised learning technique called backpropagation for training the network. We apply MLP method to inductively construct model of decision in the first-stage classifier. There are three layers (input layer, hidden layer, and output layer) with 3, 3, and 2 nodes, respectively. We set the learning rate at 0.3 and momentum at 0.2. The input layer consists of three extracted features ($f_{max\ peak}$, $f_{no\ of\ frame}$, $f_{Avg.\ of\ time}$). The hidden layer is with three nodes. The output node suggests two possible values: fall or non-fall. If the output value is classified as fall, the prototype system also sends fall alarm signal to all related stakeholders (as shown as step 4.4 in Fig. 1). Moreover, the second stage classifier are set input layer, hidden layer, and output layer with 3, 4, and 4 nodes. The input numbers are the same as the first stage. The hidden layer has four nodes. The output node has four values: forward fall, backward fall, left side fall, and right side fall.

SVM is a newer approach that is able to classify linear or nonlinear data. The basic discipline of SVM is to compute a hyperplane, defined by support vectors, between each class and the rest in a way that the margin between two classes is maximized. Our study has been tested with various kernel functions (linear, polynomial, and a radial basis function (RBF)). Among these, RBF yielded the best performance and was used in our experiment. In this study, we used libSVM algorithm in WEKA for deploying SVM. Similar to the experiment on MLP, an empirical experiment was conducted for parameter tunings and evaluation scheme setting.

Finally, we also used decision tree for predicting multiple-stage classifier, yielding the output as a flowchart-like tree structure. In the experiment, we used J48 algorithm in WEKA. Likewise, we are set parameter and evaluation the same as MLP.

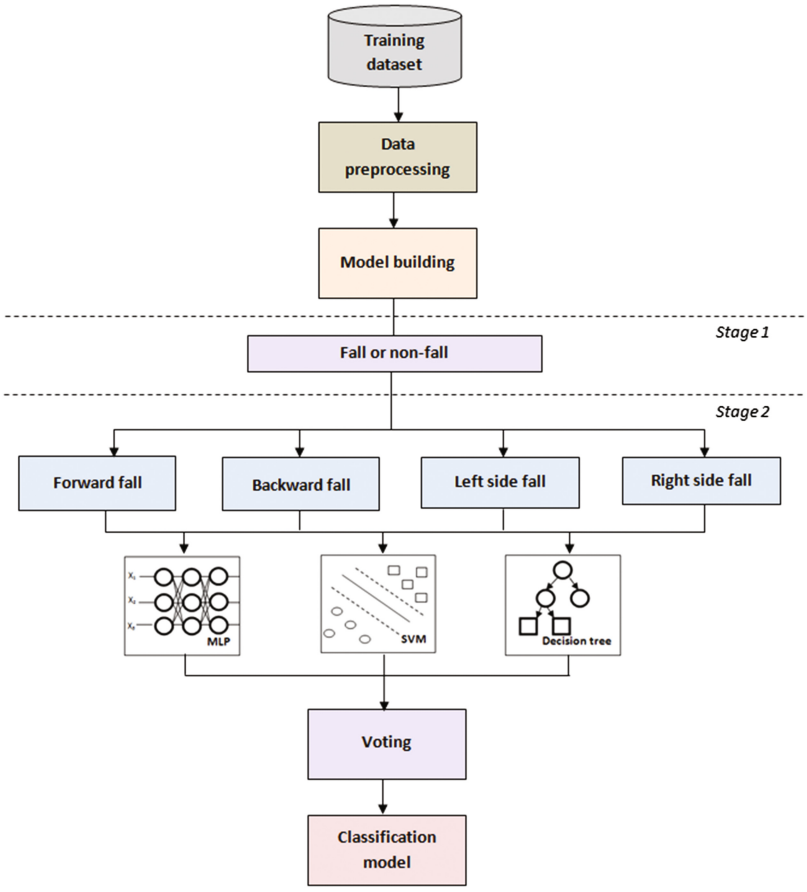


Fig. 6. Fall detection classification using hybrid model

4 Experiment and Result

In the experiment, the results of evaluating 870 video clips (332 falls (83 video clips/type of fall) and 538 non-falls) using MLP, SVM, and decision tree are depicted in a set of confusion matrix for fall detection based on Multiple-stage classifier as shown in Tables 2 and 3.

From Table 2, the result shows that SVM is the most accurate classification method, with its accuracy of 99.97%. Therefore SVM is a strong candidate for the classifier to be used for fall detection. Moreover, the result shows that MLP classifier achieves 99.77% accuracy. This method is also an effective and proper candidate for fall detection despite being slightly less accurate than SVM classifier. Finally, the result shows that decision tree can classify with 99.33% accuracy. However, this method has small error occurred in the case of the ambiguity between fall and lie down position on the floor.

Table 2. Confusion matrix of three classifiers for fall detection in the first-stage classifiers

Actual	Prediction					
	MLP		SVM		Decision tree	
	Fall	Non-fall	Fall	Non-fall	Fall	Non-fall
Fall	325	7	331	1	323	9
Non-fall	0	538	0	538	2	536

Table 3. Confusion matrix of three classifiers for fall type identification in the second-stage classifiers

Actual	Prediction			
	Forward fall	Backward fall	Left side fall	Right side fall
MLP classifiers				
Forward fall	80	0	2	1
Backward fall	0	80	3	0
Left side fall	1	0	81	1
Right side fall	2	0	1	80
SVM classifiers				
Forward fall	81	0	1	1
Backward fall	0	82	0	1
Left side fall	2	0	80	1
Right side fall	1	0	1	81
Decision tree classifiers				
Forward fall	78	0	3	2
Backward fall	0	79	2	2
Left side fall	2	0	80	1
Right side fall	3	0	3	77

From Table 3, the result shows that SVM is also the most accurate classification method, with its accuracy of 99.22%. The result of MLP achieves 99.01% accuracy. Decision tree can classify with 98.64% accuracy. This method also has error occurred in the case of the ambiguity between forward fall and (left/right) side fall.

From the experiment, the performance of each classification method is compared as seen in Tables 2 and 3. We found that each method has different accuracy dependent on the principles of each method. Both SVM and MLP are very good in detecting fall motions. SVM is better off, anyway. It has 0 FN which means all the simulated fall events in our experiment can be detected perfectly.

We are certain that the second-stage classifier is important and valuable for use in the future applications which is beneficial to doctors for support decision, in giving diagnosis and treatment for the subject, especially, in cases where the subject frequently falls. Doctors can use this information to trace abnormity of subject and examine cause of fall, because different type of fall can indicate unusual of subject such as vision issue, gait and balance problem, congenital diseases, or environment.

Table 4. The performance of individual classifier and hybrid classifiers for fall classification in multiple-stage classifier

	MLP		SVM		Decision tree	
	The 1st stage classifier	The 2nd stage classifier	The 1st stage classifier	The 2nd stage classifier	The 1st stage classifier	The 2nd stage classifier
Accuracy	99.77%	99.01%	99.97%[a]	99.22%	99.33%	98.64%
Recall	0.9974	0.9900	1.0000[a]	0.9920	0.9930	0.9860
Precision	0.9988	0.9860	0.9996[a]	0.9900	0.9730	0.9550

	Hybrid method (MLP + SVM + Decision tree)	
	The 1st stage classifier	The 2nd stage classifier
Accuracy	99.98%[a]	99.35%
Recall	1.0000[a]	0.9930
Precision	0.9998[a]	0.9930

[a]Perfect fall detection

In the next step of our study, we plan to use result of multiple-stage classifier to aid and support decision of doctor in the diagnosis for further treatment.

In addition, we proposed the performance of an individual classifier and hybrid classifiers. The result is shown in Table 4.

From Table 4, the data set described in Sect. 3.1 is being used to test the performance of an individual classifier and hybrid classifiers. Classification accuracy was evaluated using fivefold cross-validation. In the proposed approach, first the base MLP, SVM, and decision tree of multiple-stage classifier are constructed individually, which obtain a very satisfactory performance. Secondly, the ensemble of MLP, SVM, and decision tree of multiple-stage classifier is designed. In the hybrid approach, fall detection of the first-stage classifier and the fall classification of the second-stage classifier achieve with accuracy 99.98% and 99.35%, respectively. It has 0 FN which means all the simulated falls can be detected perfectly compared to an individual classifier. From our experimental result, the hybrid approach outperforms an individual classifier because it has multiple learners, which provide a set of alternative options. For the fall detection process, the hybrid approach automatically evaluates the result of an individual classifier to select highly accurate classifier. Thus, fall detection using a hybrid approach improves the accuracy of the detection system when compared to those using individual approach, especially in the cases of the ambiguity between fall and lie down position on the floor and the ambiguity between forward fall and (left/right) side fall.

5 Conclusion and Future Work

In this paper, we propose multiple-stage fall motion detection using Kinect camera. The system used a time sequential frames of fifteen body joint positions to detect a fall and then identify type of the fall based on multiple-stage classier using hybrid classification

methods. Data mining classification methods, which include MLP, SVM, and decision tree, are investigated for fall motion detection. In the experiment, the results indicate that SVM shows superior performance compared to other classification methods in both stage of classifier, and can detect fall with 99.97% accuracy. In addition, the hybrid approach for fall detection of the first-stage classifier achieves 99.98% accuracy and 99.35% accuracy for the second-stage classifier, which confirms the effectiveness of using hybrid approach in fall detection applications. In the near future, we plan to expand the heuristic knowledge of fall detection and classification (i.e. fall direction, velocity on impact, kinetic energy on impact, and sequence of body joint fall) for supporting decision of doctor in diagnosing the incurred fall for further treatment.

Acknowledgements. This work was supported by Rajamangala University of Technology Krungthep. We thank students and staffs of the SIT, King Mongkut's University of Technology Thonburi for their invaluable assistance in setting up the experimental environment for the capturing sessions.

References

1. Nevitt, M.C., Cummings, S.R., Kidd, S., Black, D.: Risk factors for recurrent nonsyncopal falls a prospective study. J. Am. Med. Assoc. **261**(18), 2663–2668 (1989)
2. Sorysang, L., Kkhompraya, J., Natetanasombut, K.: A study of fall prevention guideline in older adult living in Mitraphappatana community. J. R. Thai Army Nurses **15**(1), 122–129 (2014)
3. Webster, D., Celik, O.: Systematic review of Kinect applications in elderly care and stroke rehabilitation. J. Neuroeng. Rehabil. **11**(108), 1–24 (2014)
4. Understanding Dementia. http://www.helpguide.org/articles/alzheimers-dementia/understanding-dementia.htm
5. Patsadu, O.: Video mining for fall motion detection using kinect and hybrid classification methods. Dissertation, King Mongkut's University of Technology Thonburi (2016)
6. Patsadu, O., Nukoolkit, C., Watanapa, B.: Survey of smart technologies for fall motion detection: techniques, algorithms and tools. In: 5th International Conference on Advances in Information Technology, Thailand, pp. 137–147 (2012)
7. Vitoantonio, B., Nicola, N., Donato, B., Michele, P., Marco, S., Dario, D.A., Alessio, V., Claudio, L., Fabio, S.: Fall detection in indoor environment with kinect sensor. In: IEEE International Symposium on Innovations in Intelligent Systems and Applications, Romania, pp. 1–6 (2014)
8. Ma, X., Wang, H., Xue, B., Zhou, M., Ji, B., Li, Y.: Depth-based human fall detection via shape features and improved extreme learning machine. J. Biomed. Health Inform. **18**(6), 1915–1922 (2014)
9. Wagner, J., Morawski, R.Z.: Applicability of Mel-cepstrum in a fall detection system based on infrared depth sensors. In: 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Poland, pp. 1–6 (2015)
10. Yang, L., Ren, Y., Zhang, W.: 3D depth image analysis for indoor fall detection of elderly people. J. Digit. Commun. Netw. **2**(1), 24–34 (2016)
11. Dash, S., Patra, B., Tripathy, B.K.: A hybrid data mining technique for improving the classification accuracy of microarray data set. J. Inf. Eng. Electron. Bus. **2**, 43–50 (2012)

12. Özdemir, A.T., Barshan, B.: Detecting falls with wearable sensors using machine learning techniques. J. Sens. **14**(6), 10691–10708 (2014)
13. Sukreep, S., Mongkolnam, P., Nukoolkit, C.: Detect the daily activities and in-house locations using smartphone. In: 11th International Conference on Computing and Information Technology, Thailand, pp. 215–225 (2015)
14. Kinect for Xbox 360. http://www.support.xbox.com/en-US/xbox-360/accessories/kinect-sensor-setup
15. OpenNI. https://www.github.com/OpenNI/OpenNI
16. Jain, Y., Bhandare, S.: Min max normalization based data perturbation method for privacy protection. Int. J. Comput. Commun. Technol. **2**(8), 45–50 (2011)
17. Han, J., Kamber, M.: Data Mining Concepts and Techniques, 3rd edn. Morgan Kaufmann Publishers, Burlington (2011)

Recognizing Quality of Floor Tiling from Knocking Signals Using HMMs

Rong Phoophuangpairoj(✉)

Department of Computer Engineering, College of Engineering,
Rangsit University, Bangkok, Thailand
rong.p@rsu.ac.th

Abstract. Property buyers and homeowners throughout the world have discovered badly tiled floors in their buildings. Consequently, they spend time on expensive repairs because their tiling has a shorter than expected lifetime. If it were possible to ascertain the quality of floor tiling before making a payment, this problem could be solved. Usually, it is difficult to determine the quality of floor tiling visually. Therefore, this work proposes a non-destructive method of identifying correctly and incorrectly laid flooring using tile knocking signals. Acoustic models were created using Mel Frequency Cepstral Coefficients (MFCCs) and Hidden Markov Models (HMMs). The sounds resulting from firmly or gently knocking on tiles, and the characteristics of knocking signals were used for the acoustic model training and testing. Regardless of firm or gentle knocking, the results showed that the proposed method could distinguish between correctly and incorrectly laid floor tiles accurately and efficiently.

Keywords: Tiling quality · Knocking signals · Signal processing · MFCC · HMM

1 Introduction

It is impossible to determine the quality of floor tiling by its external appearance or even by standing on it, as shown in Figs. 1 and 2, respectively. As a result, large sums of money have been wasted on substandard floor tiling. The life expectancy of a tiled floor depends on the quality of the flooring used, and whether or not the tiles are correctly laid. Figure 3 shows correctly and incorrectly laid tiling. For correctly laid tiling, cement is spread on the floor and tile thoroughly. However, some tilers ignore quality and try to finish their work quickly; they do not spread enough cement on the floor and tile. For incorrect tiling, tiles are improperly glued to the floor, which results in the floor requiring to be retiled before the end of its expected lifetime. Therefore, an inexpensive non-destructive method of recognizing signals generated by knocking on correctly and incorrectly laid tiles should be developed and used to determine tiled floor quality.

X-rays have been used to evaluate pineapple grades using translucency. However, for intermediate levels of translucency, the X-ray method is less accurate. Real-time digital imaging equipment such as the linescan X-ray machine would be needed to evaluate the quality of floor tiling. This machine is well suited for real-time sorting;

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_12

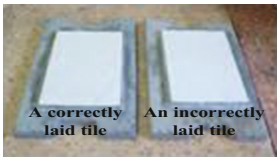


Fig. 1. The inability to distinguish between correctly and incorrectly laid tiles from their external appearance

Fig. 2. The inability to distinguish between correctly and incorrectly laid tiles by standing on them

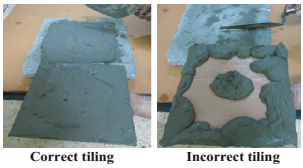


Fig. 3. Correct and incorrect tiling

however, it costs up to $100,000 per unit [1]. Accordingly, the X-ray method is expensive and unsuitable for testing the quality of floor tiling.

Speech recognition technologies have been studied for decades. MFCCs are acoustic features representing frequency components of signals, which have been widely used as speech features [2–7]. HMMs have been used to efficiently model speech phenomena and they can be found in many speech recognition systems [2–4]. A continuous HMM consisting of states and transition probabilities can efficiently capture speech phenomena at a particular time, and speech change according to time. Each HMM state comprises of Gaussian mixtures. The number of HMM states and the number of Gaussian mixtures can affect the recognition accuracy and time. Therefore, these aspects require adjustment to obtain optimal recognition performance.

Recent research has shown that the quality of fruits such as watermelons, durians, and pineapples could be determined using flicking a0nd striking signals [8–10]. Ultrasonic non-destructive methods have been used to characterize porosity and identify defects in ceramic materials [11]. An automatic grading system is essential for quality control. Several image-processing methods have been studied to detect tile defects [12–14]. However, a computerized method of classifying correctly and incorrectly laid tiles using knocking signals has yet to be studied. Hence, this work proposes a method of recognizing the quality of floor tiling using knocking signals.

2 Materials and Methods

The knocking sounds were collected by knocking a Thai 10 Baht coin on correctly and incorrectly laid tiles. The Thai 10 Baht coin is shown in Fig. 4.

Fig. 4. A Thai 10 Baht coin

Figure 5 shows the computer and microphone, which were used to collect the sounds resulting from knocking on the laid tiles with a coin.

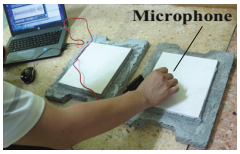


Fig. 5. Collecting tile knocking sounds

The following two stages were used to determine the quality of the laid tiles: (1) extracting the acoustic features from the tile knocking signals, and (2) recognizing the tile knocking acoustic features, as shown in Fig. 6.

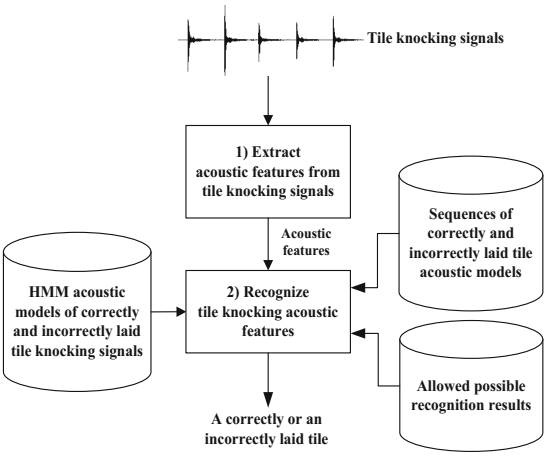


Fig. 6. Laid tile quality recognition using knocking signals

At the first stage of the process, acoustic features were extracted from tile knocking signals. The amplitude of the knocking signals could not be used to determine the quality of the tiles because they could have been knocked at different force levels. Therefore, MFCC-based spectral acoustic features were extracted from the knocking signals. At the second stage of the process, HMM acoustic models of correctly and incorrectly laid tile knocking signals, sequences of correctly and incorrectly laid tile acoustic models, and allowed possible recognition results were used to determine the extracted acoustic features to obtain a result, which indicated a correctly or an incorrectly laid tile. The proposed method is further explained as follows:

2.1 HMM Acoustic Models of Correctly and Incorrectly Laid Tile Knocking Signals

Four sources of training data consisting of (1) signals obtained from knocking on correctly laid floor tiles firmly, (2) signals obtained from knocking on incorrectly laid floor tiles firmly, (3) signals obtained from knocking on correctly laid floor tiles gently, and (4) signals obtained from knocking on incorrectly laid floor tiles gently were used to create the acoustic models. Based on the characteristics of tile knocking signals, for the acoustic model creation, the signals resulting from knocking on correctly laid tiles were transcribed as sil cI cM cF sil cI cM cF sil cI cM cF sil …. sil cI cM cF sil. Where sil represents a silent part of knocking signals and the cI, cM and cF represent initial, middle, final parts of correctly laid tile knocking signals, respectively. The signals resulting from knocking incorrectly laid tiles were transcribed as sil icI icM icF sil icI icM icF sil icI icM icF sil …. sil icI icM icF sil. Where icI, icM and icF represent the initial, middle, and final parts of incorrectly laid tile knocking signals. A repetition number of sil cI cM cF sil or sil icI icM icF sil was equivalent to the number of times a tile was knocked. First, to create the acoustic models, acoustic features were extracted from knocking signals. Next, the obtained acoustic features and their transcriptions were used to train seven HMM acoustic models, representing the initial, middle, and final parts of correctly laid tile knocking signals, the initial, middle, and final parts of incorrectly laid tile knocking signals and the silent parts of knocking signals. The silent parts of correctly and incorrectly laid tile knocking signals were used in the training a silent acoustic model. After the acoustic models were created, they were used to define sequences of correctly laid and incorrectly laid tile acoustic models.

2.2 Sequences of Correctly and Incorrectly Laid Tile Acoustic Models

Sequences of correctly and incorrectly laid tile acoustic models were defined as follows:

```
correctlylaid       [correctlylaid]    cI  cM  cF
incorrectlaid       [incorrectlaid]    icI  icM  icF
SENT-START []  sil
SENT-END []    sil
```

The first column shows the words used to represent the quality of tiling, followed by an optional output symbol in the second column, which is enclosed in square brackets, [and]. If an output symbol is not specified, the name of the word itself is output. Empty square brackets, [], were used to suppress any output when that word was recognized. The third column shows the sequence of phones (acoustic models) that were used to represent the tiling quality.

2.3 Allowed Possible Recognition Results

Before the recognition of tile knocking signals, the allowed possible recognition results were defined, as follows:

$$(\text{SENT-START} < \text{correctlylaid} \mid \text{incorrectlylaid} > \text{SENT-END})$$

The line tells the recognizer that only the repetition (<>) of "correctlylaid" or "incorrectlylaid" could be obtained as the final recognition result. Since this repetition was allowed, the system could recognize both one knock and a large number of knocks.

2.4 Extracting Acoustic Features from Tile Knocking Signals

MFCC-based acoustic features were extracted from the tile knocking signals. Initially, a pre-emphasis coefficient of 0.97 and the Hamming window were applied. Next, the Fast Fourier Transform (FFT) was used to compute the frequency spectra of the tile knocking signals. A pre-emphasis coefficient of 0.97 was used as the default value for the Hidden Markov Toolkit (HTK) [15]. Then, the log amplitudes of the spectra were mapped onto the Mel scale using a filter bank with 26 channels. After that, a discrete cosine transform (DCT) was applied to obtain 12 MFCCs, and then an energy feature was added. Subsequently, the 1^{st}- and 2^{nd}-order derivatives of the MFCCs and the energy were computed and added to the acoustic features. Finally, 39-dimension features, consisting of 12 MFCCs with energy and their 1^{st}- and 2^{nd}-order derivatives, were obtained and used as acoustic features to evaluate the tiled floor quality.

2.5 Recognizing Tile Knocking Acoustic Features

For the recognition of the tile knocking signals, HMM acoustic models were connected based on the sequences of acoustic models of correctly and incorrectly laid tile signals, and the allowed possible recognition results, to create possible recognition paths. From the extracted acoustic features, the most likely path that had the highest probability was found and used as the final result. To measure the performance of the proposed method, the percentage correctness and percentage accuracy values were calculated, using the following equations:

$$\%\, correctness = \frac{N - D - S}{N} \times 100\% \tag{1}$$

$$\%\, accuracy = \frac{N - D - S - I}{N} \times 100\% \tag{2}$$

N: the total number of tile knocks
D: the number of deletion errors
S: the number of substitution errors
I: the number of insertion errors

3 Experimental Results

Experiments were conducted to evaluate the proposed method. Tile knocking sounds were recorded using the 16-bit PCM format at 11,025 Hz. A frame size of 25 ms, with a frame shift interval of 15 ms, was used to extract the acoustic features. Knocking sounds used for training and testing were obtained from different tiles that had the same properties. Knocking sounds for training the acoustic models for correctly and incorrectly laid tile signals consisted of 20 firm and 20 gentle knocks obtained from a correctly laid tile, as well as 20 firm and 20 gentle knocks obtained from an incorrectly laid tile. To create acoustic models for correctly laid tile signals, firm and gentle knocks obtained from a correctly laid tile were used while firm and gentle knocks obtained from an incorrectly laid tile were used to create acoustic models for the incorrectly laid tile signals. The test set was collected by knocking on two tiles that were not used in the acoustic model creation. The test set composed of 400 knocks, consisting of 100 firm knocks and 100 gentle knocks obtained from knocking on a correctly laid tile, together with 100 firm knocks and 100 gentle knocks obtained from knocking an incorrectly laid tile. The HTK [16] was used to extract acoustic features from knocking signals, create the HMM acoustic models, and recognize the features. The results are divided into 3 parts: (1) tile knocking signals and spectral views, (2) laid tile quality recognition rates, and (3) laid tile quality recognition time.

3.1 Tile Knocking Signals and Spectral Views

Figures 7 and 8 show the signal and spectral views of a firmly knocked on correctly laid tile and a firmly knocked on incorrectly laid tile, respectively. The signals obtained from firmly knocking on a correctly laid tile often had lower amplitude than those obtained from firmly knocking on an incorrectly laid tile. The spectra derived from a correctly laid tile were different to those obtained from an incorrectly laid tile. Sounds resulting from knocking on incorrectly laid tiles usually had higher frequencies than those resulting from knocking on correctly laid tiles.

Figures 9 and 10 show signals and the spectral view obtained from gently knocking on a correctly laid tile and an incorrectly laid tile, respectively.

When gently knocking on an incorrectly laid tile, the waveform signals may not be different from those derived from firmly knocking on a correctly laid tile. Therefore, the

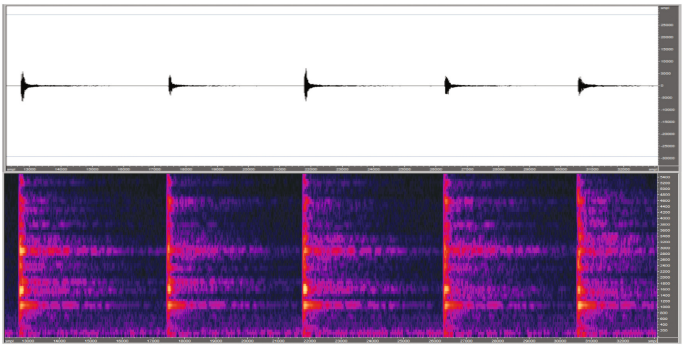


Fig. 7. Signals and the spectral view obtained from firmly knocking on a correctly laid tile with a coin

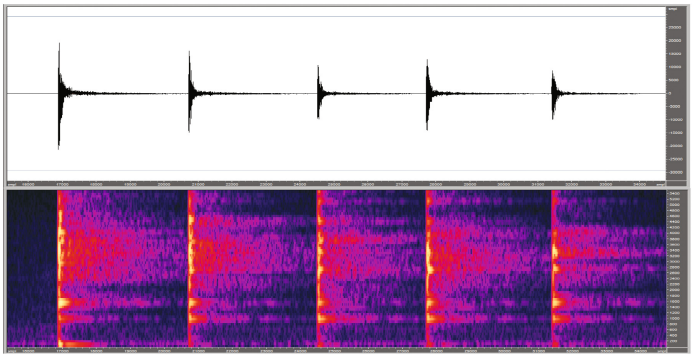


Fig. 8. Signals and the spectral view obtained from firmly knocking on an incorrectly laid tile with a coin

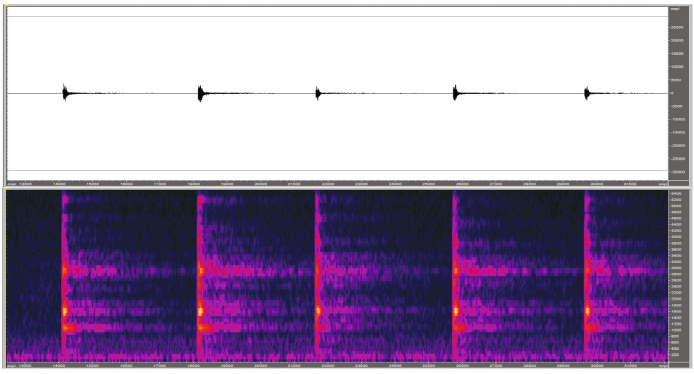


Fig. 9. Signals and the spectral view obtained from gently knocking on a correctly laid tile with a coin

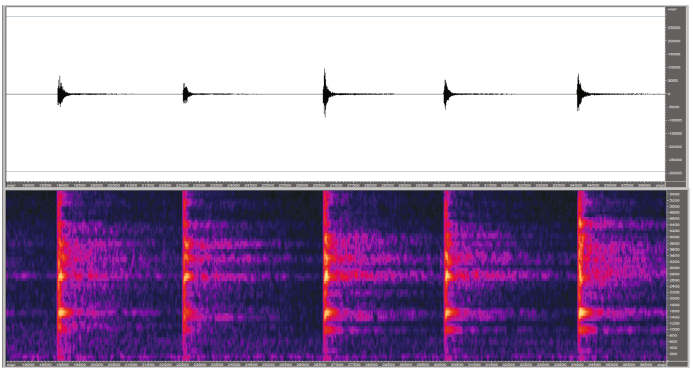


Fig. 10. Signals and the spectral view obtained from gently knocking on an incorrectly laid tile with a coin

amplitude of the knocking signals could not be used to differentiate correctly laid tiles from incorrectly laid ones. Classifying correctly laid and incorrectly laid tiles using the spectral information of their knocking signals was found to be more accurate than using the amplitude.

3.2 Laid Tile Quality Recognition Rates

Table 1 shows the laid tile quality recognition rates obtained from classifying knocking signals for the test set. The number of HMM states and Gaussian mixtures for each state were varied to obtain optimal recognition rates.

The results show that using 4 emitting states, with 2 to 5 Gaussian mixtures per state resulted in higher correctness and accuracy percentages than using 3 emitting states, with 2 to 5 Gaussian mixtures per state. When using 3 emitting states, with 5 Gaussian mixtures, correctness and accuracy percentages of 88.50 and 88.25 were obtained. An accuracy rate of 100 percent was achieved when using 4 emitting states, with 2 to 5 Gaussian mixtures per state.

Table 1. Laid tile quality recognition rates

Number of HMM emitting states	Number of Gaussian mixtures	% correctness	% accuracy
3	2	77.25	77.25
3	3	84.75	84.75
3	4	76.00	76.00
3	5	88.50	88.25
4	2	100	100
4	3	100	100
4	4	100	100
4	5	100	100

3.3 Laid Tile Quality Recognition Time

Table 2 shows the time taken to extract the acoustic features, recognize the features and the total time, which was measured from the test set. An Intel® Core (TM) i7-7402MQ 2.2 GHz computer notebook with 8 GB of memory and Microsoft Windows, were used to evaluate the method. The feature extraction time, measured from 400 knocking signals, was 306 ms. The time spent on extracting the acoustic features did not depend on the number of states and mixtures. When varying the number of HMM emitting states from 3 to 4, and the number of Gaussian mixtures in each state from 2 to 5, the time spent on recognizing the features varied from 173 to 269 ms, and the total time varied from 479 to 575 ms. When using 3 emitting states, with 2 to 5 Gaussian mixtures per state, the average time spent on extracting features and recognizing each knock was only 1.20, 1.27, 1.34 and 1.40 ms, respectively. When using 4 emitting states, with 2 to 5 Gaussian mixtures per state, the average total time spent on extracting features and recognizing each knock was 1.24, 1.32, 1.37 and 1.44 ms, respectively.

The results show that the proposed method could classify signals derived from knocking on correctly and incorrectly laid tiles accurately and rapidly. On average, the total time taken to recognize each knocking signal was less than 2 ms.

Table 2. Time taken to recognize correctly and incorrectly laid tiles

Number of HMM emitting states	Number of Gaussian mixtures in each state	Feature extraction time (ms)	Recognition time (ms)	Total time (ms)	Total time/one knock (ms)
3	2	306	173	479	1.20
3	3		203	509	1.27
3	4		230	536	1.34
3	5		254	560	1.40
4	2		188	494	1.24
4	3		220	526	1.32
4	4		240	546	1.37
4	5		269	575	1.44

4 Conclusions

This paper proposes an innovative computerized method of determining the quality of floor tiling. HMMs were used to model time-varied and continuous tile knocking signals. The results show that correctly and incorrectly laid tiles could be recognized using knocking signals. The knocking and flicking signals can be used to determine not only the quality of fruits such as watermelons, durians and pineapples [8–10] but also the quality of tiling. The highest recognition rate was achieved when using 4-state HMMs, with 2 to 5 Gaussian mixtures per state. This research has preliminarily investigated the possibility of using an inexpensive non-destructive method to classify

tiled floor quality. This study did not use a large amount of tile knocking data, more data collected from different types of flooring and wall tiles should be gathered and evaluated. In addition, other methods could be tested and compared to this method.

References

1. Haff, R.P., Slaughter, D.C., Sarig, Y., Kader, A.: X-ray assessment of translucency in pineapple. J. Food Process. Preserv. **30**, 527–533 (2006)
2. Tangruamsub, S., Punyabukkana, P., Suchato, A.: Thai speech keyword spotting using heterogeneous acoustic modeling. In: IEEE International Conference on Research, Innovation and Vision for the Future, pp. 253–260 (2007)
3. Tangwongsan, S., Phoophuangpairoj, R.: Boosting Thai syllable speech recognition using acoustic models combination. In: International Conference on Computer and Electrical Engineering, pp. 568–572 (2008)
4. Tangwongsan, S., Po-Aramsri, P., Phoophuangpairoj, R.: Highly efficient and effective techniques for Thai syllable speech recognition. In: Maher, M.J. (ed.) Advances in Computer Science – ASIAN 2004 Higher Level Decision Making. ASIAN2004. LNCS, vol. 3321, pp. 259–270. Springer, Heidelberg (2004)
5. Li, F., Ma, J., Huang, D.: MFCC and SVM based recognition of Chinese Vowels. In: Hao, Y., et al. (eds.) Computational Intelligence and Security. CIS 2005. LNCS, vol. 3802, pp. 812–819. Springer, Heidelberg (2005)
6. Phoophuangpairoj, R.: Using multiple HMM recognizers and the maximum method to improve voice-controlled robots. In: International Conference on Intelligent Signal Processing and Communication Systems, pp. 1–6 (2011)
7. Ting, H., Yingchun, Y., Zhaohui, W.: Combining MFCC and pitch to enhance the performance of the gender recognition. In: 8th International Conference on Signal Processing (2006)
8. Phoophuangpairoj, R.: Automated classification of watermelon quality using non-flicking reduction and HMM sequences derived from flicking sound characteristics. J. Inf. Sci. Eng. **30**(4), 1015–1033 (2014)
9. Phoophuangpairoj, R.: Durian ripeness striking sound recognition using N-gram models with N-best lists and majority voting. In: Boonkrong, S., Unger, H., Meesad, P. (eds.) Recent Advances in Information and Communication Technology. Advances in Intelligent Systems and Computing, vol. 265, pp. 167–176. Springer, Cham (2014)
10. Phoophuangpairoj, R., Srikun, N.: Computerized recognition of pineapple grades using physicochemical properties and flicking sounds. Int. J. Agric. Biol. Eng. **7**(3), 93–101 (2014)
11. Eren, E., Kurama, S., Solodov, I.: Characterization of porosity and defect imaging in ceramic tile using ultrasonic inspections. Ceram. Int. **38**, 2145–2151 (2012)
12. Hocenski, Z., Matić, T., Vidović, I.: Technology transfer of computer vision defect detection to ceramic tiles industry. In: International Conference on Smart Systems and Technologies (SST), pp. 301–305 (2016)
13. Mohan, V., Kumar, S.S.: An automated tiles defect detection. Int. J. Comput. Appl. **109**(11), 24–27 (2015)
14. Karimi, M.H., Asemani, D.: Surface defect detection in tiling industries using digital image processing methods: analysis and evaluation. ISA Trans. **53**, 834–844 (2014)
15. Young, S. et al.: The HTK Book (for HTK Version 3.4), December 2006
16. The Hidden Markov Model Toolkit (HTK). http://htk.eng.cam.ac.uk/

Knee Implant Orientation Estimation for X-Ray Images Using Multiscale Dual Filter and Linear Regression Model

Theerawee Kulkongkoon, Nagul Cooharojananone, and Rajalida Lipikorn(✉)

Machine Intelligence and Multimedia Information Technology Lab,
Faculty of Science, Department of Mathematics and Computer Science,
Chulalongkorn University, Bangkok 10330, Thailand
Theerawee.k@student.chula.ac.th, {Nagul.C,Rajalida.L}@chula.ac.th

Abstract. For every patient who had total knee replacement, the routine follow-up is needed for a post-operation observation. The orientations of tibia, femur and knee implants are of clinical interest to orthopedic surgeons. In a routine follow-up examination, an orthopedic surgeon has to measure the tilt angles or orientations of knee implants manually from an x-ray film which is not quite accurate and inconvenient. This paper proposes an algorithm to automatically estimate the orientations of knee implants from an x-ray image using multiscale dual filter and linear regression model. The algorithm can remove tissues and noise while preserving and enhancing the bones. The proposed algorithm was evaluated on a set of 91 frontal view X-ray images of knee with implants and the experimental results, which were verified by the orthopedic surgeons, reveal 92% acceptance rate.

Keywords: Total knee replacement · Multiscale dual filter · Linear regression · X-ray imaging · Edge detection

1 Introduction

Total knee replacement is an effective treatment for patients with osteoarthritis, rheumatoid arthritis, or other degenerative joint diseases by replacing patient's articular surfaces of the knee bones with implants that consists of two main components: tibial and femoral components. The problem with total knee replacement is that the implants wear out by long usage and all the components will gradually loosen. A patient needs to see his orthopedic surgeon for a routine follow-up examination which is usually once every six months or once a year. The orientations of tibial and femoral components are of clinical interest to an orthopedic surgeon. Using a traditional method, a surgeon has to measure the orientations of tibial and femoral components from an X-ray film which is not quite accurate. Hailey et al. [1] presented tribological study of retrieved knee explants while Mints et al. [2] evaluated and observed severe polyethylene

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_13

wear using arthroscopy. Sanzen et al. [3] used fluoroscopically guided radiograph technique to measure the femorotibial distance. Fukuoka et al. [4,5] proposed to use the 3D/2D matching algorithm [6] to estimate the 3-D pose orientations of knee implants and to measure the femorotibial distance. Recently, Banks and Hodge [7] estimated the 3D pose by matching the projected contour with a library of shapes over all possible orientations. The accuracy of this method [7] was further improved by utilizing larger image libraries [8]. The problems with this method are the large image libraries and the time it takes to compare between the observed implants and the images in the libraries.

This paper thus presents an alternative algorithm to automatically measure the orientations of tibial and femoral components from an X-ray image using the proposed multiscale dual filter and linear regression model. The paper is organized into 4 sections. Section 2 presents the multiscale dual filter whereas Sect. 3 proposes an algorithm to measure the orientations. Section 4 provides the experimental results and conclusions.

2 Proposed Multiscale Dual Filter

The proposed multiscale dual filter integrates Laplacian filter and multiscale circular averaging filter to alternatively sharpen the bones and blur the tissues. The multiscale dual filtering technique is as follows:

1. Sharpen an image using Laplacian filter of size 3×3 as defined in Eq. (1)

$$\nabla^2 = \frac{4}{1+\alpha} \begin{bmatrix} \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \\ \frac{1-\alpha}{4} & -1 & \frac{1-\alpha}{4} \\ \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \end{bmatrix} \tag{1}$$

   where α controls the weight of the neighbors and is in the range 0.0 to 1.0. The value of α is set to 0.2 in order to put more weight to the vertical and horizontal neighbors.
2. Blur an image obtained from the previous step using the circular averaging filter starting with radius as large as one-third the width of the leg in order to blend the tissues and the background.
3. Sharpen an image obtained from step 2 using Laplacian filter of size 3×3 from Eq. (1).
4. Reduce the radius of the circular averaging filter by a constant factor, c. If the radius is greater than 1 then go back to step 2 else stop.

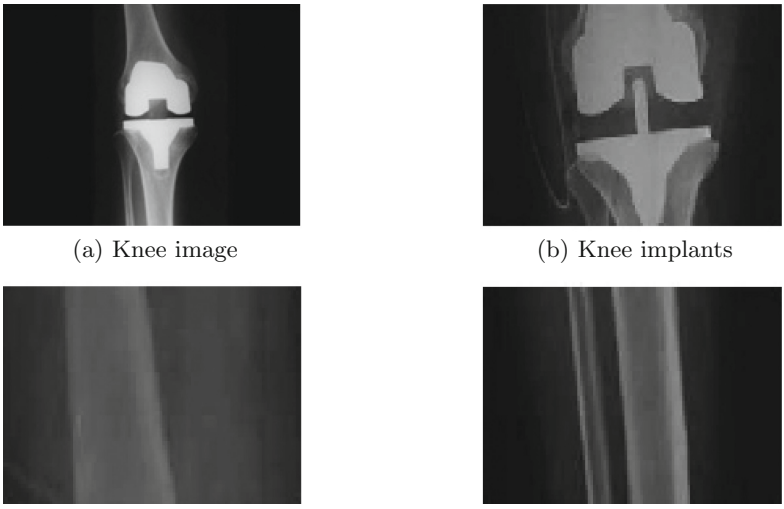
3 Orientation Estimation Algorithm

In order to estimate the orientation of knee implants, an X-ray image is converted to a gray scale image and then the orientations can be measured by performing component identification, contour finding, and orientation estimation as follows:

3.1 Component Identification

The first step of the orientation estimation process is to identify knee implants, femur, and tibia from an image as shown in Fig. 1

1. **Knee Implant Identification:** Two main components of knee implants, which are femoral and tibial components, can be identified from an image based on the fact that knee implants have higher intensities than any other components in an image by:
   (a) Setting intensity of the boundary between the leg and the background to the intensity, P_t, at the valley on the high end of a histogram(approximately at 85th percentile of the histogram of an image as shown in Fig. 2).
   (b) Selecting three intensities above the boundary value, P_t, with the highest frequencies, P_1, P_2, P_3, as shown in Fig. 2. These intensities represent the main intensities of knee implants.
   (c) Computing the average intensity of the three intensities selected from the previous step; i.e.,
   $$P_{avg} = (P_1 + P_2 + P_3)/3 \tag{2}$$
   (d) Defining the threshold value to separate knee implants from other components to be equal to
   $$P_t = P_{avg} - \sigma \tag{3}$$
   where σ represents the standard deviation.
   (e) Identifying any pixel whose intensity is greater than or equal to the threshold value, P_t, as part of knee implants as shown in Fig. 1(b).



(a) Knee image (b) Knee implants

(c) Femur (d) Tibia

Fig. 1. Segmentation of three components

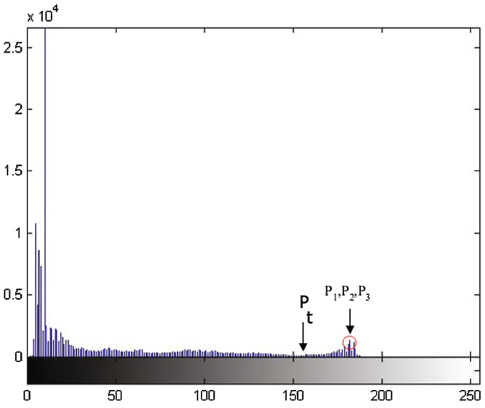


Fig. 2. Histogram of an image

2. **Femur Identification:** Because femur has lower intensities than knee implants thus the same threshold value, P_t can be used to identify femur. An algorithm starts identifying femur from the top row of an image by comparing the intensities of each row with the threshold value, P_t. This process is repeated until the row that contains intensities greater than or equal to the threshold value is reached. The process then stops because this condition indicates that the row contains part of an implant. Figure 1(c) shows an example of femur obtained after the process is terminated.
3. **Tibia Identification:** Tibia has intensities in the same range as femur thus the same algorithm is used to identify tibia except that the algorithm starts to identify tibia from the bottom row of an image until it reaches the row that contains intensities greater than or equal to the threshold value, P_t. Figure 1(d) shows an example of tibia obtained after the process is terminated.

Fig. 3. Filtered knee implant

3.2 Contour Finding

Before the orientations can be estimated, the contours of tibia and femur can be detected by applying the following three steps:

1. **Filtering:** The main purpose of this step is to remove tissues and noise around the bones from an image; however, different filtering techniques are applied to remove tissues around knee implants, femur, and tibia separately due to different characteristics of each component.
   (a) **Knee Implant Filtering:** Tissues and noise around knee implants (as shown in Fig. 1(b)) are removed by applying the following steps:
      i. Filter the region of an image that contains the knee implants using the proposed multiscale dual filtering technique. This is the most important process before the estimation can be performed because the intensities of the tissues and the bones around the knee implants are quite similar, thus a special filtering technique is needed. The proposed technique can remove tissues and noise from an image while preserving the bones and the knee implants by enhancing the edges of the bones while smoothing the tissues to the background as shown in Fig. 3.
      ii. Adjust the contrast of an image by setting

$$k(x,y) = \begin{cases} k_{avg}, & k(x,y) < k_{avg} \\ k(x,y), & k(x,y) \geq k_{avg} \end{cases} \tag{4}$$

where $k(x,y)$ is the intensity of an image at (x,y) and k_{avg} is the average intensity of an image.

Fig. 4. Filtered femur

Fig. 5. Filtered tibia

(b) **Femur Filtering:** Tissues around femur (as shown in Fig. 1(c)) can be removed as follows:
   i. Compare the maximum intensity of the region that contains the femur component, f_{max}, with the maximum intensity of the whole image, I_{max}. If $f_{max} > 0.95 * I_{max}$, it implies that the femur component contains the knee implant then the dynamic range of gray level must be adjusted to have the range between $[f_{min}, f_{max}/2]$ where f_{min} and f_{max} are the minimum and maximum intensities of the femur region, respectively.
   ii. Compare intensity of each pixel, $f(x, y)$, of the femur component with the average intensity, f_{avg} and adjust the intensity as follows:

$$f(x,y) = \begin{cases} f_{avg}, & f(x,y) \leq f_{avg} \\ f_{max}/2, & f(x,y) \geq 0.95 * f_{max} \\ f(x,y), & otherwise \end{cases} \tag{5}$$

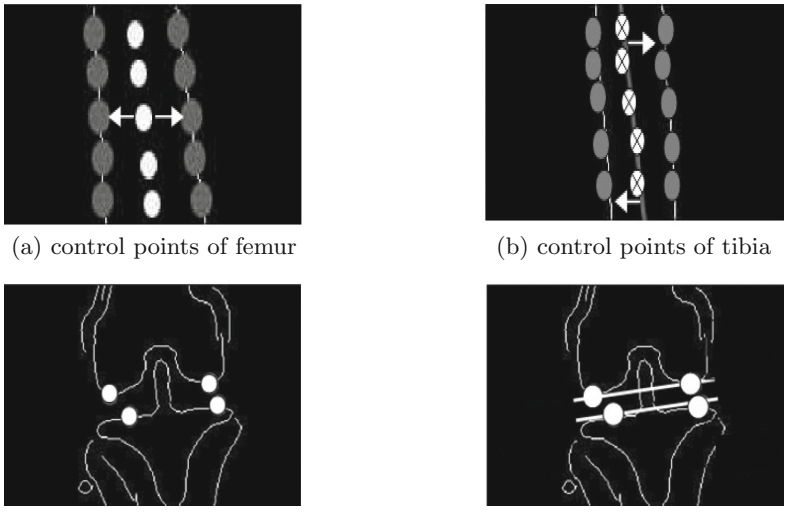
   iii. Erode the femur component with a 15×15 mask one time and then dilate it back with the same mask in order to remove noise and fill up the bone. The result is as shown in Fig. 4.

(c) **Tibia Filtering:** Tissues and noise around tibia can be removed as follows:
   i. Compare the maximum intensity of the region that contains the tibia component, t_{max}, with the maximum intensity of the whole image, I_{max}. If $t_{max} \leq I_{max}/2$ then the dynamic range of gray level must be adjusted to have the range between $[t_{min}, t_{max}/2]$ where t_{min} is the minimum intensity of the tibia region.
   ii. Erode the tibia component with a 15×15 mask twice in order to remove fibula and noise, then dilate it back with the same mask in order to fill up the bone and bring the tibia back to its original size. The result is as shown in Fig. 5

2. **Edge detection:** Before the edges can be detected, the contrast of an image is adjusted using contrast stretching in order to enhance the edges. Then Canny edge detection is used to detect the edges of knee implants, femur, and tibia.
3. **Control point selection and hypothetical line drawing:** Once the edges are detected, the next step is to select control points which are used for hypothetical line drawing. The hypothetical lines are lines that represent the orientations of femur and tibia which are used to estimate the knee implant orientation. The control point selection and line drawing can be performed as follows:
   (a) Femur: A femur is uniformly divided into 5 or more sections, then the left edge and the right edge of the bone are used to find the center point for each section. These center points of all sections are then used as control points for drawing a vertical hypothetical line using linear regression as shown in Fig. 6(a).
   (b) Tibia: A tibia is divided into 5 or more sections, however, tibia may consist of more than two edges because of fibula edges. Thus the algorithm first

finds the midpoint between the rightmost and the leftmost edges, x_{mid}, then scan outward from this midpoint to find the actual right and left edges of tibia which are assumed to be close to the midpoint. Then the control point of each section is located in the middle of these two edges. Finally, linear regression is used to draw a vertical hypothetical line that represents the orientation of tibia from control points of all sections as shown in Fig. 6(b).

(c) Knee Implants: Before the horizontal hypothetical lines that represent the orientation of femoral and tibial components can be drawn, the algorithm selects control points from both components in the following steps: (1) locate the central notch of a femoral component by checking for the inflection points then draw the vertical lines that pass through these points, one for each. The area between these two vertical lines will be discarded during control point selection since it contains the stem of tibial component. (2) locate the lowest points on the left and on the right of a femoral component then draw the line to connect these two points to represent the orientation of a femoral component as shown in Figs. 6(c)–(d). (3) locate the highest points on the left and on the right of a tibial component then draw the line to connect these two points to represent the orientation of a tibial component as shown in Figs. 6(c)–(d).



(a) control points of femur (b) control points of tibia

(c) control points of knee implant (d) lines of knee implant

Fig. 6. Control points of femur, tibia, and knee implants

3.3 Orientation Estimation

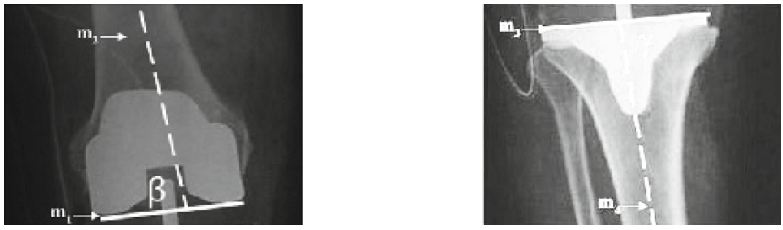
The orientation of a femoral component which is an angle between a femoral component and a hypothetical line of femur (as shown in Fig. 7(a)) can be measured by

$$\beta = tan^{-1}(\frac{m_1 - m_2}{1 + (m_1 * m_2)}) \tag{6}$$

where m_1 is the slope of a horizontal hypothetical line of a femoral component, m_2 is the slope of a hypothetical line of femur. On the other hand, the orientation of a tibial component (as shown in Fig. 7(b)) can be measured by

$$\gamma = tan^{-1}(\frac{m_3 - m_4}{1 + (m_3 * m_4)}) \tag{7}$$

where m_3 is the slope of a horizontal hypothetical line of a tibial component and m_4 is the slope of a hypothetical line of tibia.



(a) femoral orientation (b) Tibia orientation

Fig. 7. Orientation measurement

4 Results and Conclusions

The proposed method was tested on a set of 91 X-ray images where 57 of them are the images of left knee and 34 of them are the images of right knee. Among 91 images, 67 of them are mobile bearing implants and 24 of them are fixed bearing implants. It can be seen from Fig. 1 that an X-ray image contains tissues and noise that can cause errors during orientation estimation. The multiscale dual filter is presented to remove tissues and noise while preserving bones and knee implants which are needed for orientation estimation. The orientation estimation results were evaluated by the orthopedic surgeon and 84 out of 91 are acceptable which means that the angles measured by the proposed algorithm and the angles measured manually by the surgeon differ by no more than two degrees for all of these 84 images. The images that caused failures are the images of patients who have knee replacement surgeries more than one time and the images of patients whose tibial and femoral components of knee implants are too close to each other.

Figure 8 shows a sample of an image of knee implants after the second surgery, it can be seen that the stems of tibial and femoral components are longer than the normal ones. Figure 9 shows a sample of the second case when two components are too close to each other, in this case the algorithm cannot separate the tibial component from the femoral component. The experimental results reveal that the proposed algorithm can effectively estimate the orientations of the knee implants from most of the X-ray images with 92% acceptance rate.

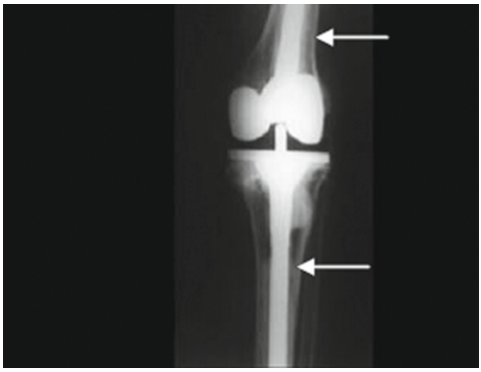


Fig. 8. The image of knee implants after the second surgery

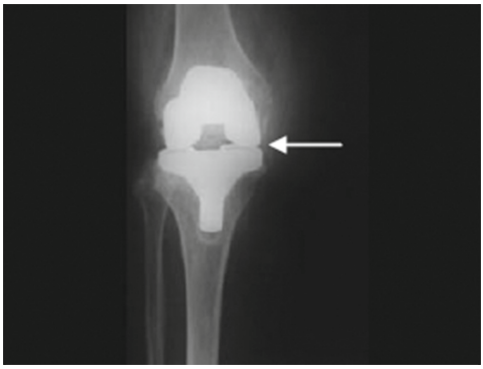


Fig. 9. The image of knee implants with two components too close to each other

References

1. Hailey, J.I., Fisher, J., Dowson, D., Sampath, S.A., Johnson, R., Elloy, M.: A tribological study of a series of retrieved accord knee explants. Med. Eng. Phys. **16**(3), 223–228 (1994)

2. Mintz, L., Tsao, A.K., McCrae, C.R., Stulberg, S.D., Wright, T.: The arthroscopic evaluation and characteristics of severe polyethylene wear in total knee arthroplasty. Clin. Orthop. **273**, 215–222 (1991)
3. Sanzen, L., Sahlstrom, A., Gentz, C.F., Johnell, I.R.: Radiographic wear assessment in a total knee prosthesis. J. Arthroplasty **11**(6), 738–742 (1996)
4. Fukuoka, Y., Hoshino, A., Ishida, A.: Accurate 3D pose estimation method for polyethylene wear assessment in total knee replacement. In: Proceedings of the 10th International Conference on IEEE/EMBS, pp. 1849–1852. IEEE (1997)
5. Fukuoka, Y., Hoshino, A., Ishida, A.: A simple radiograph measurement method for polyethylene wear in total knee arthroplasty. IEEE Trans. Rehabil. Eng. **7**(2), 228–233 (1999)
6. Lavallee, S., Szeliski, R.: Recovering the position and orientation of free-form objects from image contours using 3D distance maps. IEEE Trans. Pattern Anal. Mach. Intell. **17**(4), 378–390 (1995)
7. Banks, A., Hodge, W.A.: Accurate measurement of three-dimensional knee replacement kinematics using single-plane fluoroscopy. IEEE Trans. Biomed. Eng. **43**(6), 638–649 (1996)
8. Walker, S.A., Hoff, W., Komistek, R., Dennis, D.: In vivo pose estimation of artificial knee implants using computer vision. Biomed. Sci. Instrum. **32**, 143–150 (1996)

Bodily Posture Recognition with Weighted Dimension on Kinect Data Stream

Chattriya Jariyavajee(✉), Booncharoen Sirinaovakul, and Jumpol Polvichai

Computer Engineering, King Mongkut's University of Technology Thonburi, Bangkok, Thailand
chattriya.jar@gmail.com, boon@kmutt.ac.th, jumpol@cpe.kmutt.ac.th

Abstract. The characteristic of the data stream is continuous, non-stationary, and very large or infinite size. Data stream classification requires the algorithm that able to classify data instance and learn from data incrementally. In this paper, the algorithm with Weighted Dimension is proposed and applied for the Kinect bodily posture recognition. The human body portions, as the input features, are calculated from Skeleton Joint data. The proposed algorithm successes in recognizing three human postures: stand, sit_on_chair, and sit_on_floor. The result of classification is 99.02% on average and 100% on moving accuracy. Moreover, the algorithm always learns from the data instances and some labels so the algorithm is able to learn whether the data instances are changed. In the other words, the algorithm could handle the concept drift in the data stream.

Keywords: Data stream classification · Concept drift · Kinect · Posture recognition

1 Introduction

In machine learning and data mining, many popular classification algorithms work in real-world applications e.g. drug discovery [1], lane detection [2], medical image recognition [3]. However, these algorithms are inappropriate for organizing data stream which its characteristic is continuous, non-stationary, and very large or infinite size. The data stream requires the algorithm that is fast, less memory usage and robust for data changing over time.

In the past, people used to record the limit number of data samples into a dataset. Many machine learning or data mining algorithms [4] to learn from the dataset was invented e.g. Classification Tree, k-NN, Neural Network, and SVM. However, in the era of big data, data from many devices are independently transmitted to the cloud server. The data, i.e. continuous sensor data, social media stream or network traffic information, becomes real-time and unlimited. This kind of data is called the data stream and some of the methods to handle it are called Online Machine Learning [5].

The data stream characteristic is continuous and not limit the number of data instances. Moreover, some data streams might change their characteristic over a period of time. For example, in the classification of data instance in a data stream, the arriving

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_14

data that is analogous to the previous data might belong to the other class. The changing of the characteristic is called concept drift. Many algorithms are not able to process the data stream because of their computation time and memory usage. Most are improper for the data stream because they normally use the data sample multiple times. For example, every time while decision tree [4] is selecting a splitting node at some particular level of the tree, it uses the data samples for calculating the information gain for every candidate node. In the next level of the tree, it uses the same data samples again.

The data stream with concept drift requires an algorithm that learns fast and is able to adapt their knowledge base over the time. The infinite number of input is continuously fetched to the system one-by-one. Labeler (e.g. human) is required to input the correct label for some of the data instances. Once the classification model learned, it would be able to tackle the data without any labeler because the classification model both classifies and learns from a data instance. The single-time data accessing greatly saves the memory usage and possibly makes the lesser computation time in data stream classification. However, the trade-off is the possibility to get lesser accuracy.

The model learned from a dataset classification algorithm is static. It has a weakness in concept drifting environment because the model cannot adapt itself to the data after concept drift. Even recreating the model when concept drifted is a choice, it requires the mechanism for detecting a concept drift. In this work, we introduce a new approach which is an algorithm that classifies the data stream by using each data sample once and its application to Kinect Bodily Posture Recognition. The recognition includes the three postures: (i) Stand (ii) Sit_on_chair (iii) Sit_on_floor.

2 Related Works

2.1 Distance-Based Classification for Dataset

K-NN [6] is one of the popular distance-based methods for classifying dataset. The database memorizes the data instances and their belonging class. When the unseen data arrives, the algorithm selects the k nearest neighbors to vote the belonging class. The nearest neighbors are picked from the database by choosing the ones that have a minimum distance to the data instance. Usually, the distance is calculated from the Euclidian distance as in the Eq. (1).

$$d(X, Y) = \|X - Y\| = \sqrt{\sum_i (X_i - Y_i)^2} \tag{1}$$

The method of nearest neighbors stores multiple data instances in order that each acts as a prototype or representative for the class. There also be the method called the nearest mean [4]. It stores the means of each class as the representative to reduce the size of database and computation time. Both nearest neighbor and mean methods classify the unseen data instance by majority voting. The class of the N nearest representatives would be the predicted class. This kind of algorithm is called a lazy learning. Its mechanism mimics the human that looks for the similar things in his mind to label the unknown thing.

3 Kinect Bodily Posture Recognition

Kinect [7, 8] is a sensor technology invented by Microsoft. It consists of an infrared projector and camera. The camera features the RBG and depth images. Kinect SDK includes skeleton tracking API which allows the developer to retrieve the coordinates of the human joints. The sensor operates practically in 2–3 m as illustrated in Fig. 1(a). The human joints and coordinates are declared as in the Fig. 1(b).

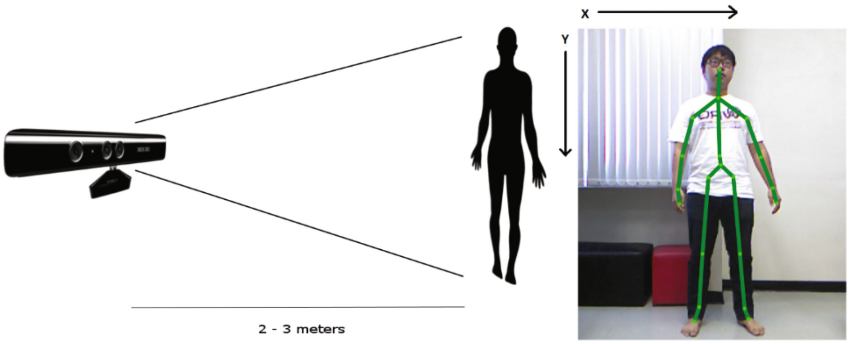


Fig. 1. (a) The operation range of the Kinect. (b) The human joints and XY-coordinates

O. Patsadu et al. [9] proposed three postures recognition {stand, sit down, lie down} by various machine learning models, i.e., Backpropagation Neural Network (BPNN), Support Vector Machine (SVM), Decision Tree and Naïve Bayes. In the work, data was collected as a training and testing set: 7200 samples for the training set and 3600 samples for testing set. The result shows that BPNN and SVM reach 100% accuracy.

C. Youness et al. [10] proposed recognition of 18 human postures by many machine learning techniques. They extracted the features by calculating the length along X- and Y-axis between many skeleton joints. The result showed that SVM, ANN, k-NN, and NB outperformed very high accuracy.

W.J. Wang et al. [11] proposed the recognition of human postures {standing, sitting, stooping, kneeling, lying}. Before processing with the algorithm, they extracted the features (i) The center of gravity; (ii) The ratio of the upper and lower human body; and (iii) The tip points (head, hands and feet) along contour of silhouette Then, the features are used in LVQ neural network for classification. This works results in about 99% accuracy.

C.W. Chang et al. [12] proposed posture recognition for universal semaphore signals. The training and testing sets from Kinect skeleton data were recorded. They introduced the using of Self-Organizing Map algorithm for grouping the similar posture and classifying the semaphore signal of the posture.

H.C. Mo et al. [13] proposed human action recognition {walking, swinging, bending down, sitting, and falling down} in a video sequence. They determine the centroid of a human object, find the left, right, highest and lowest points of the body

image, and calculate the distance from terminal points to the centroid. All of them are features to be used in the Multi-category SVM.

However, the different between the related and this work is the kind of data to be processed. The previous works perform on the dataset. They are not the data stream classification problem which is the focus of this paper.

Since, in previous work, these created models are static. It might cause the problem in the real usage. For example, posture recognition for a person might not get the same result for the others due to the different size of bodies e.g. child and man. In data stream classification, a model is possible to adapt itself for fitting to the person. Hence, we design a classification algorithm using Weighted Dimension to solve this problem.

4 Proposed Work

In data stream classification, the algorithm is designed to have capabilities to classify the data stream and refine model incrementally. Some data instance have their labels taken from labeler. The algorithm learns from the data instance. If the data instance has no label, the algorithm will classify before learn.

4.1 Generic Framework for Data Stream Classification

The framework for data stream classification allows any classification algorithm that provides two methods: (i) classify (ii) learn.

Given the following variables: M is a classification model, X_t is the unseen data instance of N dimensions at time t, Y_t is the belonging class of data instance X_t. y_t is a predicted class. The framework is described by the following instructions.

```
Framework for Data Stream Classification
for each unseen data instance and given label {X_t, Y_t}
  y_t := M.classify(X_t)
  if Y_t does not exist (no belonging class)
    output y_t
    M.learn(X_t, y_t)
  else if Y_t exist and Y_t ≠ y_t
    output Y_t
    M.learn(X_t, Y_t)
```

4.2 Representative and Dimension Weight

In our method, we use the centroid of data instances of the class, as a representative of that class members. We use a single representative per class. Thus, when the unseen instance arrives, we use One Nearest Neighbor classification. The minimum distance from the representative to the data instance are calculated. In some distance calculations, e.g. Euclidian, the importance of every dimension are equal. However, in out

method, we assign the weights to each dimension of each representative and the summation of weights for each class representative must be 1. Because, in the most situation, the dimensions of data samples have the different level of importance. The representatives and dimension weights of the classes are illustrated as in Fig. 2.

		dimension 1	dimension 2	dimension 3
Class 1	R_1	R_{11}	R_{12}	R_{13}
	W_1	W_{11}	W_{12}	W_{13}
Class 2	R_2	R_{21}	R_{22}	R_{23}
	W_2	W_{21}	W_{22}	W_{23}
Class 3	R_3	R_{31}	R_{32}	R_{33}
	W_3	W_{31}	W_{32}	W_{33}

Fig. 2. The illustration of representatives and weights for each class

4.3 Classification Based on Distance

To calculate the distance between the unseen instance X and representative R with the dimension weights W, we perform the Weighted Euclidian distance. The distance calculation formula is shown as the Eq. (2).

$$distance\left(\{R, W\}_c, X\right) = \sum_{i=1}^{D} W_{ci} \cdot \|R_{ci} - X_i\| \tag{2}$$

To classify the unseen data instances, the distance from unseen instance to all representatives would be calculated. The predicted class is the belonging class of the nearest representative. The predicted class could be written as the Eq. (3).

$$predicted\ class = \text{argmin}_{c \in\ ClassSet}\ distance\left(\{\mathrm{R}, \mathrm{W}\}_c, X\right) \tag{3}$$

4.4 The Adaptation of Representative and Dimension Weight

For the learning of classification model, the representatives and dimension weights are the knowledge base of the learning from data instances and labeler. To make the classification model learn incrementally from the data instance, the representatives and weights of the model have to be adapted at every time the data instance arrived.

Given the following notations: (i) $\mathrm{R_c}$ is the representative vector of class c, and (ii) β is the converging factor and in range (0, 1]. The representatives are updated the by following the Eq. (4).

$$R'_c = R_c + \beta(X_t - R_c) \tag{4}$$

Furthermore, the dimension weights could be updated by the level of importance. Since each weight is the multiplier, the distance of each dimension is scaled. The dimension d with the least weighted distance could be counted as the most importance distance. Hence, we look for the dimension d by the Eq. (5).

$$d = \text{argmin}_{j\in[1,D]}\left(W_{cj} \cdot \left\|X_j - R_{cj}\right\|\right) \tag{5}$$

To maintain the summation of weights as 1, we adjust the values of weights by (i) increasing the weight of dimension d (ii) reducing the weights of the others. The adaptation of weights would minimize the total sum of weighted distance so it helps the representatives nearer to the data instance. The adjustment of dimension weight follows the Eq. (6) where α is a learning factor and in range (0, 1].

$$W'_{ck} = \begin{cases} W_{ck} + \alpha(1 - W_{ck}), & k = d \\ W_{ck}(1 - \alpha), & k \neq d \end{cases} \tag{6}$$

The method *classify* and *learn* could be written by the following pseudo codes.

```
M.classify(X)
Initialize min_dist := infinity, predicted_class := empty
for each class c
  calculate dist by the equation (2)
  if dist < min_dist
    min_dist := dist
    predict_class := c
  end if
end for
return predict_class

M.learn(X, c)
Initialize R := 0, W := 1⁄D, α := 0.0005, β := 0.005
Update R_c by the equation (4)
Find the dimension d by the equation (5)
For each dimension k
  Update W_ck value by the equation (6)
end for
```

5 Experiments

5.1 Data Collection and Feature Extraction

The Kinect Skeleton data consists of the XYZ coordinate values of 20 joints. The connections between joints construct the bones as illustrated in the Fig. 3. We write the C# code to retrieve the data through the Kinect SDK at 30 FPS speed.

Fig. 3. The connection between joints of {Stand, Sit on Chair and Sit on floor} postures

Body portions are used as the features. We calculate the lengths along Y-axis between these pairs of joints: (i) *head* and *spine*, (ii) *foot_left* and *spine*, and (iii) *foot_right* and *spine*. We then calculate the height of body skeleton by the Eq. (7).

$$SkeletonHeight = \max\left(Y_{foot_{left}} - Y_{foot_{right}}\right) - Y_{head} \tag{7}$$

Finally, we divide all the lengths with the skeleton height. The division results are the features for the classification. The *foot_left* and *foot_right* to *spine* are separately collected because there is some situation that the human could sit with leg crossed or stand with leg lifted.

5.2 Experiment Setup and Validation

Data stream classification algorithm has two purposes to be validated: (i) the ability to learn from the data stream (ii) the ability to classify unlabeled data instances. We separate the validation into two experiments in which each experiment would accomplish a single purpose.

For the first experiment, the first person is in front of the Kinect sensor with 2–3 m far. He acts the postures: stand, sit_on_chair, and sit_on_floor. For every 1–2 min, he moves his body to another random posture. We kept the data stream for 6 min. The second person, a human labeler, inputs the current one's posture into the system. All of the data instances would be labeled. The expected results of this experiment must show that the representatives converge to some values.

For the second experiment, we organize the same experiment setup but no human labeler gives the input to the system. However, there is a person to input the label for verifying the correctness. We also use the representatives and weights created from the first experiment to be the initial model.

6 Results and Analysis

The result of the first experiment is shown in the Fig. 4 where the X-axis represents the time from 0 to 6 min and Y-axis represents the values of R and W for each dimension of each class. The labels {S, SC, and SF} stand for the posture of human {Stand, Sit on Chair and Sit on Floor} at the particular time interval.

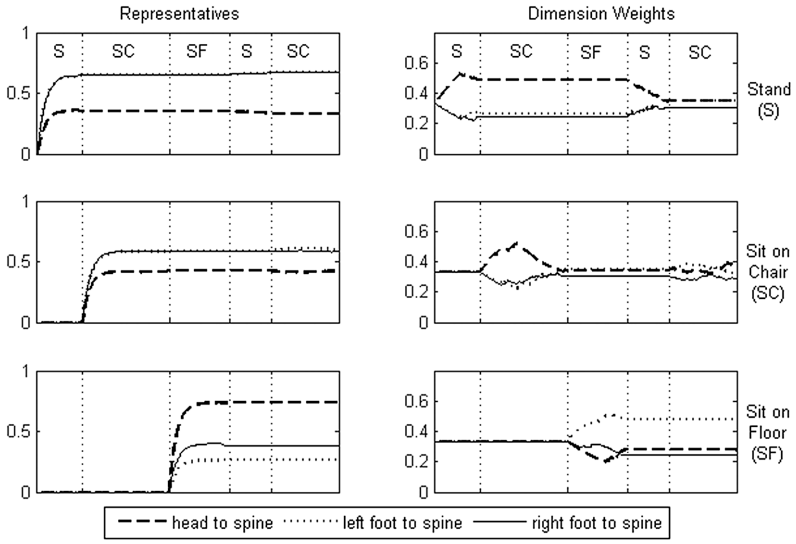


Fig. 4. The representatives and dimension weights of each class

From the result, the representative starts from zero and changes its value to converge to the body portion of each posture. For example, in the first posture, the value of *head* to *spine*, *foot_left* to *spine*, and *foot_right* to *spine* portions converge to 0.3303, 0.6714 and 0.6714 respectively. Whether the values of representatives are converged or not yet converged, the dimension weights adjust the values themselves. The adaptation of weights would minimize the total weighted distance between the data instance and the representative because the lesser dimensional distance would multiply by the higher weight and. the more dimensional distance would multiply by the lesser weight. Hence, the class would be more accurately predicted. The adaptation of R_c and W_c of a class would be performed only in the learning of class c. On the other hand, it would be paused in the learning phase of the other class. For instance, the values of R_{stand} and W_{stand} are adapted in the Stand(S) posture period and remain constants in the Sit_on_Chair(SC) and Sit_on_Floor(SF) posture periods.

In the second experiment, the algorithm performs well in accuracy as shown in the Fig. 5 and the contingency matrix is shown in Table 1. In the normal case, the algorithm predicts completely the same label with the human. However, because the algorithm predicts the next posture in the different time to human inputs the next label,

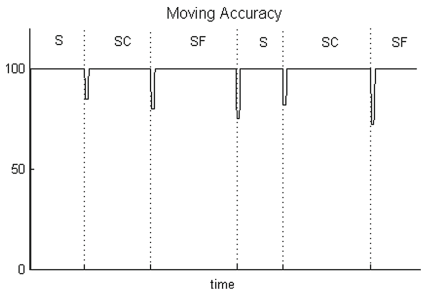


Fig. 5. The moving accuracy of the classification

Table 1. The contigency matrix

Actual	Predict		
	S	SC	SF
S	**2895**	0	0
SC	42	**4226**	0
SF	7	57	**3573**

the moving accuracy is decreasing in the transition between two postures. The average accuracy is 99.02% and the moving accuracy is 100% in the most case.

The high accuracy results are caused by the three reasons: (i) we use the body portions, which directly describe the three postures, as the features; (ii) the distance calculation of the representatives and unseen instance is dimensionally weighted to help in reducing the effect of unimportance features for the class; and (iii) the representatives adapts itself by the data instances so it is always up-to-date to the data instances and the classification would have the higher probability in the correct classifying.

Because the learning is incremental, the algorithm has a capability to handle the data stream with concept drift. When the concept is drifted, the data instances with the label would adapt the model because the framework checks for the wrong classification for every data instances and command the algorithm to learn from it. This mechanism would not cause the change to the other class representatives because it is separated in learning.

7 Conclusions and Future Works

In this work, the algorithm for data stream classification is introduced. It works with the posture recognition using Kinect Skeleton data. The algorithm is tested with three postures: (i) Stand, (ii) Sit on Chair and (iii) Sit on Floor. The recognition system sets a person to stand in front of Kinect sensor about 2–3 m apart. The other person gives the label of posture to the system as an input and validation. The result shows that the algorithm performs well in learning and classifying. The average and moving accuracy of classification is 99.02% and 100%. The algorithm is fast enough to outperform the predicted label and learn the data instance at the speed of 30 FPS. In addition, the learning algorithm is able to adapt itself to fit the data instances and has a capability to handle concept drift. In the future, we plan to test and improve the algorithm robustness and efficiency.

References

1. Lavecchia, A.: Machine-learning approaches in drug discovery: methods and applications. Drug Discov. Today **20**(3), 318–331 (2015)
2. Kim, J., Kim, J., Jang, G.-J., Lee, M.: Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection. Neural Netw. **87**, 109–121 (2017)
3. Zhou, K.: Medical Image Recognition, Segmentation and Parsing. Academic Press, Cambridge (2015)
4. Friedman, J.H., Tibshirani, R., Hastie, T.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York (2001)
5. Shalev-Shwartz, S.: Online learning and online convex optimization. Mach. Learn. **4**(2), 107–194 (2011)
6. Guo, G., Wang, H., Bell, D.A., Bi, Y., Greer, K.: KNN model-based approach in classification. Lecture Notes in Computer Science 2888, pp. 986–996 (2003)
7. Microsoft Corporation. https://developer.microsoft.com/en-us/windows/kinect
8. Microsoft Corporation. https://developer.microsoft.com/en-us/windows/kinect/hardware
9. Patsadu, O., Nukoolkit, C., Watanapa, B.: Human gesture recognition using Kinect camera. In: Ninth International Joint Conference on Computer Science and Software Engineer (JCSSE), Bangkok, Thailand, pp. 28–32 (2012)
10. Youness, C., Abdelhak, M.: Machine learning for real time poses classification using Kinect skeleton data. In: 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV), Beni Mellal city, Morocco, pp. 307–311 (2016)
11. Wang, W.-J., Chang, J.-W., Haung, S.-F., Wang, R.-J.: Human posture recognition based on images captured by the Kinect sensor. Int. J. Adv. Robot. Syst. **13**(2), 54–69 (2016)
12. Chang, C.W., Nian, M.D., Chen, Y.F., Chi, C.H., Tao, C.W.: Design of a Kinect sensor based posture recognition system. In: Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), Kitakyushu, Japan, pp. 856–859 (2014)
13. Mo, H.-C., Leou, J.-J., Lin, C.-S.: Human behavior analysis using multiple 2D features and multicategory support vector machine. In: IAPR Conference on Machine Vision Applications, Yokohama, Japan (2009)

A New Streaming Learning for Stream Chunk Data Classification Based on Incremental Learning and Adaptive Boosting Algorithm

Niphat Claypo(✉), Anantaporn Hanskunatai, and Saichon Jaiyen

Department of Computer Science, King Mongkut's Institute of Technology Ladkrabang, Ladkrabang, Bangkok, Thailand
{55650805, anantaporn.ha, saichon.ja}@kmitl.ac.th

Abstract. Currently, stream data classification is a challenge task to discover new useful knowledge from massive and dynamic data in big data era. This paper proposes a streaming learning method based on the incremental learning using a new adaptive boosting algorithm for stream data. The proposed adaptive boosting consists of a new method for updating distribution weight and the new weight voting. This learning method concentrates on learning from sequential chunks of data stream. The distribution weight updating method uses error of previous hypothesis to update the weight. The learning method uses only one data chunk to create a new hypothesis at a time and after learning, the learned data chunk can be thrown away and can learn the new data chunk without using the previous learned data. The experimental results show that the accuracy of the proposed method is higher than other methods in all datasets.

Keywords: Incremental learning · Adaptive boosting · Stream data · Classification

1 Introduction

Currently, stream data classification is a main task to get useful knowledge from massive and dynamic data. Additionally, the amount of data have been continuously generated every day. Although the ensemble learning is a popular method for stream data classification, the learning method still focus on distribution weight updating and majority voting method [1]. Learn++ was the incremental learning based on ensemble learning [2]. The structure of Learn++ consisted of multiple neural networks combined together. This method was able to learn the data in incremental fashion by creating a new based classifiers and updating the current voting weights based on the distribution weights of previous dataset. In each iteration, Learn++ created the new based classifier and combined all classifiers by using weight voting. Learn++ was able to learn effectively but required much more computational resources. Learn++.NC improved the learning method of Learn++ to solve problems when it learned a new class introduced by a subsequent dataset [3]. The weight voting of Learn++.NC focused on the new class of additional data. Learn++.NC created the new classifiers with a new dataset and combined them by using dynamically weighted consulting and voting (DW-CAV). A new neural learning method based on

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_15

radial-shaped function and discarding the learned data after learning from stream data was presented in [4]. This method was called Class-wise Incremental Learning (CIL). The structure of the CIL method was adapted from the concept of one-pass-throw-away learning as introduced in [5]. The experimental results illustrated that the CIL was faster than the other incremental learning methods. A fast incremental extreme learning machine algorithm was proposed in [6]. This approach was based on extreme learning machine called IDS-ELM. IDS-ELM used ELMs as base classifiers and adaptively decided the number of neurons. In addition, IDS-ELM improved the performance by randomly selecting the activation functions from a set of functions.

This paper proposes a streaming learning method based on the incremental learning using a new adaptive boosting algorithm for stream data. A new adaptive boosting technique is the modification of distribution weight updating and weight voting. The proposed learning method concentrates on learning from sequential chunks of data stream. The updating of the distribution weights use error from previous hypothesis to update the distribution weights of current chunk dataset.

2 Problem Description

We define a data stream $DS = \{D_1, D_2, \ldots, D_t, \ldots\}$ as a sequence of data chunk or a set of samples of each time step. Let $D_t = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be the data chunk t where $\mathbf{x}_i$ is the data object i. Each data object $\mathbf{x}_i$ has a label $y_i \in \boldsymbol{Y} = \{y_1, y_2, \ldots, y_c\}$ [7–9]. The conventional classification methods are difficult to classify the stream data because they cannot leave the previous learned data in order to learn a new data. The learning process of the conventional classifiers must combine the previous learned data with the new data and retrain the model. They cannot discard the previous learned data in training dataset. Similarly, for data stream learning, when the conventional classifiers need to learn the new incoming chunk data at time t, these classifiers must combine all previous learned data with new incoming chunk data and retrain the classifiers again.

3 The Proposed Method

This paper proposes a new stream learning method for classifying stream data. The proposed method is based on the incremental learning method and adaptive boosting technique (IABS). This method can incrementally learn the new data chunk by using the distribution weights from the previous weights of previous learned data. In addition, the new incremental learning method for updating distribution weights and voting weights is proposed. The structure of the proposed model is demonstrated in Fig. 1.

Figure 1 illustrates incremental learning model for stream data classification. The learning method consists of two main parts: updates distribution weights in W by using error of current chunk data from the previous hypothesis and creates model at any time t. Finally, the model classifies a new data by using weight voting to predict the outcome of the new data.

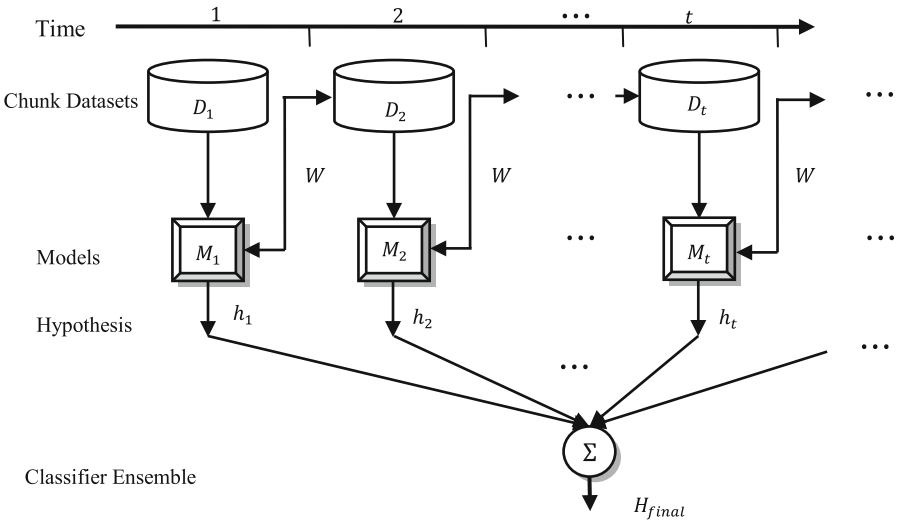


Fig. 1. Incremental learning model for stream data classification

3.1 Distribution Weights Updating

In order to learn incremental data based on adaptive boosting, the distribution weight updating is very important to improve the classification accuracy and to select the samples for learning in the next iteration. The main idea of the proposed method is to use the previous hypothesis to update the distribution weights of data in the current data chunk. The details of distribution weight updating can be described as follows.

Let D_t is the new chunk of training data at time t and $\mathbf{H} = \{h_1, h_2, \ldots, h_t\}$ are the hypothesis and $W_t = \{w_t(1), w_t(2), \ldots, w_t(n)\}$ be the distribution weight of each instances in data chunks. The number of distribution weights in W is equal to the number of instances in each chunk data. At the first time, the initialized distribution of D_1 before update of is calculated as $w_1(i) = 1/n$ because the dataset D_1 has not yet been learned.

In the next time step, the learning method in each time creates the model of the new chunk from data stream without involving the previous learned data chunk. The distribution weight W was updated by using classification error of current chunk dataset that classify by the previous model M_{t-1}.

Then, the error E_t of chunk data D_t is calculated by using previous hypothesis h_{t-1} as presenting in Eq. (1).

$$E_t = \sum\nolimits_{i, y_i \neq h_{t-1}(\mathbf{x}_i)} w_{t-1}(i) \tag{1}$$

The next step, the normalized error B_t is computed by using the error E_t from Eq. (1) as:

$$B_t = \frac{E_t}{1 - E_t} \tag{2}$$

This normalized error B_t is used to update distribution weights of current chunk data. The next step, the distribution weight W_{t-1} is updated by using B_t in Eq. (2) as:

$$w_t(i) = w_{t-1}(i) \times \begin{cases} B_t, & if\ h_{t-1}(\mathbf{x}_i) = y_i \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

where w_{t-1} is the distribution weight of previous time $t-1$. After updating the distribution weights of all instances, these weights are normalized as follows:

$$W_t = w_t / Z_t \tag{4}$$

where Z_t is the normalize value and $Z_t = \sum_{i=1}^{n} w_t(i)$. Finally, the sum of distribution weight w_t should be equal to 1 as follows:

$$\sum\nolimits_{i=1}^{n} w_t(i) = 1 \tag{5}$$

From the distribution weight, the instances $\mathbf{x}_i$ with the largest distribution weight should be more likely to be selected to create a training dataset at time $t+1$.

3.2 Model Learning and Classifying

At time t, the bootstrap training data D_t^* is drawn according distribution weight w_t from data chunk D_t. After that the new based classifier as the current classifier is created and trained by using D_t^* to obtain the hypothesis h_t. Then, the error ε_t of h_t is calculated as follows:

$$\varepsilon_t = \sum\nolimits_{i=1}^{n} w_t(i)[h_t(\mathbf{x}_i) \neq y_i] \tag{6}$$

where w_t is the distribution weight of the sample i that is incorrectly classified. For a two-class classification problem, if the error ε_t is greater than 0.5, then hypothesis h_t will be discarded and go to the step of selecting the new training data to train the new based classifier. For the multiclass classification problems, the error generated by random guessing ε_t should be greater than $(N-1)/N$ where N is the number of classes. Therefore, if the error ε_t is greater than $(N-1)/N$, then hypothesis h_t should be discarded. The next step, the normalized error β_t is computed as follows:

$$\beta_t = 1/(1-\varepsilon_t) \tag{7}$$

The β_t is the weight of each hypothesis and is used to combine the classifiers in order to predict the class label of unknown data $\mathbf{x}$ in the final step. The proposed method repeats the same procedure when the next chunk of new datasets D_{t+1} is fed.

After generating all classifiers of the specific problem, the output of the proposed model is calculated as follows:

$$h_{final} = \arg\min_{y \in C} \sum\nolimits_{t=1}^{T} \log\left(\frac{1}{\beta_t}\right)[h_t(\mathbf{x}) = y] \tag{8}$$

where T is the number of incrementally developed hypotheses in the learning process and h_{final} is the final classification decision of all hypothesis voted by weight vote of each hypothesis. The proposed learning algorithm is detailed in algorithm 1.

Algorithm 1: Incremental learning for classification stream chunk data (IABS)

Input at time t:

- A dataset $D_t = \{(\mathbf{x}_i, \omega_1), \ldots, (\mathbf{x}_n, \omega_j)\}, i = 1{:}n$ where ω_i is the class label of $\mathbf{x}_i$.
- Hypothesis h_{t-1}.
- Distribution weight from previous time $W_{t-1} = \{w_1, w_2, \ldots, w_n\}$

Output:

The final hypothesis

$$h_{final} = \underset{y \in C}{\operatorname{argmin}} \sum_{t=1}^{T} \log\left(\frac{1}{\beta_t}\right)[h_t(\mathbf{x}) = y]$$

Method:

1. **If** $t = 1$ **then** initialize $w_t(i) = 1/n$ and go to step 6.
2. Calculate the error of chunk data D_t

$$E_t = \sum_{i, y_i \neq h_{t-1}(\mathrm{x}_i)} w_{t-1}(i)$$

3. Calculate normalized error

$$B_t = \frac{E_t}{1 - E_t}$$

4. Update the distribution weight in W_{t-1}

$$w_t(i) = w_{t-1}(i) \times \begin{cases} B_t, & if\ h_{t-1}(\mathbf{x}_i) = y_i \\ 1, & \text{otherwise} \end{cases}$$

5. Normalize the distribution weight

$$W_t = W_t / Z_t$$

6. Draw bootstrap training data D_t^* according to the distribution weight W_t from data chunk D_t.
7. Train the based classifier with D_t^* to obtain the hypothesis h_t.
8. Calculate the error of h_t :

$$\varepsilon_t = \sum_{i=1}^{n} w_t(i)[h_t(\mathbf{x}_i) \neq y_i]$$

9. **If** the error $\varepsilon_t > 0.5$ **then** discard hypothesis $h_t$and go to step 6.
10. Compute the normalized error β_t:

$$\beta_t = 1/(1 - \varepsilon_t)$$

11. Repeat the same procedure when the next chunk of new datasets D_{t+1} is received.

4 The Experimental Results

For performance evaluation, the proposed IABS method was tested by six real-world datasets from the University of California at Irvine (UCI) Repository and the experimental results were subsequently compared with Learn++ and Online Bagging algorithms. In this section, the details of these six datasets were firstly explained. Then, the experimental setup was described, and the experimental results between these two methods were reported.

4.1 Datasets

This experiment focused on six real-world datasets from the University of at Irvine (UCI) Repository of the machine learning database [10]. The datasets contain both of two-class and multi-class classification. The detailed information of these datasets is shown in Table 1.

Table 1. The characteristics of the datasets.

Datasets	Number of instances	Number of attributes	Number of classes
Spambase	4601	57	2
Magic	19020	10	2
EEG	14980	15	2
Credit Card	30000	24	2
Abalone	4177	8	3
Sat	6435	36	6

4.2 Performance Evaluation

In this paper, the performance of all classification models are measured by true positive rate of each class and accuracy rate [11]. The true positive rate TP_{rate} is the percentage of positive instances correctly classified by classifier. The true positive rate TP_{rate} can be computed as:

$$TP_{rate} = \frac{TP}{TP + FN} \tag{9}$$

where TP is the number of positive instances correctly classified by classifier and FN is the number of positive instances misclassified by classifier. The accuracy rate Acc is most commonly used in empirical measure. The accuracy rate can be compute as:

$$\text{Acc} = \frac{TP + TN}{TP + FN + FP + TN} \tag{10}$$

where TN is the number of negative instances correctly classified by classifier and FP is the number of negative instances misclassified by classifier. In this experiment, the true

positive rate is applied to evaluate the prediction accuracy for each individual class and the accuracy rate is used to evaluate the overall classification performance for each dataset.

4.3 Experimental Setup

For this experiment, it was simulated on Matlab. As previously mentioned, the stream chunk data were focused. Therefore, each dataset was divided into 10 chunks with equal size, by initial randomizing instances within each dataset. For each of 10 experiments, one chunk was used as the testing set and the other chunks were used as the training sets. The accuracies obtained from these 10 experiments were averaged and reported as the performance of each method. For our proposed method and Online Bagging method, CART is adopted as BasedClassier to create the ensemble model. For the structure of Learn++, the number of hidden neurons in hidden layer of MLP was set as 10.

4.4 Classification Results

The experimental results between our proposed IABS method and the other incremental learning algorithm, Learn++ and Online Bagging, were summarized among all six datasets, as demonstrated in Table 2.

The Spambase dataset consists of 57 attributes and 4,601 instances with 2 classes. Table 2 shows the average accuracy of each class and overall accuracy between the proposed IABS method, Learn++ and Online Bagging. According the results, the overall accuracy of the proposed IABS are slightly higher than that of Learn++ and obviously higher than that of Online Bagging. For each class, the accuracy of the proposed method is higher than that of Learn++ for class 1 but it is lower than that of Online Bagging for class 2. The Magic dataset consists of 10 attributes and 19,020 instances with 2 classes. For this dataset, it shows that the overcall accuracies of the proposed IABS method are higher than that of Learn++ and Online Bagging. For each class, the accuracy of the proposed method is higher than that of Learn++ for class 1.

For Credit card dataset, it consists of 24 attributes and 30,000 instances with 2 classes. The experimental results gained by this dataset and Magic dataset show that the overall accuracy and the accuracy of class 1 are higher than Learn++ and Online Bagging, while it provides the lower accuracy than the others for class 2.

The EEG consists of 15 attributes and 14,980 instances with 2 classes, the experimental results illustrate that the proposed IABS method outperforms the other incremental learning algorithms including Learn++ and Online Bagging for overall accuracy. It can achieve significantly higher overall accuracy and the accuracy of class 1 than Learn++ and Online Bagging but it is lower than that of Learn++ for class 2.

The Abalone dataset consists of 8 attributes and 4,177 instances with 3 classes. Especially, the experimental results illustrate that the proposed IABS method outperforms the other incremental learning algorithms. It can achieve significantly higher accuracy than that of Learn++, for all individual classes and overall classes.

Table 2. Performance between the proposed IABS method, Learn++ and online bagging for stream chunk data classification.

Datasets	Methods	Prediction accuracy of 10 chunks							Percentage increase
		Class 1	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Class 6*	*Overall*	
Spambase	IABS	**93.93**	93.92	–	–	–	–	**93.93**	
	Learn++	90.79	92.63	–	–	–	–	91.93	2.17
	Online Bagging	73.61	**99.35**	–	–	–	–	77.95	20.50
Magic	IABS	**87.38**	85.82	–	–	–	–	**86.93**	
	Learn++	85.22	82.74	–	–	–	–	83.42	4.21
	Online Bagging	72.25	**98.49**	–	–	–	–	75.16	15.66
Credit card	IABS	**84.54**	67.61	–	–	–	–	**82.31**	
	Learn++	68.03	82.47	–	–	–	–	81.09	1.51
	Online Bagging	78.16	**88.08**	–	–	–	–	78.23	5.22
EEG	IABS	**85.21**	86.23	–	–	–	–	**85.63**	
	Learn++	58.73	**99.43**	–	–	–	–	71.09	20.45
	Online Bagging	75.01	69.32	–	–	–	–	61.12	40.10
Abalone	IABS	**53.85**	**55.30**	**70.90**	–	–	–	**60.41**	
	Learn++	47.33	54.26	70.49	–	–	–	55.23	9.37
	Online Bagging	50.69	52.17	65.33	–	–	–	58.42	3.41
Sat	IABS	93.47	**97.56**	**88.93**	**68.75**	**90.44**	86.71	**88.85**	
	Learn++	**94.39**	95.99	87.99	55.23	83.30	80.87	85.57	3.83
	Online Bagging	91.10	96.18	87.62	68.35	84.52	**87.16**	87.36	1.71

For Sat dataset, it consists of 36 attributes and 6,435 instances with 6 classes. It also found that the overall accuracy of the proposed IABS method are higher than those of Learn++ and Online Bagging, but lower than that of Learn++ for class 2 and Online Bagging for class 6.

Considering to the two-classes datasets, the proposed IABS can achieve significantly higher overall accuracy than online bagging method in all datasets and significantly higher accuracy than Learn++ in EEG dataset. Especially in EEG dataset, the proposed IABS can enhance the classification overall accuracy up to 40.10% and 20.45% from online bagging and Learn++ respectively. For the multiclass class datasets, the proposed IABS has slightly higher overall accuracy than learn++ and online bagging method in all datasets. In summary, the experimental results show that our proposed IABS method outperform the other incremental learning algorithms.

5 Conclusion

This paper proposes an incremental adaptive boosting algorithm for stream data. The proposed IABS adopts adaptive boosting technique and incremental learning methods for learning stream datasets. For the proposed algorithm, it uses only one chunk data to update distribution weight and create hypothesis of each time. The distribution weight of presently time depends on the distribution weight and the knowledge of the previous time. In each chunk data, the proposed method creates a new hypothesis and combines these hypotheses by using a new weight voting in order to improve the performance of classification in the final step. For performance evaluation, the proposed IABS method was tested by six real-world datasets, each dataset was subsequently divided into 10 chunks of equal size, and the results were compared with those of the other incremental learning algorithms, Learn++ and Online Bagging. According to the results, it showed that our proposed IABS method can outperform Learn++ and Online Bagging. Furthermore, it can achieve higher overall accuracy than the others for two-class and multi-class datasets. Therefore, the proposed incremental learning method can efficiently learn and classify the stream data with high accuracy and dramatically reduce the computational time and resources. Since the distribution weight updating of the proposed learning method uses only one chunk data, the computational time is less.

References

1. Oza, N.C.: Online bagging and boosting. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 2340–2345 (2005)
2. Polikar, R., Upda, L., Honavar, V.: Learn++: an incremental learning algorithm for supervised neural networks. IEEE Syst. Man Cybern. **31**, 497–508 (2001)
3. Topalis, M.A., Polikar, R., Learn, N.C.: Combining ensemble of classifiers with dynamically weighted consult-and-vote for efficient incremental learning of new classes. IEEE Neural Netw. **20**, 152–168 (2008)
4. Junsawang, P., Phimoltares, S., Lursinsap, C.: A fast learning method for streaming and randomly ordered multi-class data chunks by using one-pass-throw-away class-wise learning concept. Expert Syst. Appl. **26**, 249–266 (2016)
5. Jaiyen, S., Lursinsap, C., Phimoltares, S.: A very fast neural learning for classification using only new incoming datum. IEEE Neural Netw. **21**, 381–392 (2010)
6. Xu, S., Wang, J.: A fast incremental extreme learning machine algorithm for data streams classification. Expert Syst. Appl. **65**, 332–344 (2016)
7. Pang, S., Ozawa, S., Kasabov, N.: Incremental linear discriminant analysis for classification of data streams. IEEE Syst. Man Cybern. **25**, 1901–1914 (2005)
8. Nguyen, H.L., Woon, Y.K., Ng, W.K.: A survey on data stream clustering and classification. Knowl. Inf. Syst. **45**, 535–569 (2015)
9. Sun, Y., Tang, K., Minku, L.L., Wang, S., Yao, X.: Online ensemble learning of data streams with gradually evolved classes. IEEE Knowl. Data Eng. **28**, 1532–1545 (2016)
10. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository [Online] (2007). http://www.ics.uci.edu/∼mlearn/MLRepository.html
11. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Syst. Man Cybern. **42**, 463–484 (2012)

An Application of Data Mining and Machine Learning for Weather Forecasting

Risul Islam Rasel[1(✉)], Nasrin Sultana[2], and Phayung Meesad[3]

[1] Department of Computer Science and Engineering,
International Islamic University Chittagong, Chittagong, Bangladesh
risul-islam@cse.iiuc.ac.bd
[2] Department of Computer Science and Engineering, University of Chittagong,
Chittagong, Bangladesh
nasrin_cse@cu.ac.bd
[3] Department of Information Technology Management,
KMUTNB, Bangkok, Thailand
phayung.m@it.kmutnb.ac.th

Abstract. Weather forecasting for an area where the weather and climate changes occurs spontaneously is a challenging task. Weather is non-linear systems because of various components having a grate impact on climate change such as humidity, wind speed, sea level and density of air. A strong forecasting system can play a vital role in different sectors like business, agricultural, tourism, transportation and construction. This paper exhibits the performance of data mining and machine learning techniques using Support Vector Regression (SVR) and Artificial Neural Networks (ANN) for a robust weather prediction purpose. To undertake the experiments 6-years historical weather dataset of rainfall and temperature of Chittagong metropolitan area were collected from Bangladesh Meteorological Department (BMD). The finding from this study is SVR can outperform the ANN in rainfall prediction and ANN can produce the better results than the SVR.

Keywords: Data mining · Machine learning · SVM · ANN · Weather forecasting · Temperature · Rainfall

1 Introduction

The climate is the condition of the environment, whether it is hot or cool, wet or dry, quiet or stormy, clear or shady [1–3]. Most climate marvels happen in the troposphere, just beneath the stratosphere. Weather prediction is one of the most challenging tasks to accomplish because many natural and man-made components are involved in weather change such as change of seasons, greenhouse effect, deforestation etc. Those collectively make weather prediction more challenging [4, 5].

Weather prediction plays a significant role in many components in decision making related to many fields such as agriculture, business, tourism, energy management, human and animal health etc. [1]. Climate determining includes anticipating how the present circumstance with the air will change in which display atmosphere conditions

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_16

are taken by ground recognition, for example, from boats, plane, Radio sound, Doppler radar, and satellites. The gathered information is then sent to meteorological centers in which the data are assembled, examined, and made into a combination of frameworks, maps, and diagrams. Calculations trade countless onto the surface and upper air maps, and draw the lines on the maps with help from meteorologists. Calculations draw the maps and foresee how the maps will take a gander eventually later on. The determination of atmosphere condition utilizing calculations is plot as numerical or computational weather prediction. Generally the climate and atmosphere expectation issues have been seen as various disciplines [1, 4]. Numerical Weather Prediction (NWP) is urgently subject to characterizing an exact starting state and running at the most astounding conceivable resolutions, while atmosphere prediction has tried to incorporate the full multifaceted nature of the Earth framework keeping in mind the end goal to precisely catch long time-scale varieties and inputs deciding the present atmosphere and potential atmosphere change. The idea of a unified or seamless structure for climate and atmosphere expectation has pulled in a lot of consideration in the most recent couple of years the field of Data Mining and Machine learning has progressed rapidly over the last few decades [5–7]. Predictive analysis has gone to a very new level with the use of machine learning techniques. Weather data used in this study data are dependent on their nature and thus, their estimation is not effectively made with numerous quantitative methodologies. However, they can be portrayed, estimate and arranged quantitatively by utilizing probability theory.

The goal of this paper is to find the challenging pattern of weather of Chittagong, Bangladesh and to predict the weather. To tackle these challenges, we use a jointly predicts rainfall and temperature across space and time. The study combines a bottom-up predictor for each individual variable with a Support Vector Regression (SVR) [11] and an Artificial Neural Network (ANN) model [12] to determine an effective and efficient model. So, a comparative study between these algorithms is done in this study. The climate of Chittagong is described by tropical storm atmosphere. The dry and cool season is from November to March; the pre-storm season is from April to May which is exceptionally hot. The sunny and the rainstorm seasons are from June to October, which is warm, overcast and wet.

2 Methodology

2.1 Support Vector Regression (SVR)

SVM regression [11] performs linear regression in the high dimension feature space using ε – insensitivity loss and, at the same time tries to reduce model complexity by minimizing $\|\omega\|^2$. This can be described by introducing slack variables ξ_i and ξ_i^* where $i = 1, \ldots, n$ to measure the deviation of training sample outside ε - sensitive zone [8, 9].

$$\frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{1}$$

$$\min \begin{cases} y_i - f(x_i, \omega) \leq \varepsilon + \xi_i^* \\ f(x_i, \omega) - y_i \leq \varepsilon + \xi_i \\ \xi_i, \xi_i^* \geq 0, \quad i = 1 \ldots n \end{cases} \tag{2}$$

This optimization problem can transform into the dual problem and solution is given by

$$f(x) = \sum_{i=1}^{n_{sv}} (\alpha_i - \alpha_i^*) K(x_i, x) \tag{3}$$

$$\text{Subject to,} \quad 0 \leq \alpha_i^* \leq C, 0 \leq \alpha_i \leq C$$

Where n_{sv} is the number of support vector (SVs) and the kernel function

$$K(x, x_i) = \sum_{j=1}^{m} g_j(x) g_j(x_i) \tag{4}$$

It is well-known that SVM generalization performance depends on a good setting of meta-parameters C, ε and kernel parameters [9].

2.2 Artificial Neural Network (ANN)

ANN [10, 12] is based on a large collection of artificial neurons mathematically simulating the biological brain in solves problems. ANN can perform as a linear or non-linear function mapping from input data to output target. Multilayer perceptron (MLP) is a one of the most well-known neural networks able to learn any nonlinear function if there are enough training data and given a suitable number of neurons.

The activation function of the artificial neurons in ANNs implementing the back-propagation algorithm is a weighted sum (the sum of the inputs x_i multiplied by their respective weights w_{ji}) [12]:

$$A_j(\bar{x}, \bar{w}) = \sum_{i=0}^{n} x_i w_{ji} \tag{5}$$

The activation depends only on the inputs and the weights. If the output function would be the identity, then the neuron would be called linear. The most common output function is the sigmoidal function [12]:

$$O_j(\bar{x}, \bar{w}) = \frac{1}{1 + e^{A_i(\bar{x}, \bar{w})}} \tag{6}$$

The goal of the training process is to attain a desired output when certain inputs are given. Since the error is the difference between the actual and the desired output, the error depends on the weights, and we need to adjust the weights in order to minimize the error. We can define the error function for the output of each neuron [12]:

$$E_j(\bar{x}, \bar{w}, d) = (O_j(\bar{x}, \bar{w}) - d_j)^2 \tag{7}$$

The error of the network will simply be the sum of the errors of all the neurons in the output layer:

$$E(\bar{x}, \bar{w}, \bar{d}) = \sum\nolimits_j (O_j(\bar{x}, \bar{w}) - d_j)^2 \tag{8}$$

The backpropagation algorithm now calculates how the error depends on the output, inputs, and weights. After that, it adjusts the weights using the gradient descendent method [12]:

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} \tag{9}$$

The goal of the backpropagation algorithm is to find the derivative of E in respect to w_{ji}. First, we need to calculate how much the error depends on the output, which is the derivative of E in respect to O_j (from (7)).

$$\frac{\partial E}{\partial O_j} = 2(O_j - d_j) \tag{10}$$

And then, how much the output depends on the activation, which in turn depends on the weights. From (5) and (6):

$$\frac{\partial O_j}{\partial w_{ji}} = \frac{\partial O_j}{\partial A_j} \frac{\partial A_j}{\partial w_{ji}} = O_j(1 - O_j)x_i \tag{11}$$

And from (10) and (11) it can be seen that:

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial O_j} \frac{\partial O_j}{\partial w_{ji}} = 2(O_j - d_j)O(1 - O_j)x_i \tag{12}$$

The adjustment to each weight will come from (9) and (12):

$$\Delta w_{ji} = -2\eta(O_j - d_j)O_j(1 - O_j)x_i \tag{13}$$

Now, we can use the Eq. (13) for training an ANN with two layers [12].

3 Evaluation Process

3.1 Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) [9, 10] or Root Mean Square Deviation (RMSD) is a commonly used measure of the contrasts between qualities anticipated by a model or an estimator and the qualities really observed. The RMSE serves to total the extents of the

errors in forecasts for different times into a solitary measure of prescient force. RMSE is a decent measure of precision, however just to analyze estimating blunders of various models for a specific variable and not between variables, as it is scale-dependent.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_t - \widehat{y}_t)^2}{n}} \tag{14}$$

Here, y_t is the original value of a point for a given time t; n is the total number of fitted points, and $\widehat{y}_t$ is the fitted forecast value for the time t [9].

3.2 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) [9, 10] is a typical measure of figure mistake in time series data where the expressions "Mean Absolute Deviation" is occasionally utilized as a part of perplexity with the more standard meaning of mean absolute deviation.

$$MAE = \frac{SAE}{N} = \frac{\sum_{i=1}^{n} |x_i - \widehat{x}_i|}{N} \tag{15}$$

Here, x_i is the actual observations time series, $\widehat{x}_i$ is the estimated or forecasted time series. SAE is the Sum of the Absolute Error. N is the number of non-missing data points [9].

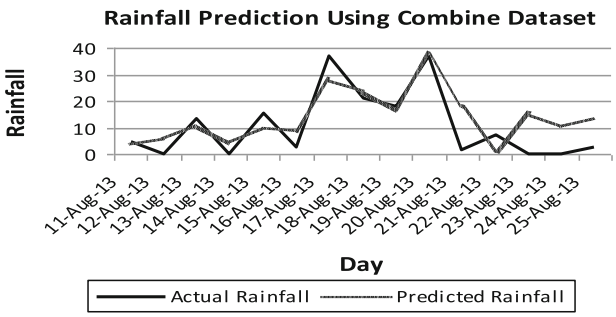


Fig. 1. Rainfall prediction using SVR with combine dataset

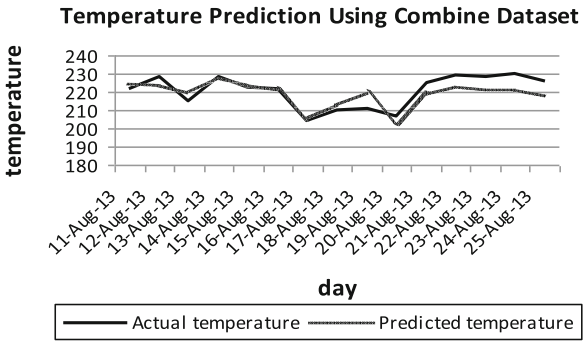


Fig. 2. Temperature prediction using SVR with combine dataset

4 Experiment Design

4.1 Windowing Operator Analysis

Windowing operator transforms a given example set containing series data into a new example set containing single valued examples. For this, windows with a specified window and step size are moved across the series and the attribute value lying horizon values after the window end is used as label which should be predicted [8].

Table 1 illustrates the parameter set up for windowing input into support vector regression model. The values of parameters here included only the best-optimized combination for forecasting rainfall and temperature. Horizon means how many days a-head to predict where training and testing window width are the major part for training the model in order to predict future value based on that learning. Step size is the sliding a-head value of the window that feed the input set into machine learning process. From Table 1, it is cleanly seen that all the parameters setup are same for both domain. These values are obtained by doing repetitive simulation process for 1 day a-head, 7 days a-head, and 10 days a-head future value prediction.

Table 1. Sliding window parameter for SVR

Model	Horizon	Training window width	Step size	Testing window width	Cumulative training
Rainfall	1	5	1	5	No
	7	5	1	5	
	10	5	1	5	
Temperature	1	5	1	5	
	7	5	1	5	
	10	5	1	5	

Table 2 describes the parameter setting for Artificial Neural Network (ANN) model. From Tables 1 and 2, there is a clear indication about training and testing window setup

Table 2. Sliding window parameter for ANN

Model	Horizon	Training window width	Step size	Testing window width	Cumulative training
Rainfall	1	2	1	2	No
	7	2	1	2	
	10	2	1	2	
Temperature	1	2	1	2	
	7	2	1	2	
	10	2	1	2	

that SVR needs more input than the ANN for producing good prediction results and ANN needs less input for training the model.

Kernel parameter analysis is one of the most important parts of the SVR simulation. Because appropriate kernel selection and optimized kernel parameter findings play vital rules for producing less erroneous results. In this work RBF, Gaussian, Polynomial, ANOVA and Neural kernels are analyzed with different parameter but only ANOVA produced better results among those kernels with priory mentioned parameters setup in Table 3.

Table 3. Kernel analysis for SVR

Model	Horizon	Kernel type	C
Rainfall	1	ANOVA	100
	7		120
	10		200
Temperature	1		100
	7		150
	10		300

Table 4 shows the optimized parameter setting for ANN models. Here training cycle, learning rate and values of *M* for every model setup is almost same. Every model uses 2 hidden layers for producing weighted input values for machine learning process.

Table 4. ANN parameter settings

Model	Horizon	Training cycle	Learning rate	M	Hidden layer
Rainfall	1	120	0.3	0.2	2
	7	120	0.3	0.2	
	10	110	0.3	0.2	
Temperature	1	120	0.3	0.2	
	7	100	0.3	0.2	
	10	110	0.3	0.2	

5 Experiment Results

Table 5 shows the result analysis for rainfall prediction using SVR and ANN model. Two types of simulations were undertaken, one is using only historical rainfall dataset and other is using a combined rainfall and temperature dataset for predicting only rainfall. Two evaluation processes RMSE and MAE were applied for understanding the error. From Table 5, it can be said that rainfall has a clear impact on temperature because when combined dataset were used the error rate were minimal than the only rainfall dataset produced. In addition, SVR outperformed ANN in predicting rainfall as it produced 0.95% and 0.17% error rate in both single and combined dataset.

Table 5. Rainfall prediction result

Model	Horizon	Rainfall using only rainfall dataset (Aug'13–Dec'14)		Rainfall using rainfall and temperature combine dataset (Aug'13–Dec'14)	
		(RMSE)	(MAE)	(RMSE)	(MAE)
SVR	1	20.33	**0.95**	19.88	1.93
	7	27.68	1.71	27.6	**0.17**
	10	28.57	2.25	30.96	4.51
ANN	1	21.41	3.54	18.43	2.42
	7	31.97	**10.87**	27.53	**11.33**
	10	27.34	1.02	25.84	3.31

Bold symbolizes the maximum and minimum error rate among the others values.

Table 6 shows the outcomes for temperature prediction using both SVR and ANN. From Table 6 it shows that ANN outperforms SVR for both single and combined dataset in temperature prediction. For ANN, the activation function which has the most significance in modeling needs non negative or non positive values as input rather than

Table 6. Temperature prediction result

Model	Horizon	Temperature using only rainfall dataset (Aug'13–Dec'14)		Temperature using rainfall and temperature combine dataset (Aug'13–Dec'14)	
		(RMSE)	(MAE)	(RMSE)	(MAE)
SVR	1	4.27	5.3	5.03	6.25
	7	9.82	4.18	11.7	**5.71**
	10	9.98	**2.56**	12.32	6.34
ANN	1	3.31	2.29	4.14	4.86
	7	7.89	**0.72**	8.4	**1.72**
	10	7.96	5.46	8.03	1.98

Bold symbolizes the maximum and minimum error rate among the others values.

the zero values for proper execution of the model. So in this experiment we used Leaky rectified linear unit (*Leaky ReLU*) which allows a small, nonzero gradient when the unit is not active [13].

6 Conclusion and Future Works

The purpose of this study was to observe weather forecasting performance of different Machine Learning and Data Mining techniques to propose a weather forecasting model to forecast weather with high accuracy. Two well-known data mining techniques: Support Vector Regression (SVR) and conventional Artificial Neural Network (ANN) were used to conduct the study. The data were fed to the algorithms using conventional windowing technique to train and test the model. A sliding window validation process was done to find convenient amount of training and testing input set to feed into machine learning process. Experiments were done using the same size of window for both SVR and ANN. However, Tables 1 and 2 show only the best fit of the training and testing window parameters settings, which have produced good results in forecasting rainfall and temperature. RMSE and MAE error calculation approaches were applied to calculate the error margin between actual and predicted values. The finding from this study is; SVR can outperform the ANN in rainfall prediction with marginal error rate using both types of dataset and ANN can produce the better results than the SVR with acceptable deviation of error rate.

In this study, dataset from a single station of a country have been used, other datasets were not applied into proposed techniques in order to compare the results. Only 6-year dataset was considered to build the models. For ANN models, maximum 3 hidden layer networks were used in this study. In future work, different dataset from different areas of the world will be applied and different settings of hidden layers in ANN and other different types of kernel for Support Vector Regression will be experimented.

References

1. Xiong, L., O'Connor, K.M.: An empirical method to improve the prediction limits of the glue methodology in rainfall–runoff modeling. J. Hydrol. **349**(1), 115–124 (2008)
2. Wu, J., Huang, L., Pan, X.: A novel Bayesian additive regression trees ensemble model based on linear regression and nonlinear regression for torrential rain forecasting. In: Third International Joint Conference on Computational Science and Optimization (CSO), vol. 2, pp. 466–470 (2010)
3. Wu, J., Chen, E.: A novel nonparametric regression ensemble for rainfall forecasting using particle swarm optimization technique coupled with artificial neural network. In: 6th International Symposium on Neural Networks, pp. 49–58 (2009)
4. Lin, G.F., Chen, L.H.: Application of an artificial neural network to typhoon rainfall forecasting. Hydrol. Process. **19**(9), 1825–1837 (2005)
5. Hong, W.C.: Rainfall forecasting by technological machine learning models. Appl. Math. Comput. **200**(1), 41–57 (2008)

6. Lu, K., Wang, L.: A novel nonlinear combination model based on support vector machine for rainfall prediction. In: Fourth International Joint Conference on Computational Sciences and Optimization (CSO), pp. 1343–1346 (2011)
7. Mellit, A., Pavan, A.M., Benghanem, M.: Least squares support vector machine for short-term prediction of meteorological time series. Theor. Appl. Climatol. **111**(1–2), 297–307 (2013)
8. Rasel, R.I., Sultana, N., Meesad, P.: An efficient modeling approach for forecasting financial time series data using support vector regression and windowing operators. Int. J. Comput. Intell. Stud. **4**(2), 134–150 (2015)
9. Hasan, N., Nath, N.C., Rasel, R.I.: A support vector regression model for forecasting rainfall. In: 2nd International Conference on Electrical Information and Communication Technology (EICT), pp. 1–6 (2015)
10. Rasel, R.I., Sultana, N., Hasan, N.: Financial instability analysis using ANN and feature selection technique: application to stock market price prediction. In: International Conference on Innovations in Science, Engineering and Technology (ICISET-2016), pp. 1–4 (2016)
11. Gunn, S.R.: Support vector machines for classification and regression. Technical Reports, University of Southampton (1998)
12. Gershenson, C.: Artificial neural networks for beginners. Technical Reports, University of Sussex
13. Rectifier (neural networks). https://en.wikipedia.org/wiki/Rectifier_(neural_networks)

The Poet Identification Using Convolutional Neural Networks

Sajjaporn Waijanya(✉) and Nuttachot Promrit(✉)

Department of Computing, Faculty of Science, Silpakorn University, Nakhon Pathom, Thailand
{waijanya_s,promrit_n}@silpakorn.edu

Abstract. In this article, we propose to identify the amateur poet by using a Convolutional Neural Networks (CNNs). The poets were selected from the composing of Thai poem Klon-Suphap. The poems content are classified into 7 groups including with (1) royal, (2) parents and teachers, (3) fall in love, (4) broken, (5) festival, (6) advise, (7) depressed and there are poems of each poet in every groups. To identify the poet, input of model represented by the vector (Word2Vec) which had generated from Thai-Text corpus 5.9 Million words. The training data is Thai poem 900 units (baat) and testing data is Thai poem 96 units. CNNs showed the accuracy of 2 poets identification is 100%, 3 poets identification is 80.55%, 4 poets identification is 72.92% and 5 poets identification is 55.25%. In additional, we used 5 participants to read the poems of 2 poets and has predicted in testing data. The average of accuracy is 57.32% which less than the proposed model.

Keywords: Poet identification · Convolutional Neural Networks · Thai poem · Klon-Suphap · Klon-8 · Word2Vec

1 Introduction

Klon-suphap (Klon-8) has been featured extensively in Thailand since 200 years ago in the period of Thai greatest poet name "Sunthorn Phu". Sunthorn Phu was a honored by UNESCO [1] as a great world poet. The characteristic of Thai poem Klon-suphap is a complex prosody and different from other kind poems such as Sonnet and Hiku. To compose Thai poem Klon-suphap to have a very beautiful and correct prosody, the poets need to practice themselves and work very hard. Nowadays, the most publishing works of professional poets and amateurs are appeared in social network and website community. The amateur poets can learn and practice from many poet styles and they able to create their own stylistics.

We had started the project by developing the Artificial intelligence to compose Klon-suphap. Initially, machine ability had performed similar to amateur poets. It had created itself stylistic by learning real poets in social network and website. Therefore, identifying the poet was machine ability which able to proof machine understanding in stylistic of different among poets. Moreover, the poet identification is useful for plagiarism detection.

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_17

In this article, we had found that identifying the poet with a Convolutional Neural Networks (CNNs) has to use text feature extraction the same as with authorship identification of other contents. Thus, incomplete sentences of Thai poem had been occurred from the stringent prosody such as a number of syllables, rhyme structure and words rhymes. The example of Klon-suphap 1 unit (1 baat) in Thai is "น้ำต้นไม้ใบหญ้าปาขุนเขา นกกาเหว่าเสือสิงห์ลิงสมัน" the phonetic alphabet is "nam^4 ton^3maj^4 baj^1ja;^3 pa;^2 khun^5**khaw^5** - nok^4 ka;^1**waw^2** sv;a^1 sing^5 ling^1 sa^2man^5". The translation in English is "Water, trees, grass, forests, mountains – bird, koel, tiger, lion, monkey, schomburgk's deer". From the example, underline words are internal rhymes and bold words are external rhymes. All sample words had a part of speech "noun". The symbols ^1, ^2, ^3, ^4, ^5 are phonetic symbol of Tone in Thai syllable. Words pronounced in difference Tone can have difference meaning.

The feature extractions such as syntactic feature and entity feature need to have the complete sentence. These are the reasons that we had never used the syntactic feature and entity feature for poet identification. We used Word2Vec for instead which embedding feature extraction for Thai poem Klon-suphap.

We found some researcher works for poet identification [2] but never found any works using CNNs especially in Thai poet identification. Moreover, the poem set in other works are not strict in prosody if compare with Thai poem Klon-suphap. To measure our machine ability, the poem set in this article was classified to 7 groups include (1) royal, (2) parents and teachers, (3) fall in love, (4) broken, (5) festival, (6) advise, (7) depressed and the poets had their poems in every group. Machine should identify the poet even they had been composed the poems in the same groups.

2 Related Works

Author identification is one important task in Natural Language Processing (NLP). It had been used use in applied tasks such as authorship characterization, detecting plagiarism and etc. The key task of identifying authors is Feature Extraction. Text engineering for feature extraction including with syntactical parsing, entity extraction, statistical features and word embedding. The popular methodologies of author identification is lexical feature (TF-IDF). We found the Bangla poet identification by using lexical and style to identify the poets [2], each poet had different styles of poems. But all poets in this article have the similar styles. Text feature extraction technique such as TF-IDF and LDA must have a large number of training dataset to build bag of word. On the other hand, Word2Vec [3] has used bag of words from text corpus instant of training data. Normally, collecting Thai text corpus is easier than collecting Thai poem training set. For other text feature extractions such as syntactical parsing or named entity recognition, they are necessary for prose text and completed sentence but the characteristic of Thai poem is not similar. Thai poem has complex prosody and most of them have the incomplete sentence.

Word2Vec has used in this research instant of TF-IDF to embed the words in the poems. Word2Vec is used for learning vector representations of words. We applied the continuous skip-gram model in Word2Vec that used the current word to predict the

surrounding window. Then the position of words in vector space can be represented the semantic words.

Since, Thai Poem text feature extraction used Word2Vec represent word not whole sentence. The effortless way to extract feature be applied by Convolutional Neural Networks (CNNs) [4, 5]. CNNs is popular methodology in many domain [6–8]. We found some researcher use CNNs for author identification for source code [9]. However we have never found any researcher use CNNs for poet identification.

3 Model and Methodology

3.1 Process Overview

Thai language used to be input of process in this article. We had 2 main process groups including with (1) the preparation process and (2) word embedding and identification process. Process overview of Convolutional Neural Networks for Poet Identification Model has shown in Fig. 1. The preparation process has included Thai word segmentation and Thai poem transformation. According to the sentence in Thai language has not spaces or without any stop words then the word segmentation process is need. Thai word segmentation has separated words from 2 types of content Thai-poem and Thai-prose (news, encyclopedia, books, etc.). The output of Thai word segmentation is ThaiText corpus and only output of word segmentation from Thai-poem has been sent to Thai poem transformation process and created Thai poem training data and test data.

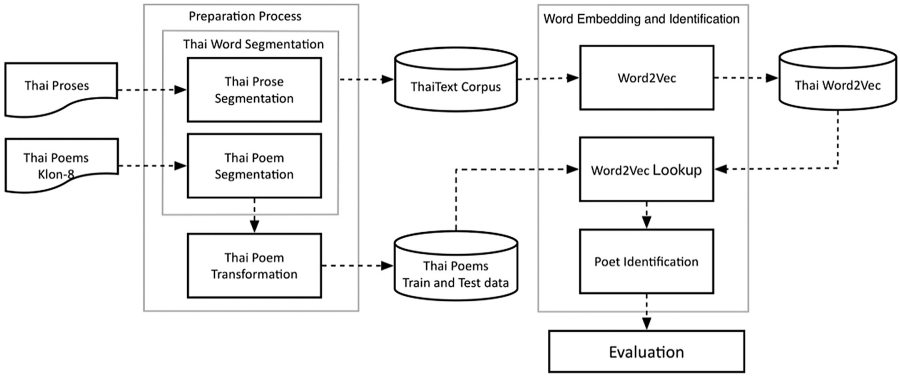


Fig. 1. Process overview of Convolutional Neural Networks for Poet Identification Model

At the word embedding and identification process, Word2Vec process used data from ThaiText corpus and the result of this process is Thai word embedding (Thai-Word2Vec), it was an input of Word2Vec lookup process. And the result of Word2Vec lookup is an input of poet identification process.

The last process, evaluation process has evaluated the performance of our model versus 5 participants.

3.2 Thai Poem (Input)

The objective of this research, attempts to identify the Thai poet who compose Thai poem Klon-suphap. Understanding about Klon-suphap poem, the rhyme and prosody term has shown in Fig. 2, its prosody is number of syllable in line. There are only 7 to 9 syllables allowed in each line and it will not greater than 9 or less than 7 syllables, an error is implicated in the length of the line. Moreover, Thai poem Klon-suphap has rule of syllables relation.

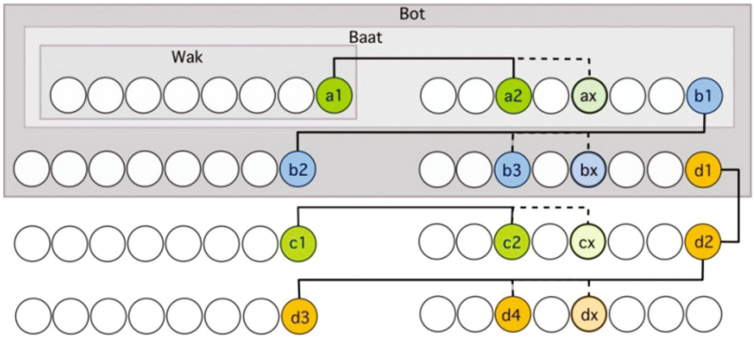


Fig. 2. Thai poem Klon-suphap structure and prosody

The syllables relation of Klon-suphap means syllables in rhyme positions must having same vowel sound and same spelling-sound such as "kh**aw^5**" (เขา: he, she, mountain) and "r**aw^5**" (เรา: we) but its phoneme must not be duplicated.

For prose writing, Syntactic will be completed by grammar. But each "Wak", "Baat" and "Bot" in Thai poem can be written without syntactic grammar. The poets sometimes starting their poem by Verb. The example of "Wak" such as "ส่งความรักฝากไปกับสายฝน" translates word by word to be "send love deliver by the rain". It was not right grammar and incomplete in sentence structure, but in Thai language, reader can understand the meaning of this "Wak", the meaning as "Somebody send his (or her) love to another one, love will be delivered by the rain".

3.3 Thai Word Segmentation

In this work used Thai word segmentation API [10] to cut words from Thai prose content in preparation process to be ThaiText corpus. We also cut words from Thai Poem both training set and test set by using longest matching with dictionary-base technique.

3.4 Word2Vec

Preparing Word Vectors to be input data of Convolutional Neural Network model, we have created Word2Vec by defining size of word vectors 200 dimension and train Word2Vec by skip-gram model by using ThaiText corpus 5.9 Million words from 5 online resources including with (1) "BEST I Corpus" by NECTEC, (2) Contemporary Poets Society: www.kawethai.com, (3) www.wannakadee.com, (4) www.thaipoem.com and (5) www.aromklon.com. The number of bag of words is 101,432.

The result of Word2Vec from ThaiText corpus in Fig. 3 shows a vector that represents semantic attributes of the words with t-SNE [11]. The example of words with similar meaning in Thai are "ไม่" = "Not" and "มิ" = "Not". The example of words with semantic relation in Thai are "ศาล" = "court", "สิทธิ" = claim, "กฎหมาย" = laws, "รัฐธรรมนูญ" = constitution.

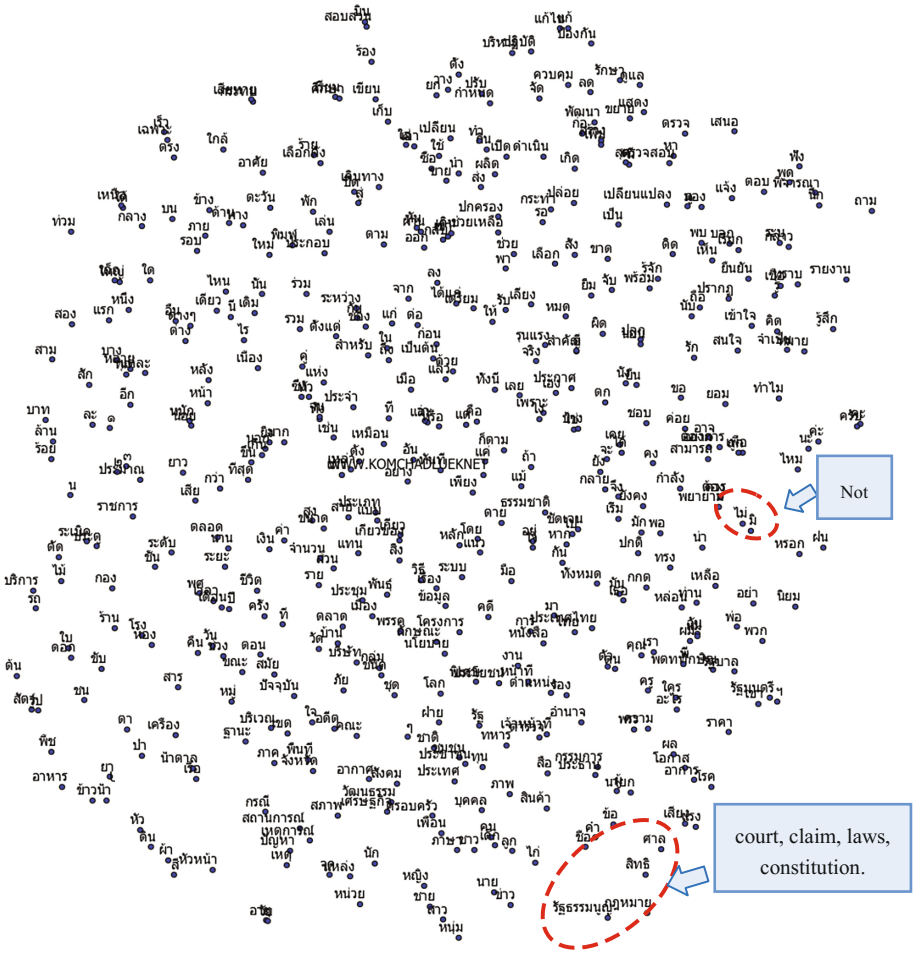


Fig. 3. Word2Vec from ThaiText corpus

3.5 Convolutional Neural Networks

One of the characteristics of Convolutional Neural Networks (CNNs) is feature extraction before feeding into fully connected network. Therefore applying CNNs is the effortless way to extract feature.

Our model in Fig. 4 shows the process of CNNs model. Thai Poem Word Embedding was an input. In each poem, word is represented by each vector. Size of vector is k-dimension and the number of words are n. To identify poet, we defined n is 18 words. At the step of the matrix preparation, if number of word has not full then we will pad by zero vectors. Then the shape of input matrix is $n \times k$.

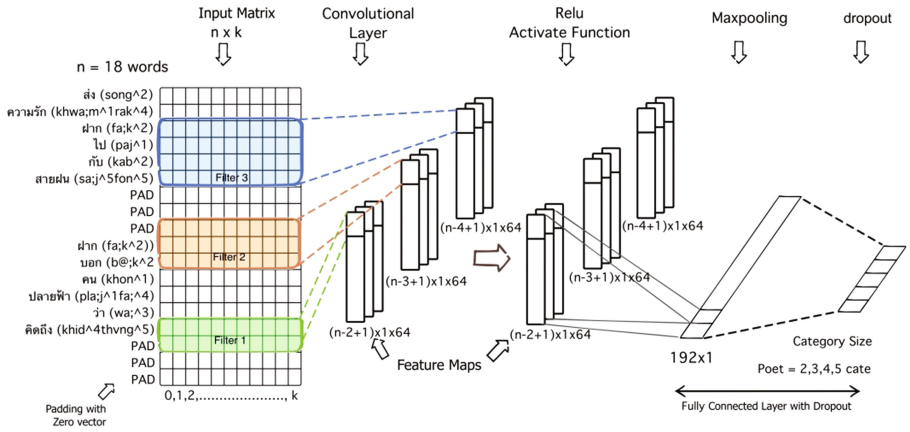


Fig. 4. Convolutional Neural Network Model

Convolutional Layer has input shape is $n \times k$ then transforms to be 3 shapes of feature maps. The filter had been slide down on input matrix by 1 word for create the feature map. The shape of filter is $ROW \times k$, when ROW including with 2, 3 and 4. Then the shape of feature maps is $W_1 \times H_1 \times D_1$. The W_1 can calculate by (1) H_1 is 1 and D_1 is 64.

$$W_1 = (n - ROW + 1) \tag{1}$$

Next step, we created new feature maps by adding a previous feature maps with bias and sent to Relu activate function.

After new feature maps layer, we created the matrix size $\mathbf{192 \times 1}$ by using **1_maxpooling** method. It was selected maximum value in each feature map and concatenates each other. We used dropout technique to solve the over fitting problem by defining dropout rate is 0.5 while training the model. Finally, after dropout in fully connected layer, the max value was selected to present the poet's identification.

4 Experimental and Result

In this article, we experimented to measure the performance of model by identifying two to five poets. Table 1 shows the information of training data set, batch size, validation data set for identifying two, three, four and five poets respectively.

Table 1. The information of experiment

Number of poet	Training dataset (baat)	Batch size (baat)	Validation dataset (baat)
2	216	50	24
3	324	75	36
4	444	100	48
5	520	100	60

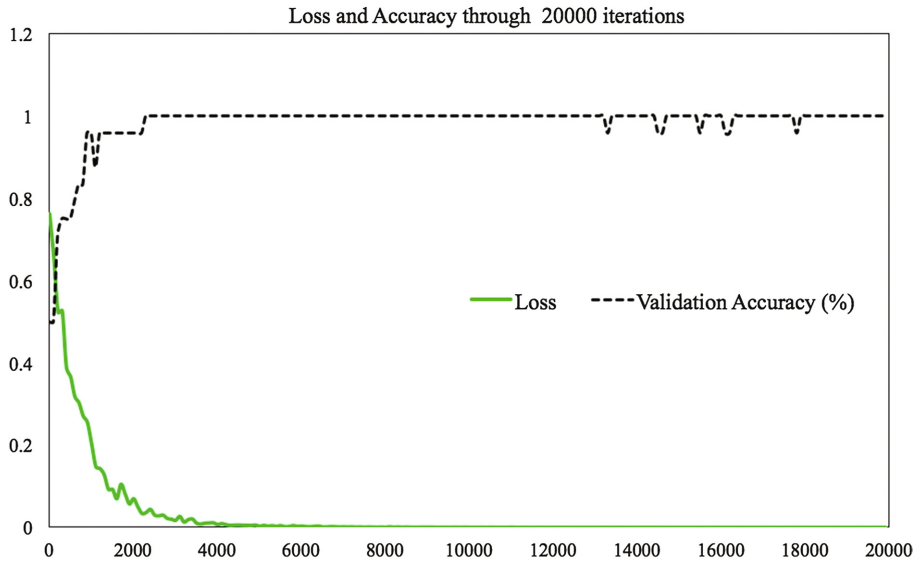


Fig. 5. The loss and accuracy of CNNs training

The experiment, we had defined the parameter of this model, we used *k*-dimension = 200 to generate the Word2vec. We had defined 20,000th iterations for training. Every training iteration we took the training poem dataset of each poet. The training dataset including from every category equally. Additional, we used the data for validation in each 100th training iterations. The loss and accuracy of CNNs training was showed in Fig. 5. The overall accuracy of 100% was achieved, after 2,200th training iterations. Moreover, the loss value has decreased obviously near 4,000th training iterations. Therefore, we can use the model with training less than 20,000th iterations.

We used the model to predict the poet identification. The accuracy of two, three, four and five poets identification was showed in Table 2.

Table 2. The accuracy of poets identification.

Number of poet	Accuracy (%)
2	100
3	80.55
4	72.92
5	57.32

In additional, we had compared the ability of model with 5 participants. They are undergraduate students in Computer Science Program and Information Technology Program. 3 of them known the prosody of Klon-suphap and 1 participant can compose the Klon-suphap poem. Each participant read the poems 216 baat from 2 poets reveal the poet names. These poems were training poem dataset of the model. After read the poems from training dataset, they had experiment by using 24 baat. The experimental result of poet identification by participants has shown in Table 3.

Table 3. The experimental result of poet identification by participants

Participants	Accuracy (%)
1	66.66
2	50
3	54.16
4	66.66
5	48.52
Average	57.20

The identification results by the model with k-dimension = 200 show the accuracy 100% of two poets identification. The accuracy had decreased when we added the number of poets. The accuracy had decreased to 57.32% when the model identifies five poets. But in case of identification 5 participants, the average of accuracy was only 57.20%. This accuracy value is similar to wildcat values. Therefore the CNNs model in this work is able obviously to identify better than the participants. Moreover the experimental result can show the high ability of CNNs model to finding the pattern from Klon-suphap whenever we used the word embedding (Word2Vec) to be input of model.

5 Conclusion and Future Work

In this article, we propose CNNs model and adjusted parameters for identifying Thai poet who's compose Klon-suphap. This is first research which identify poet by using CNNs. The CNNs has input term as the matrix. For the input matrix, the value in each row is vector that represent Thai words (Word2Vec Embedding). The Word2Vec had built amount of Thai Text corpus 5.9 million words. It has bag of words 101,432 words that enough for building input matrix.

All of Thai poem Klon-suphap 520 baat from 7 categories by 5 poets will be represented as the vector and use to be training dataset. Although we have small number of poem training dataset in many categories, but our model able to identify the poet and it shows the good performance. Moreover when we compare the accuracy with the participants whose read the poems of 2 poets, the accuracy by average of 5 participants similar wildcat value. The CNNs has high ability more than participants obviously. By this reason it impossible to develop machine that able to compose Thai poem Klon-suphap with its style. Next, we will apply CNNs and Word2Vec to compose Thai poem Klon-suphap in the future.

References

1. Thailand's Shakespeare? Sunthorn Phu|ThingsAsian. http://thingsasian.com/story/thailands-shakespeare-sunthorn-phu
2. Das, A., Gambäck, B.: Poetic machine: computational creativity for automatic poetry generation in Bengali. In: 5th International Conference on Computational Creativity, ICCC (2014)
3. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv:13013781 Cs. (2013)
4. Le, Q.V.: A Tutorial on Deep Learning. Part 1: Nonlinear Classifiers and the Backpropagation Algorithm (2015). http://robotics.stanford.edu/∼quocle/tutorial1.pdf.
5. Le, Q.V.: A Tutorial on Deep Learning. Part 2: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks (2015). http://robotics.stanford.edu/∼quocle/tutorial2.pdf.
6. Rios, A., Kavuluru, R.: Convolutional neural networks for biomedical text classification: application in indexing biomedical articles. In: Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics, pp. 258–267. ACM, New York (2015)
7. Zhang, Y., Wallace, B.: A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv:151003820 Cs. (2015)
8. Weston, J., Chopra, S., Adams, K.: #TAGSPACE: semantic embeddings from hashtags. Presented at the the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar (2014)
9. Bandara, U., Wijayarathna, G.: Deep neural networks for source code author identification. In: Lee, M., Hirose, A., Hou, Z.-G., Kil, R.M. (eds.) Neural Information Processing, pp. 368–375. Springer, Heidelberg (2013)
10. Veer Sattayamas. https://github.com/veer66/PhlongTaIam
11. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**, 2579–2605 (2008)

Complex Networks and Systems

An Enhanced Deep Convolutional Encoder-Decoder Network for Road Segmentation on Aerial Imagery

Teerapong Panboonyuen[1(✉)], Peerapon Vateekul[1(✉)], Kulsawasd Jitkajornwanich[2], and Siam Lawawirojwong[3]

[1] Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand
teerapong.pan@student.chula.ac.th, peerapon.v@chula.ac.th
[2] Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
kulsawasd.ji@kmitl.ac.th
[3] Geo-Informatics and Space Technology Development Agency (Public Organization), Bangkok, Thailand
siam@gistda.or.th

Abstract. Object classification from images is among the many practical examples where deep learning algorithms have successfully been applied. In this paper, we present an improved deep convolutional encoder-decoder network (DCED) for segmenting road objects from aerial images. Several aspects of the proposed method are enhanced, incl. incorporation of ELU (exponential linear unit)—as opposed to ReLU (rectified linear unit) that typically outperforms ELU in most object classification cases; amplification of datasets by adding incrementally-rotated images with eight different angles in the training corpus (this eliminates the limitation that the number of training aerial images is usually limited), thus the number of training datasets is increased by eight times; and lastly, adoption of landscape metrics to further improve the overall quality of results by removing false road objects. The most recent DCED approach for object segmentation, namely SegNet, is used as one of the benchmarks in evaluating our method. The experiments were conducted on a well-known aerial imagery, Massachusetts roads dataset (Mass. Roads), which is publicly available. The results showed that our method outperforms all of the baselines in terms of precision, recall, and F1 scores.

Keywords: Deep convolutional neural network · Remote sensing · Image processing · Deep learning · Road segmentation

1 Introduction

Several approaches for extracting road objects from aerial images are proposed, but there is still demand in achieving higher accuracy of the extracted road result sets through which many applications can benefit from. Examples include urban planning,

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_18

map updates, route optimization and navigation. The road extraction research has primarily been based on unsupervised learning, such as k-means [8], graph cut [18], homogram thresholding [11], and global optimization techniques [18]. Nonetheless, these unsupervised learning have one limitation in common; most of them are color-sensitive. That is, the segmentation algorithms will not perform well if the road colors presented in the suburban aerial images contain more than one color (e.g., yellowish brown roads in the countryside regions and cement-grayed roads in the suburban regions). This, in fact, has become a motivation of this work.

Deep learning, a large convolutional neural network whose performance can be scaled depending on size of training data, model complexity as well as processing power, has shown significant improvements in object segmentation from images as seen in many of the recent works [2–7, 9, 13–16]. Unlike unsupervised learning, more than one features—other than color—can be extracted: line, shape, and texture, among others. The traditional deep learning methods such as deep convolutional neural networks (CNN) [1, 3], deep deconvolutional neural networks (DeCNN) [5], recurrent neural network, namely reSeg [12], and fully convolutional networks [4]; however, are all suffering from the accuracy performance issues.

A deep convolutional encoder-decoder (DCED) architecture, one of the most efficient newly developed neural networks, has been proposed for object segmentation and given good performance in the experiments tested on PASCAL VOC 2012 data—a well-known benchmark dataset for image segmentation research [6, 9, 21]. Two main components of DCED are an encoder network and a decoder network. The encoder network consists of 13 convolutional layers corresponding to the first 13 convolutional layers in the VGG16 network; the remaining layers are removed from the fully-connected layers. So, the revised encoder network is significantly smaller, which in turn makes it easier to train compared to many other architectures. Rectified linear unit (ReLU) is used in this architecture. The decoder network converts the encoder feature maps to full input resolution feature maps for pixel-wise classification. We use stochastic gradient descent (SGD) with a fixed learning rate of 0.001 and momentum of 0.9 to train all the variants in this architecture.

In this paper, we present an improved deep convolutional encoder-decoder network (DCED) for segmenting road objects from aerial images. Several aspects of the proposed method are enhanced, incl. incorporation of ELU (exponential linear unit)—as opposed to ReLU that typically outperforms ELU in most object classification cases; amplification of datasets by adding incrementally-rotated images with eight different angles in the training corpus (this eliminates the limitation that the number of training aerial images is usually limited), thus the number of training datasets is increased by eight times; and lastly, adoption of landscape metrics to further improve the overall quality of results by removing false road objects. The most recent DCED approach for object segmentation, namely SegNet, is used as one of the benchmarks in evaluating our method. The experiments were conducted on a well-known aerial imagery, Massachusetts roads dataset (Mass. Roads), which is publicly available. The results showed that our method outperforms all of the baselines in terms of precision, recall, and F1 scores.

The paper is organized as follows. Related work is discussed in Sect. 2. Section 3 describes our methodology. Performance evaluations and experiments are presented in Sects. 4 and 5, respectively. We conclude our work and discuss future work in Sect. 6.

2 Related Work

Deep learning has increasingly become a promising tool for accelerating image recognition process with high accuracy results [4, 6, 10]. This related work is divided into three subsections: we first discuss deep learning concepts for semantic segmentation, followed by a set of road object segmentation techniques using deep learning, and finally activation functions of deep learning are discussed.

2.1 Deep Learning for Semantic Segmentation

Semantic segmentation algorithms are often formulated to solve structured pixel-wise labeling problems based on convolutional neural network (CNN), the state-of-the-art supervised learning algorithms in modeling and extracting latent features hierarchies. Noh et al. [5] proposed a novel semantic segmentation technique utilizing a deconvolutional neural network (DeCNN) and the top layer from CNN adopted from VGG16 [20]. DeCNN structure is composed of upsampling layers and deconvolution layers, describing pixel-wise class labels and predicting segmentation masks, respectively. Their proposed deep learning methods yield high performance in PASCAL VOC 2012 dataset [21], with the 72.5% accuracy in the best case scenario (the highest accuracy—as of the time of writing this paper—compared to other methods that were trained without requiring additional or external data). Long et al. [4] proposed an adapted contemporary classification networks incorporating Alex, VGG and GoogLe networks into fully CNN. In this method, some of the pooling layers were skipped: layer 3 (FCN-8s), layer 4 (FCN-16s), and layer 5 (FCN-32s). The skip architecture reduces the potential over-fitting problem and has showed improvements in performance, ranging from 20% to 62.2% in the experiments tested on PASCAL VOC 2012 data. Ronneberger et al. [16] proposed U-Net, a CNN for biomedical image segmentation. The architecture consists of a contracting path and a symmetric expanding path that capture context and consequently, enable precise localization. The proposed network claimed to be capable to learn despite the limited number of training images, and performed better than the prior best method (a sliding-window CNN) on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks.

In this work, VGG16 is selected as our baseline architecture since it is the most popular architecture used in various networks for object recognition. Furthermore, we will investigate the effect of the skipped layer technique, especially FCN-8s, since it is the top-ranking architecture as shown in Long et al. [4].

2.2 Deep Learning for Road Segmentation

There have been many approaches in road network extraction from very-high-resolution (VHR) aerial and satellite imagery literature. Wand et al. [1] proposed a CNN- and FSM (finite state machine)-based framework to extract road networks from aerial and satellite images. CNN recognizes patterns from a sophisticated and arbitrary environment while FSM translates the recognized patterns to states such that their tracking behaviors can be captured. The results showed that their approach is more accurate compared to the traditional methods. The extension of the method for automatic road point initialization was left for future work. CNN for multiple object extraction from aerial imagery was proposed in [3] by Saito et al. Both features (extractors and classifiers) of CNN were automated in that a new technique to train a single CNN for extracting multiple kinds of objects simultaneously was developed. Two objects were extracted: buildings and roads, thus a label image consists of three channels: buildings, roads, and background. Finally, the results showed that the proposed technique not only improved the prediction performance but also outperformed the cutting-edge method tested on a publicly available aerial imagery dataset. Muruganandham et al. [2] designed an automated framework to extract semantic maps of roads and highways, so the urban growth of cities from satellite images can be tracked. They used VGG16 model—a simplistic architecture with homogeneous 3×3 convolution kernels and 2×2 max pooling throughout the pipeline—as a baseline for fixed feature extractor. The experimental results showed that their proposed technique for the prediction performance was improved with F1 scores of 0.76 on the Mass. Roads dataset.

2.3 Activation Functions in Deep Learning

While the most popular activation function for neural networks is the rectified linear unit (ReLU), Clevert et al. [10] have just proposed exponential linear unit (ELU), which can speed up the learning process in CNN and therefore, lead to higher classification accuracies as well as overcome the previously unsolvable problem, i.e., vanishing gradient problem. Comparing to other methods with different activation functions, ELU has greatly improved many of the learning characteristics. In the experiments, ELUs enable fast learning as well as more effective generalization performance than ones of ReLUs and LReLUs (leaky rectified linear unit) on the networks with five layers or more. In ImageNet, ELU networks substantially increase the learning time compared to ReLU networks with the identical architecture; less than 10% classification error was presented for a single crop, model network.

3 Methodology

We proposed an adapted, improved DCED network (or SegNet) to efficiently segment road objects from aerial images. Three aspects of the proposed method are enhanced: data amplification, modification of DCED architecture, and adoption of landscape metrics.

3.1 Dataset Amplification

We increase the size of our datasets (Mass. Roads, made publicly available by Mnih [7] on website: http://www.cs.toronto.edu/∼vmnih/data/) to improve efficiency of the method by rotating them incrementally with eight different angles (as shown in Fig. 1). All images are standardized and cropped into 1,500 × 1,500 pixels with a resolution of 1 m^2/pixel. The datasets consist of 1,108 training images, 49 test images, and 14 validation images. The original training images were further extended to 8,864 training images.

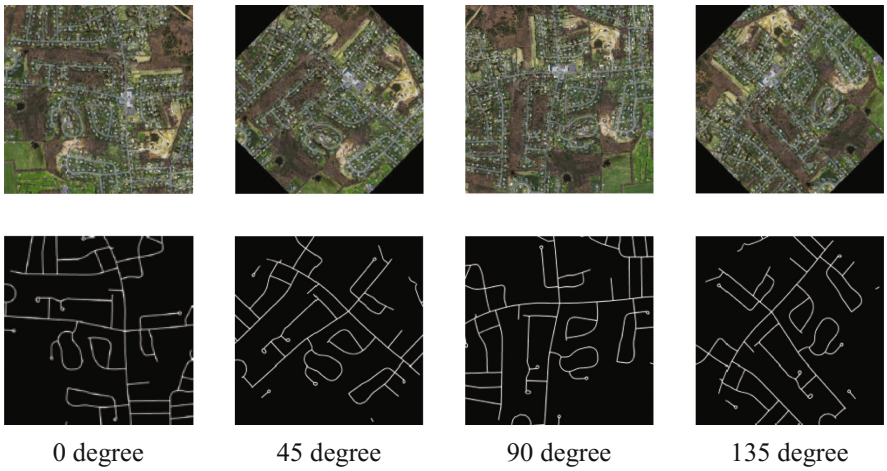


Fig. 1. Our sample aerial image and target road map in four (out of the eight) different angles

3.2 Modification of DCED Architecture

SegNet, one of the deep convolutional encoder-decoder architectures, consists of two main networks: encoder and decoder, and some outer layers. The two outer layers of the decoder network are responsible for feature extraction task, the results of which are transmitted to the next layer adjacent to the last layer of the decoder network. This layer is responsible for pixel-wise classification (determining which pixel belongs to which class). There is no fully connected layer in between feature extraction layers. In the up sampling layer of decoder, pool indices from encoder are distributed to the decoder where kernel will be trained in each epoch (training round) at convolution layer. In the last layer (classification), softmax is used as a classifier for pixel-wise classification.

The encoder network consists of convolution layer and pooling layer. A technique, called batch normalization (proposed by Ioffe and Szegedy [19]), is used to speed up the learning process of the CNN by reducing internal covariate shift. In the encoder network, the number of layers are reduced to 13 layers (VGG16) by removing the last three layers (fully connected layers) [9] due to the following two reasons: to maintain the high-resolution feature maps in the encoder network, and to minimize the countless

number of parameters from 134 million features to 14.7 million features compared to the traditional deep learning networks such as CNN [4] and DeCNN [5], where the fully connected layer remains intact. In the activation function of feature extraction, ReLU, max-pooling, and 7 × 7 kernel are used in both encoder and decoder networks. For training images, three-channel images (r/g/b) are used.

Exponential Linear Unit (ELU) was introduced in [10], which can speed up learning in deep neural networks, offer higher classification accuracies, and give better generalization performance than ReLUs and LReLUs on networks. In SegNet architecture, to do optimization for training networks, stochastic gradient descent (SGD) with a fixed learning rate of 0.1 and momentum of 0.9 are used. In each training round (epoch), a mini-batch (a set of 12 images) is chosen such that each image is used once. The model with the best performance on the validation dataset in each epoch will be selected.

Our architecture (Fig. 2) is adapted from SegNet, consisting of two main networks responsible for feature extraction. In each network, there are 13 layers with the last layer being the classification based on softmax supporting pixel-wise classification. In our work, an activation function called ELU is used—as opposed to ReLU—based on its performances. For the network training optimization, stochastic gradient descent (SGD) is used and configured with a fixed learning rate of 0.001 and momentum of 0.9 to delay the convergence time and so, can avoid local optimization trap.

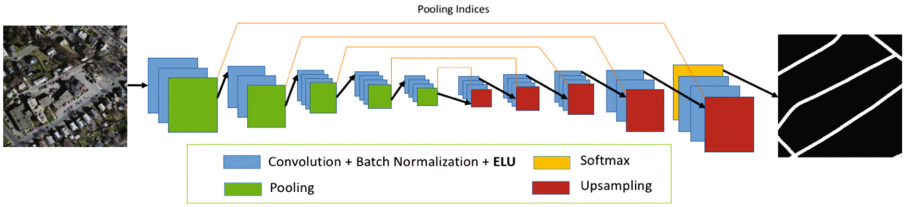


Fig. 2. Our adapted SegNet architecture

3.3 Landscape Metrics

In this paper, *shape metrics* (one of the landscape metrics for measuring spatial object complexity) is used [17]. Geometrical characteristics of the roads are captured and differentiated from other spatial objects in the given image. Other geometry metrics can also be used such as rectangular degree, aspect ratio, etc. More information on other landscape metrics can be found in [11, 17].

4 Performance Evaluations

The road extraction task can be considered as binary classification, where road pixels are positives and the remaining non-road pixels are negatives. Let TP denote the number of true positives (the number of correctly classified road pixels), TN denote the

number of true negatives (the number of correctly classified non-road pixels), FP denote the number of false positives (the number of mistakenly classified road pixels), and FN denote the number of false negatives (the number of mistakenly classified non-road pixels).

The performance measures used are precision, recall, and F1 as shown in equations (Eqs. 1–3). Precision is the percentage of correctly classified road pixels among all predicted pixels by the classifier. Recall is the percentage of correctly classified road pixels among all actual road pixels. F1 is a combination of precision and recall.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

5 Experiments

The proposed deep learning network is based on SegNet with three improvements: (i) employ ELU as activation function, (ii) increase training data by adding rotated images, (iii) apply landscape metrics to filter false detected road pixels. Table 1 illustrates all variations of the presented deep learning method. The implementation is based on a deep learning framework, called "Lasagne".

Table 1. Variations of the proposed deep learning methods

Abbreviation	Description
ELU-SegNet	SegNet + **ELU activation**
ELU-SegNet-**R**	SegNet + ELU activation + **Rotated Images**
ELU-SegNet-R**L**	SegNet + ELU activation + Rotated Images + **Landscape Metrics**; (all modules)

The experiments were conducted on a standard benchmark, "Massachusetts roads dataset" (Mass. Roads) and compared to four baselines: basic-model (CNN), FCN-no-skip, FCN-8s, and SegNet, in terms of precision, recall, and F1. All experiments are performed on a server with Intel® Core™ i5-4590S Processor (6 M Cache, up to 3.70 GHz), 8 GB of memory, and Nvidia GeForce GTX 960 (4 GB). The training procedure took approximately 32 h for the original training datasets and 48 h for the amplified training datasets, and finished after 200 epochs. In each epoch, 576 s were used for the original training datasets and 864 s were used for the amplified training datasets.

Table 2 illustrates the results of the proposed methods and existing techniques on the Mass. Roads dataset. ELU-Segnet-RL gives the best performance of all the methods with more than 0.8 in all measures: 0.854, 0.861, and 0.857; in terms of precision, recall, and F1; respectively. Particularly, its F1 scores outperform all the prior attempts. Comparing to the original SegNet, the result shows that each proposed module can really improve the F1 performance. The F1 scores are improved by 2% when adding ELU, by 4.4% when also adding rotated images, and by 8.9% when combining all modules.

Table 2. A comparison between our proposed methods and baselines in terms of precision, recall, and F1

	Model	Precision	Recall	F1
Baseline	Basic-model [2]	0.657	0.657	0.657
	FCN-no-skip [2]	0.742	0.742	0.742
	FCN-8s [2]	0.762	0.762	0.762
	SegNet [6]	0.773	0.765	0.768
Our proposed	ELU-SegNet	0.852	0.733	0.788
	ELU-SegNet-R	0.780	0.847	0.812
	ELU-SegNet-RL	**0.854**	**0.861**	**0.857**

There will be further discussions on two improvement aspects on: (i) the deep learning process and (ii) the deep learning output.

5.1 Discussion on Enhanced Deep Learning Framework

To enhance the framework, there are two proposed strategies. First, the activation function is replaced by ELU due to its outstanding overall performance as reported in [10]. Second, the number of training data is increased by adding more rotated images, thus the network can learn more road patterns from various angles.

For the effect of ELU, Table 2 shows that the precision of ELU-SegNet is higher than that of the original SegNet by 7.9%—without losing recall. This can imply that ELU is more robust than ReLU to detect road pixels. Figure 3 shows the results of extracted roads in five aerial images using different methods compared to input and target images. When comparing the results between the original SegNet method (Fig. 3c) and the ELU-SegNet method (Fig. 3d), it clearly shows that ELU can improve the network performance in detecting more roads, especially in the first and second images in that the extracted roads are filled and thickened.

For the effect of adding rotated images, Table 2 shows that the recall of ELU-SegNet-R is higher than that of ELU-SegNet by 11.4%, meaning that it can detect more patterns of the roads. However, the precision is affected and dropped by 7.2% since too many pixels are labeled (both road and non-road pixels) and classified as road pixels. In Fig. 3(d and e), a comparison between the extracted images without and with adding rotated images, respectively in the training process are demonstrated. In the

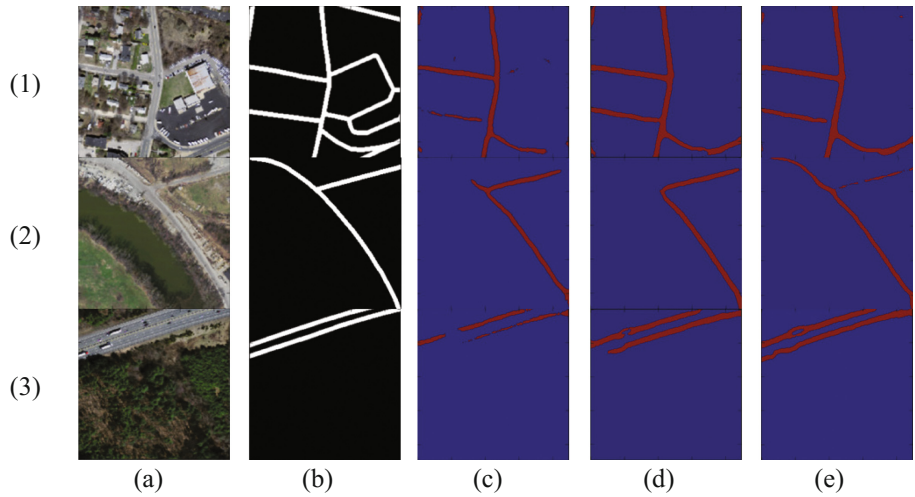


Fig. 3. Road extraction of two aerial images (each row) among different methods in Table 2: (a) input image, (b) target road map, (c) extract roads using SegNet, (d) extracted roads using ELU-Segnet, and (e) extracted road using ELU-SegNet-R

second image, there are more lines of extracted roads in the last column than the prior column. To filter the excessive extracted roads, the landscape metrics are applied to the proposed framework; this will be discussed in the next section.

5.2 Discussion on Landscape Metrics

The landscape metrics are applied to our framework in order to remove all inaccurately extracted roads (false positives: FP), considered as a negative effect of the rotated image strategy as discussed in the previous section. Table 2 shows that the precision of the network is increased by 7.4% by applying the landscape metrics filtering. This shows that the FP issue has been resolved. Figure 4 aims to illustrate that the excessive

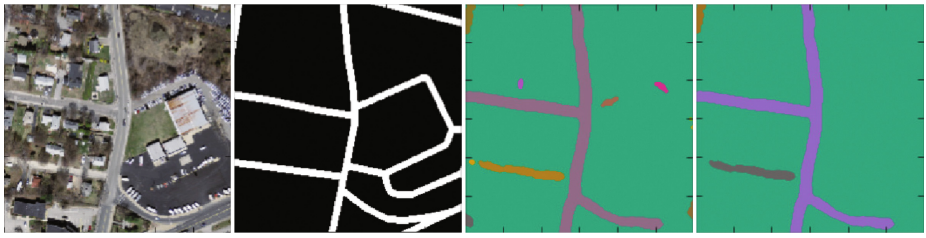


Fig. 4. Road extraction of an aerial image: (a) input image, (b) target road map, (c) extracted roads, and (d) extracted roads with removing noises, which are objects whose shape index is less than 1.25 (this parameter can be obtained from an experiment on a validation data set)

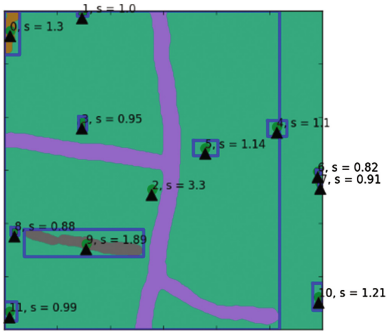


Fig. 5. Illustration of shape index scores on the extracted objects in Fig. 4(c)

roads from the learning model (Fig. 4c) are cleansed by adding the filtering strategy (Fig. 4d). Figure 5 demonstrates the shape index scores of all extracted objects on Fig. 4(c). All rounded objects, which are non-road, have low shape index scores and subsequently were filtered out.

6 Conclusion

In this paper, we present a novel deep learning network framework to extract road objects from aerial images. The network is based on Deep Convolutional Encoder-Decoder Network (DCED), called "SegNet." To improve the network's precision, we incorporate the recent activation function, called Exponential Linear Unit (ELU), into our proposed method. The proposed model is also improved to detect more road patterns by adding eight different rotated images. Excessive detected roads are further be eliminated by applying landscape metrics thresholding. The experiments were conducted on Massacusetts roads dataset and compared to the existing road extraction techniques. The results show that the enhanced SegNet outperforms the original one—10.6% for F1—as well as all other baselines.

In future work, more choices of image segmentation techniques, optimization techniques and/or other activation functions will be investigated and compared to obtain the best DCED-based framework for semantic road segmentation.

Acknowledgements. T. Panboonyuen thanks the scholarship from Chulalongkorn University to commemorate the 72nd Anniversary of H.M. King Bhumibala Aduladeja. He also thanks Dr. Panu Srestasathiern from GISTDA for his invaluable guidance.

References

1. Wang, J., Song, J., Chen, M., Yang, Z.: Road network extraction: a neural-dynamic framework based on deep learning and a finite state machine. Int. J. Remote Sens. **36**, 3144–3169 (2015)

2. Muruganandham, S.: Semantic segmentation of satellite images using deep learning. M.S. thesis, Czech Technical University in Prague and Luleå University of Technology (2016)
3. Saito, S., Yamashita, T., Aoki, Y.: Multiple object extraction from aerial imagery with convolutional neural networks. J. Imaging Sci. Technol. **60**(1), 1–9 (2016)
4. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
5. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: International Conference on Computer Vision (2015)
6. Badrinarayanan, V., Handa, A., Cipolla, R.: Segnet: a deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. arXiv:1505.07293v1 (2015)
7. Mnih, V.: Machine learning for aerial image labeling. Ph.D. thesis, University of Toronto (2013)
8. Maurya, R., Gupta, P.R., Shukla, A.S.: Road extraction using k-means clustering and morphological operations. In: International Conference on Image Information Processing, pp. 708–714 (2011)
9. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: a deep convolutional encoder-decoder architecture for image segmentation. arXiv:1511.00561v3 (2016)
10. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (ELUs). In: 4th International Conference on Learning Representations (2016)
11. Xu, G., Zhang, D., Liu, X.: Road extraction in high resolution images from google earth. In: 7th International Conference on Information and Communication Systems, pp. 556–560 (2009)
12. Visin, F., Ciccone, M., Romero, A.: Reseg: a recurrent neural network-based model for semantic segmentation. arXiv:1511.07053 (2015)
13. Volpi, M., Ferrari, V.: Semantic segmentation of urban scenes by learning local class interactions. In: Computer Vision and Pattern Recognition Workshops, pp. 1–9 (2015)
14. Liu, J., Liu, B., Lu, H.: Detection guided deconvolutional network for hierarchical feature learning. Pattern Recogn. **48**(8), 2645–2655 (2015)
15. Hong, S., Noh, H., Han, B.: Decoupled deep neural network for semi-supervised semantic segmentation. In: Conference on Neural Information Processing Systems, pp. 1495–1503 (2015)
16. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical. arXiv:1505.04597v1 (2015)
17. Mcgarigal, K.: Landscape metrics for categorical map patterns. McGarigal (Lecture notes), vol. 2001, Chap 5, pp. 1–77 (2001)
18. Poullis, C.: Tensor-cuts: a simultaneous multi-type feature extractor and classifier and its application to road extraction from satellite images. ISPRS J. Photogramm. Remote Sens. **95**, 93–108 (2014)
19. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning, pp. 448–456 (2015)
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations (2015)
21. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vision **111**(1), 98–136 (2015)

Pseudo-ranging Based on Round-Trip Time of Bluetooth Low Energy Beacons

Supatana Hengyotmark[1(✉)], Teerayut Horanont[1], Kamol Kaemarungsi[2], and Kazuhiko Fukawa[3]

[1] School of ICT, Sirindhorn International Institute of Technology, Thammasat University, Khlong Luang, Thailand
h.supatana@gmail.com, teerayut@siit.tu.ac.th
[2] National Electronics and Computer Technology Center, Pathumthani, Thailand
kamol.kaemarungsi@nectec.or.th
[3] Tokyo Institute of Technology, Ookayama Campus, Tokyo, Japan
fukawa@radio.ce.titech.ac.jp

Abstract. This paper presents a utilization of Bluetooth Low Energy (BLE) for a pseudo-ranging determination in indoor positioning systems. Most of BLE-based ranging techniques rely on Received Signal Strength Indicator (RSSI) of the broadcasting beacons, which has a shortcoming as it is highly sensitive to the surroundings or the orientation of the device. Therefore, this paper proposes the two-way Time-of-Flight (ToF) approach for ranging of BLE beacons. The Round-Trip Time (RTT) of the beacons has been measured using a CPU clock cycle generated by an external Temperature Compensated Crystal Oscillator (TCXO), and the pseudo-ranging calculation has been performed afterwards. The study results indicate that the accuracy can be improved by considering the RTT from the advertising channel that gives the best RSSI. The equation representing the relationship between the RTT and distance has been derived.

Keywords: Bluetooth low energy · Pseudo-ranging · Round-trip time

1 Introduction

The Global Navigation Satellite System (GNSS) has proven itself to provide a precise positioning accuracy for many outdoor applications. However, its signal cannot penetrate into the building. This leads many researchers to develop indoor positioningsystems by utilizing various mediums such as sound, light or radio wave to achieve pseudo-ranging process.

Bluetooth Low Energy (BLE) is one of the most famous mediums used for indoor positioning systems. It has been part of Bluetooth Core Specifications [1] since the release of version 4.0, which mostly aims for sensor devices that periodically advertise data beacons with a very low power consumption. BLE can be found in almost all of smart phones nowadays. It has been widely used in indoor positioning due to low cost, less effort for implementation, and low power consumption. But because of lacking precise timesynchronization in BLE, it is difficult to determine pseudo-ranging by Time-of-Flight

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_19

(ToF) approach such as Time-of-Arrival (ToA) or Time-Difference-of-Arrival (TDoA). Moreover, Angle-of-Arrival (AoA) approach requires directional or antenna array, which is rarely found in BLE device [2]. Therefore, almost all researches using BLE for ranging have been paid attention to Received Signal Strength Indicator (RSSI) of the beacons.

2 Related Works

BLE-based positioning system mostly relies on the RSSI. Pseudo-ranging is then obtained fromthe relationship between the RSSI and displacement that can be created from radio propagation lost equation or experimentation [3]. Alternatively, a pre-defined mapping or fingerprintingof RSSI [4, 5] can also be used to determine the position. The significant shortcoming of the RSSI-based positioning is that it is highly sensitive, even with a slight change, to the surroundings or the orientation of the device.

The Round-Trip Time (RTT) has also been used for ranging because of its advantage as there is no need to perform time synchronization.However, this method introduces another source of error called internal node delay that comes from internal processing of the device. The additional node is usedas a reference to estimate such internal delay [6]. This was done with Wi-Fi protocol by modifying a device driver of an operating system in order to reduce the internal processing time.

Since BLE signal is a radio wave that travels at speed of light, and therefore it needs a high clock resolution in order to measure ToF. The concept of utilizing CPU clock cycle of microprocessor as a stopwatch to measure ToF has been proposed in [7]. This might be a problem in microcontroller to achieve that high resolution. Nevertheless, a reason making ToF possible to be measured is that the modern microcontroller has integrated the RF front-end as its peripheral into a single chip. This leads to the capability to control sending and receiving statesas well as the ability for hardware interrupt at low-level. This helps reducing internal delay and estimatingthe travel time in a more predictable way.

In this paper, BLE has been used for determining pseudo-ranging since it is inexpensive and also found in almost all of smart phones nowadays. To avoid the complication of time synchronization, the RTT scheme is proposed by using a well-established SCAN_REQ protocol defined in [1]. The RTT is then measured by CPU clock. The internal delay will then be determined from the relationship between the distance and RTT of the devices.

3 Pseudo-ranging from Bluetooth Low Energy Beacons

This paper considers the travel time of a BLE beacon packet after sending to the device and waiting for the corresponding response beacon. The communication protocol has already been established in the Bluetooth Core Specifications [1], which means any device conforming to the specifications can be used as a ranging device given that it needs to calibrate priori. Since there is no special hardware or protocol required, this leads to an ease of implementation.

Multipath interference may occur during receiving the incoming beacon. It will degrade to a quality of the signal while demodulating within the physical layer (PHY) of BLE stack. This may cause bit error in the packet, which in turn results in a bad Cyclic Redundancy Check (CRC) checksum. In this paper, only good BLE packets received at Link-Layer level have been considered by verifying a valid CRC checksum associated with the packet. If not, the packet will be discarded.

3.1 Pseudo-ranging Scheme

According to Bluetooth Core Specifications [1] for an unconnected device, it can be mostly either Advertising State or Scanning State. Scanner scans for advertising Packet Data Unit (PDU) broadcasted by advertiser. A single PDU limits data up to 39 bytes. However, the scanner can request for one more PDU by sending SCAN_REQ packet, the advertiser will then response by sending SCAN_RSP packet back. The advertiser is required to advertise the PDU type as ADV_IND or ADV_SCAN_IND in order to inform the scanner that it can response to SCAN_REQ packet.

In this paper, the capability of scannable advertiser has been utilized to measure travel time of SCAN_REQ and SCAN_RSP packet. Firstly, the scanner scans for ADV_IND or ADV_SCAN_IND packets and then sends SCAN_REQ to a target advertiser. Immediately after sending SCAN_REQ, the scanner starts a timer. The SCAN_REQ packet is subsequently traveled to the advertiser. As soon as the advertiser received SCAN_REQ packet, it responses back a SCAN_RSP packet with an empty payload to the scanner. Once the scanner received SCAN_RSP packet, it stops the timer. The total RTT is measured in term of CPU clock cycles, which can be converted to an actual travel time if the CPU clock frequency is known. The timing diagram of measuring the RTT is shown in Fig. 1.

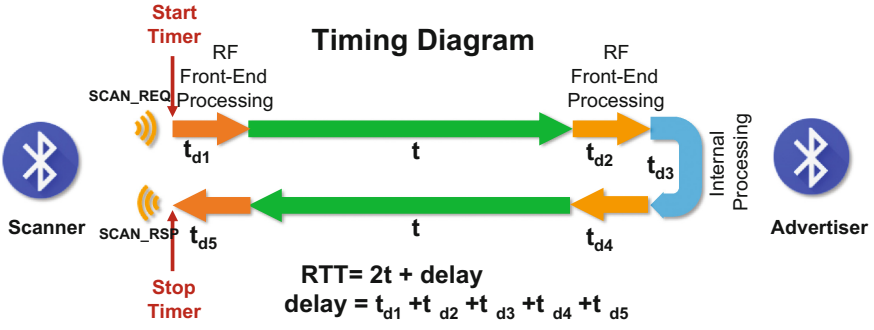


Fig. 1. Timing diagram of RTT determination by sending SCAN_REQ and receiving SCAN_RSP packet.

3.2 Hardware Consideration

In modern BLE modules, they are built based on System on Chip (SoC) microcontroller. It can be programmed to any general purpose software like other microcontrollers. Most of them are based on ARM Cortex-M Series. An nRF5122 from Nordic Semiconductor [8] is one of widely used BLE modules. It is an ultra-low power 2.4 GHz wireless SoC based on a 32 bit ARM-Cortex M0 CPU running at 16 MHz that supports 2.4 GHz multiprotocol including BLE. According to product specifications [9], the radio tasks are operated by RADIO peripheral block that can be controlled by CPU or other peripherals through Programmable Peripheral Interconnect (PPI). The PPI system enables one peripheral to directly activate other peripherals without waking up the CPU. For instance, the RADIO peripheral can start or stop the TIMER peripheral while the CPU is sleeping.

3.3 Delay Time

The delay time is commonly encountered in the RTT ranging approach. This delay comes from an internal processing of both scanner and advertiser. It can adversely reduce the accuracy in ToF-based ranging. Nonetheless, the PPI system in nRF5122 can help reducing the delay since the RADIO peripheral can start TIMER peripheral directly

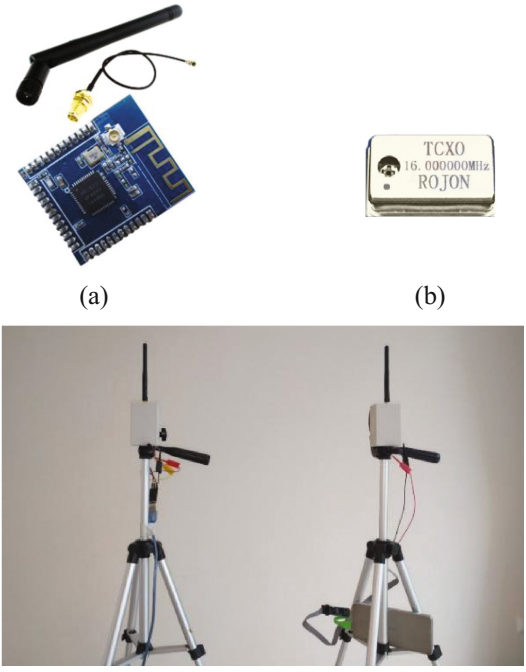


Fig. 2. Hardware implementation. (a) An nRF51822 module with Omni directional antenna. (b) A Temperature Compensated Crystal Oscillator (TCXO). (c) BLE scanner (left) and BLE advertiser (right).

through internal interrupt without waking up the CPU. One of the reasons that causes uncertainty of the delay is a clock drift in the clock source, i.e. a crystal oscillator.

3.4 Implementation

In this research, a firmware has been implemented to perform specific tasks for measuring the RTT in nRF51822. The firmware was programmed in a bear-metal programming fashion without a full-fledged vendor library called SoftDevice [10]. Hence, the implemented firmware includes only necessary tasks for the RTT measurement.

The omnidirectional antenna is provided instead of an on-board antenna of nRF51822 module in order to make rotational invariance.

Since the RTT is measured by using the CPU clock, it is necessary to have a high accuracy oscillator. The external Temperature Compensated Crystal Oscillator (TCXO) is considered in this research for reducing the uncertainty of clock drift. Figure 2 shows the hardware implemented in this research.

4 Experimentation

Regarding to a proof of concept for the RTT ranging using a low-cost device, it needs to minimize unknown factors. Therefore, the experimentation is conducted in a line-of-sight area.

Testingdistance between theadvertiser and thescanner is conducted for one meter apart at a time ranging from 1 m up to 10 m apart. For each observed distance, there are three advertising channels being considered and 5,000 samples were collected for each advertising channel, i.e. 15,000 samples in total. Sample rate used in the study is approximately 20 samples/sec. The RTT and RSSI are collected at each sample (Fig. 3).

Fig. 3. Experimentation setup

5 Results

The probability distribution of CPU cycle counts while measuring the RTT of each advertising channel are illustrated in Fig. 4. The three advertising channels are designated as Channels 37, 38 and 39. The testing distances at 2.0 m, 4.0 m, 6.0 m and 8.0 m are presented in Fig. 4(a–d), respectively.

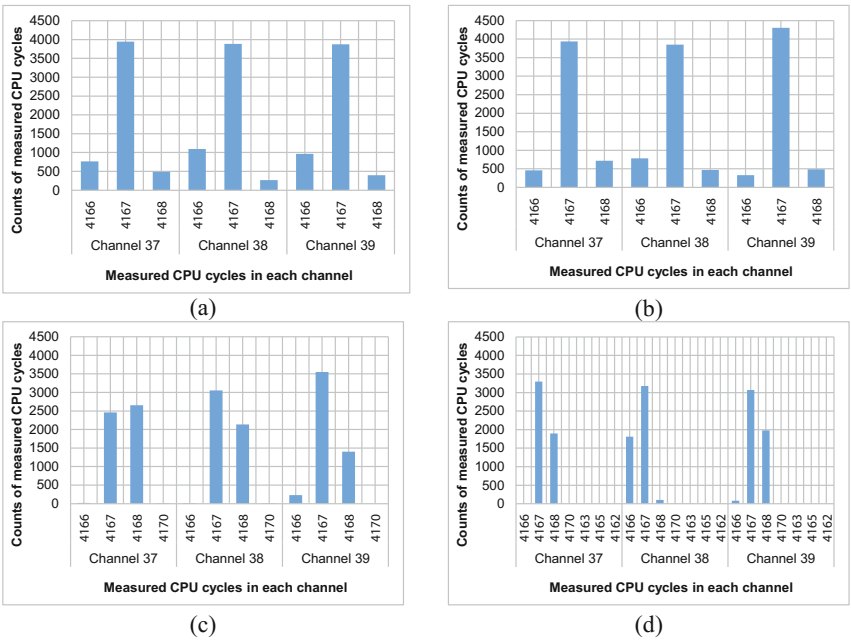


Fig. 4. Distribution of CPU cycle counts in each advertising channel. (a) Data collected at 2.0 m. (b) Data collected at 4.0 m. (b) Data collected at 6.0 m. (b) Data collected at 8.0 m.

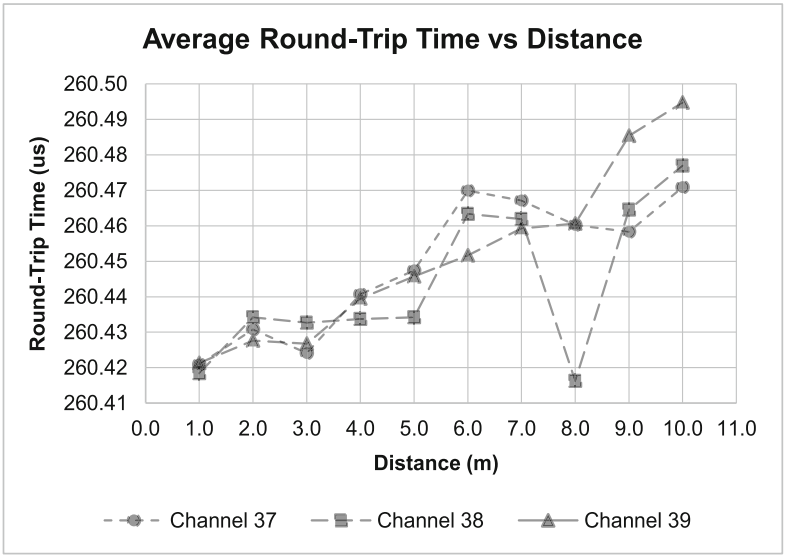
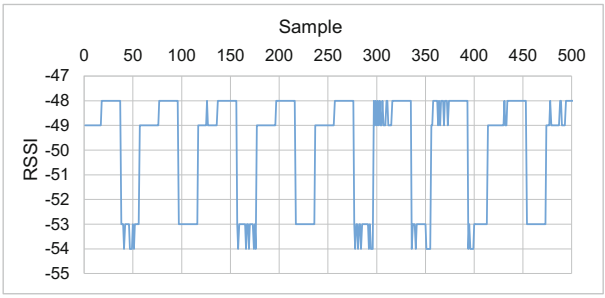
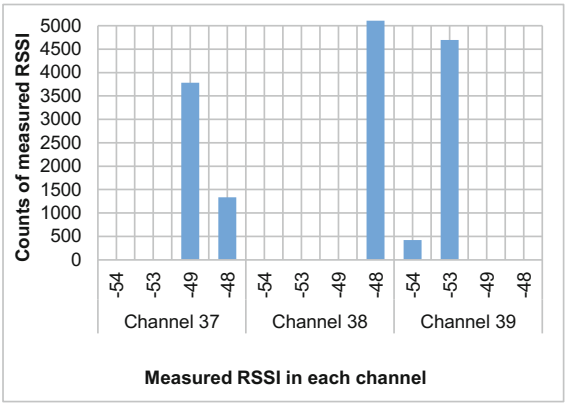


Fig. 5. Relationship between average RTT and distance.



(a)



(b)

Fig. 6. RSSI samples at distance 4.0 m. (a) The first 500 RSSI samples. (b) Distribution of RSSI in each advertising channel.

The CPU cycle counts can be converted to the actual RTT by multiplying the period of CPU clock that is 1/16 μs. The relationships between the average RTT and distance of each advertising channel are plotted in Fig. 5. It can be seen that the relationships are relatively different among the three channels. Channel 39 tends to exhibit the best linearity.

Moreover, it has been observed that the RSSI exhibits a periodic pattern as shown in Fig. 6(a). This is because the scanner periodically scans in Channels 37, 38 and 39. The distribution of the RSSI of all samples in each channel is also shown in Fig. 6(b).

Figure 7 shows that the sensitivity in each channel varies along the distance. For example, the best RSSI at distance 3.0 m is from Channel 38 whereas the best RSSI at 8.0 m is from Channel 37.

Finally, the equation to represent the relationship between the RTT and distance has been developed. This relationship is applied to the specified hardware only. In other words, another experimentation will be required for determining an applicable relationship equation when utilizing a different hardware. The equation can be obtained by

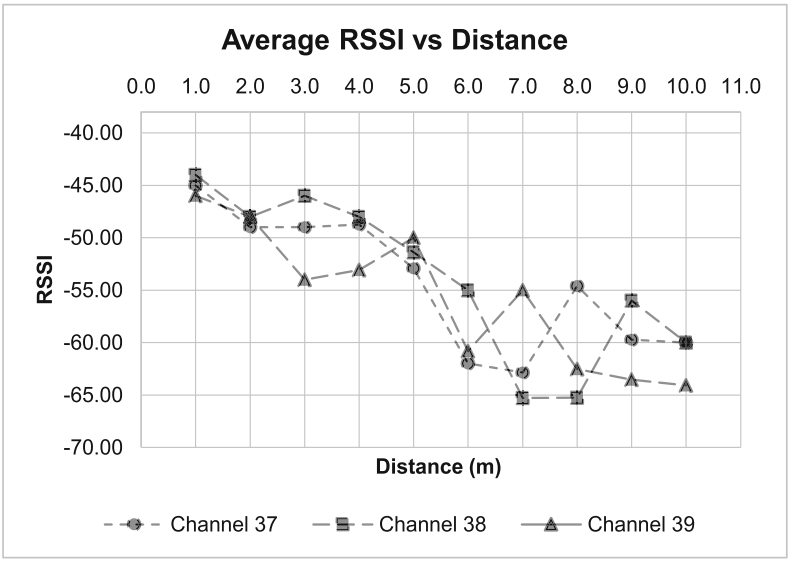


Fig. 7. Average RSSI vs. distance.

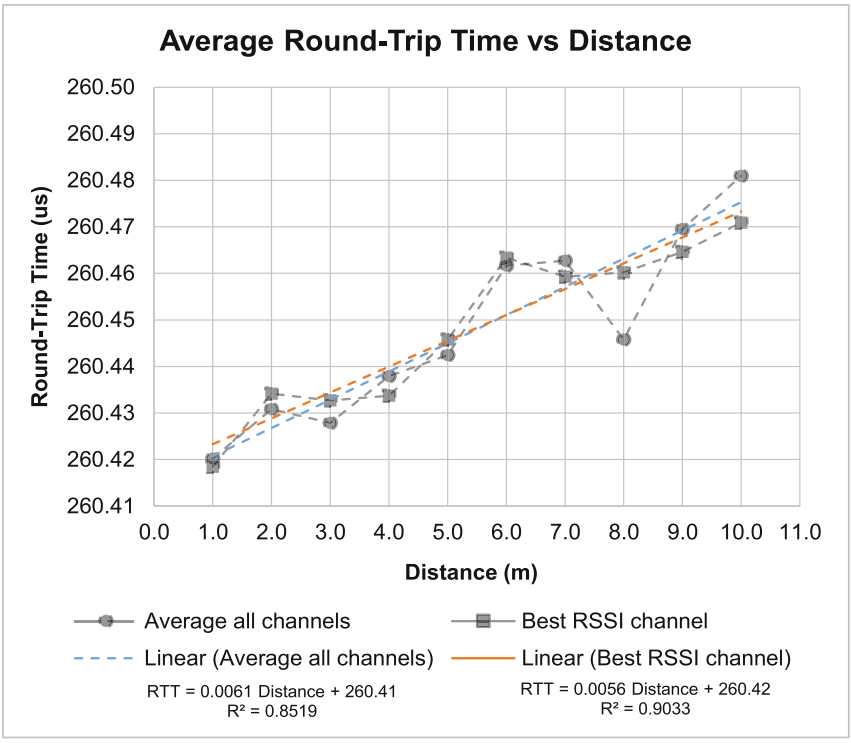


Fig. 8. Representation relation of RTT and distance.

applying the linear regression to the average RTT of all channels shown in Fig. 8 in order to determine parameters as derived in (1).

$$\text{RTT} = 0.0061 \cdot \text{distance} + 260.41;\ R^2 = 0.8519 \quad (1)$$

Alternatively, by considering Fig. 5 in conjunction with Fig. 7, the channel offering the best RSSI can be selected and used to derive the relationship of the RTT and distance as expressed in (2) and shown in Fig. 8. This alternative equation will improve the results as verified by the coefficient of determination, R^2. Thus, Eq. (2) has been used for estimating the distance once the RTT of the BLE packet is known.

$$\text{RTT} = 0.0056 \cdot \text{distance} + 260.42;\ R^2 = 0.9033 \quad (2)$$

6 Conclusions

Pseudo-ranging from the RTT of the BLE beacon has been proposed in this paper. The RTT of the BLE beacon was measured by the CPU clock cycles. Even though the clock runs at low frequency, the accuracy can still be achieved by the statistical approach. The RSSI is also utilized in order to improve the estimation accuracy of the RTT associated with the distance. The results show that the relationship between the distance and RTT tends to be linear. The equation representing the RTT-distance relationship is derived and presented in the paper.

7 Further Study

The study using a new chip named nRF52832, which has a higher clock speed, has been investigated. This chip is an Arm Cortex-M4 running 64 MHz, which gives more clock resolution than its predecessor chip. However, the RADIO peripheral still operates at 16 MHz. Therefore, the clock resolution is constrained by its RF front-end module. The new BLE chip having a higher RADIO peripheral clock is preferable and under an investigation.

Acknowledgements. This research is financially supported by Thailand Advanced Institute of Science and Technology (TAIST), National Science and Technology Development Agency (NSTDA), Tokyo Institute of Technology, Sirindhorn International Institute of Technology (SIIT), Thammasat University (TU) under the TAIST Tokyo Tech Program.

References

1. Bluetooth SIG: Bluetooth Core Specification v4.2 (2014)
2. Wang, Y., Yang, X., Zhao, Y., Liu, Y., Cuthbert, L.: Bluetooth positioning using RSSI and triangulation methods. In: 2013 IEEE 10th Consumer Communications and Networking Conference CCNC 2013, pp. 837–842 (2013)

3. Dong, Q., Dargie, W.: Evaluation of the reliability of RSSI for indoor localization. In: 2012 International Conference on Wireless Communications Underground Confined Areas, ICWCUCA 2012, pp. 2–7 (2012)
4. Faragher, R., Harle, R.: Location fingerprinting with bluetooth low energy beacons. IEEE J. Sel. Areas Commun. **33**, 2418–2428 (2015)
5. Rodrigues, M.L., Vieira, L.F.M., Campos, M.F.M.: Fingerprinting-based radio localization in indoor environments using multiple wireless technologies. In: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC, pp. 1203–1207 (2011)
6. Park, S., Ahn, H.S., Yu, W.: Round-trip time-based wireless positioning without time synchronization. In: ICCAS 2007—International Conference on Control Automation and Systems, pp. 2323–2326 (2007)
7. Casacuberta, I., Ramirez, A.: Time-of-flight Positioning using the existing wireless local area network infrastructure. In: Proceedings of International Conference on Indoor Positioning Indoor Navigation, pp. 1–8 (2012)
8. Nordic Semiconductor ASA: nRF51822 - Bluetooth Low Energy Products. https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF51822
9. Nordic Semiconductor ASA: nRF51822 Product Specification v3.3 (2016)
10. Nordic Semiconductor ASA: S132 SoftDevice Specification v3.0 (2016)

A Three Level Architecture for Wireless Communication Using Li-Fi

Satyanarayana Degala[1(✉)] and Sathyashree Selvaraj Degala[2(✉)]

[1] Department of Electrical and Computer Engineering, University of Buraimi, Al Buraimi, Sultanate of Oman
degala.s@uob.edu.om

[2] Department of Information Technology, College of Applied Sciences, Sohar, Sultanate of Oman
sathya_degala.soh@cas.edu.om

Abstract. The usage of wireless radio devices has increased at an exponential rate as the manufacturers produce the mobile devices at cheaper prices. Accordingly, the radio spectrum used by these devices has been exhausted due to the high usage of wireless applications. The Li-Fi technology is used as an alternative method of wireless radio communication to hold back the radio spectrum. In this paper, we have proposed a three level architecture for wireless communication using Li-Fi system, which can reduce the radio spectrum scarcity. The simulation work shows that the proposed architecture provides wireless communication system with high date rate than the traditional wireless radio systems.

Keywords: Wireless communication · Radio spectrum · Visible light communication · Light-Fidelity

1 Introduction

The wireless communication is more on demand due to its wide range of applications and high usage of mobile devices. Currently, majority of the wireless communication has been deployed to use the radio spectrum. However, the use of radio spectrum has grown in the exponential rate as the number of wireless users have increased in the current world; hence the radio spectrum scarcity is considered as a serious problem. Several researchers have worked on a promising technology called Cognitive Radio Networks (CRN) [1, 2], which increases the spectral efficiency of the networks. The CRN collects all the unused channels of the primary users and allocates them to the secondary users. If a primary user needs the channel, the secondary users will wait until the channels of primary user are free. The CRN merely optimizes the allocation of existing radio channels, but the demand for the radio spectrum is increasing heavily every day. An alternative solution to this bandwidth scarcity problem is to use visible light as communication medium rather than the radio spectrum. Nevertheless, the visible light is a license free spectrum and it is 10000 times larger than radio spectrum [22], see the Fig. 1. The latest technology light fidelity (Li-Fi) is a high bandwidth wireless communication technology that uses the visible light as the communication

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_20

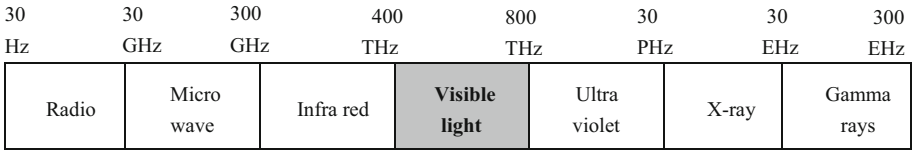


Fig. 1. The spectrum ranges.

medium [3, 4]. The Li-Fi uses Visible Light Communication (VLC) technology to transfer the data using the line of sight propagation mode. The Li-Fi uses cost effective transceivers, precisely the LEDs as transmitter and the photo detectors as receiver for the communication. Since the Li-Fi uses line of sight propagation mode, it is not recommended to use the Li-Fi in all the places. Precisely, the Li-Fi can be more suitable for indoor communication such as in offices, homes, and in-building communication systems. The characteristics of visible light spectrum require a careful design of Li-Fi communication architecture. The author Satyanarayana et al. [13] has proposed a two level architecture for wireless communication using Li-Fi, where the Level 1 uses the Li-Fi communication system and Level 2 uses the core communication network. This type of architecture has its own limitations on the scalability of the network. In other words, there is a need to design an architecture for flexible deployment of Li-Fi in the current communication systems. Our proposal aims to achieve this by proposing a three level architecture for Li-Fi which is more flexible for deployment of the network and also scalable to the existing communication systems.

The Sect. 2 describes the definition and related work. The Sect. 3 explains the three level architecture for Li-Fi. The Sect. 4 describes the simulation work and finally, the conclusions are described in the Sect. 5.

2 Related Work

The wireless communication is a method for communication between the users without wires. The most popular method is wireless radio communication, which uses the radio as communication medium. The radio spectrum has been pre-allocated by the government agencies. On the other hand, the pre-allocation has restricted the users to use only specific radio band for the tasks. But the exponential growth in the number of mobile devices or applications has exhausted all the spectrum of public users. To overcome this problem, the researchers have invented a new technology called Cognitive radio Networks [1, 2], which collects the unused channels from other spectrum band and allocates to the cognitive users. The wireless communication using visible light can provide many opportunities to solve the problem of radio spectrum scarcity. On this basis, the researchers have explored the wireless communication using visible light communication (VLC) [5, 6, 22].

The VLC has many advantages over radio communication. One of the main advantages is that the visible light has large amount of free spectrum available. This license free spectrum is 10000 times larger than the radio spectrum [22]. In addition,

the spectrum is safe and harmless for the living beings and the environment. A researcher Harald et al. described Light Fidelity (Li-Fi) which provides much higher data rates of up to 3 Gbps than radio communication [3, 4, 7, 8]. The IEEE communication standard for Li-Fi is 802.15.7 [5]. In this standard, the medium access control and physical layer details are described. Conversely, it is necessary to do further research at the network, transport and application levels for efficient and reliable communication.

The devices used for Li-Fi communication system are 10 times cheaper than the devices used for radio communication. This includes LED bulbs for transmitting the data and the photo detectors for receiving the data over the visible light. The binary data transmitted by LED will be captured by photo detectors in the form of light receptors, which in turn transfers the data to different types of connecting devices such as computer tablets, phones, televisions, or appliances. The light pulses which are used for generating the binary data does not cause any damage or discomfort to the eye as the LED light pulses are imperceptible to human eye. The working procedure for Li-Fi is as follows: the LEDs are placed at the transmitter end and the photo detectors (light sensors) are placed at the receiver end. The LED is switched ON if the user desires to send a binary one and the opposite procedure follows for the binary zero, see the Fig. 2.

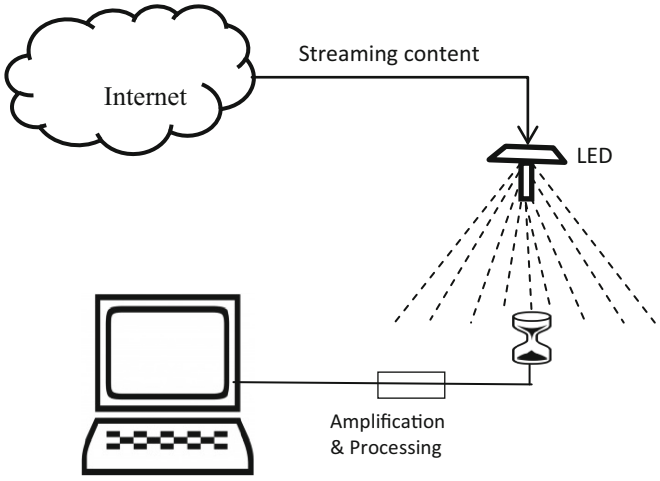


Fig. 2. The binary data transmission using LED and photo detector

The IEEE 802.15.7 standard defines physical layer and MAC layer properties for Li-Fi communications [5]. Since the standard provides high data rates, the Li-Fi can be used for many applications that require high data rates such as audio, video and streaming communication services [6, 11, 12]. The MAC layer provides services to TCP/IP protocol at transport and network layers, respectively. The physical layer provides three different data rates at different contexts. The PHY 1 is prepared for outdoor communications with the data rate from 11.67 to 267.6 kbps. The PHY II layer

allows the communication from 1.25 to 96 Mbps, whereas PHY III is used for high data rates from 12 to 96 Mbps.

An author Saito et al. [21] has proposed an efficient method for downlink of multiuser access for an attocell in Li-Fi communication. The attocell is a cellular structure used for Li-Fi [22]. The attocell does not interfere with radio frequency (RF), hence the RF network performance is improved if both types of networks are installed in the same place.

The access point (AP) of Li-Fi is placed on the ceiling of the room and the k mobile users are scattered underneath. The spectral efficiency of such a system is denoted in Eq. (1).

$$T_k = \begin{Bmatrix} \log\left(1 + \frac{(h_k a_k)^2}{\sum_{i=k+1}^{k} (h_k a_i)^2 + \frac{1}{p}}\right), & k \neq K \\ \log_2\left(1 + p(h_k a_k)^2\right), & k = K \end{Bmatrix} \tag{1}$$

where $h_1 \leq \ldots \leq h_k \leq \ldots \leq h_K$. h_k represents the channel gain of the k_{th} user, a_k represents the power partition parameter of the k_{th} user, $a_1 \leq \ldots \leq a_k \leq \ldots a_K$, and p represents the transmitted Signal to Noise Ratio (SNR).

The single carrier modulation methods for Li-Fi are similar for wireless infrared communication systems [14]. These methods include Pulse Amplitude Modulation (PAM), On-Off-Keying (OOK), and Pulse Position Modulation (PPM). The PPM method saves power better than OOK. A variant of pulse position modulation method is called Variable PPM (VPPM) [15]. A novel method for a single carrier and modulation scheme called as Optical Spatial Modulation [16] is both power efficient and bandwidth efficient for indoor optical wireless communication. In the optical wireless communication, the multiuser modulation (MCM) method is used to increase the speed of the data transfer. The most common MCM scheme is OFDM [17, 18], which can transmit the parallel data streams simultaneously. The variations of OFDM are Asynchronous metrically Clipped Optical OFDM (ACO-OFDM) [19] and Asymmetrically clipped Direct current biasing OFDM (ADO-OFDM) [20].

It is important to discuss about the light propagation phenomena as Li-Fi uses the visible light as the communication medium. The Maxwell derived the Eqs. (2), (3), (4), and (5) that describe how the light signal is converted to electricity [9]. The magnetic and electric fields are constrained to the z and y directions, respectively and are functions of only x and t. From the Maxwell's equation, the wave equations in the free space are

$$\nabla \cdot E = 0 \tag{2}$$

$$\nabla \cdot B = 0 \tag{3}$$

$$\nabla_x E = -\frac{\partial B}{\partial t} \tag{4}$$

$$\nabla_x B = \mu_0 \varepsilon_0 \frac{\partial B}{\partial t} \tag{5}$$

where

∇ is the divergence,
∇_x is the curl equation,
E is Electric field,
B is Magnetic field,
$\mu_0 = 4\pi \times 10^{-7}$ (H/m), and $t_0 = (1/36\pi) \times 10^{-9}$

From the above equations, the speed of the light has been derived.

The first ever commercial product for Li-Fi technology is OLEDCOMM [10]. The product provides the users with facilities like listening to music, connecting to the internet, and playing the videos through the LED bulbs attached in the ceiling of room. The researchers are actively exploring different solutions for the problems of Li-Fi in advance to the arrival of products in the market. It has been presented at the Fudan University that the communication using Li-Fi is carried at the data rate if 15 Mbps by using few LEDs. Furthermore, the speed can be increased up to 3.5 Gbps.

The author Satyanaranaya et al. [13] has proposed architecture for deploying Li-Fi into the current communication system, see the Fig. 3. This architecture is not flexible for deploying large scale networks. To solve this problem, we have proposed a three level architecture for Li-Fi which is described in the next section.

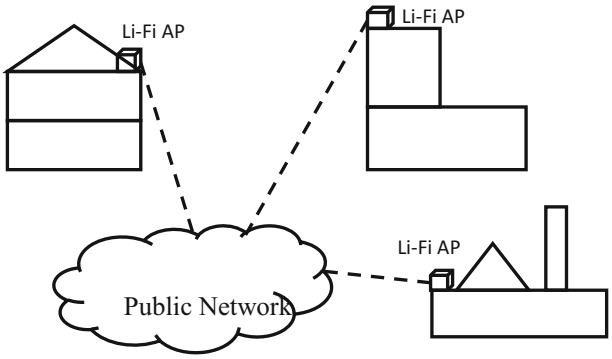


Fig. 3. Data communication between Li-Fi APs and public network.

3 The Proposed Architecture

The scarcity of radio spectrum due to the high usage of mobile devices and their applications lead the world to consider an alternative method for wireless radio communication. On these lines, the researchers have invented a new technology called Light Fidelity (Li-Fi) [3, 4]. The Li-Fi uses the visible light as the communication medium. In other words, there exists abundant unused visible light spectrum frequency

which is 10000 times larger than radio spectrum [22]. In the Li-Fi system, the visible light is generated using LEDs at the transmitter and the photo detectors (light sensors) at the receiver to decode the light signals. Some researchers have claimed that the amount of data to be transmitted with this technology can be 100 times larger than the radio wireless communication systems [6, 11].

One of the important limitations of Li-Fi is the line of sight propagation. Hence, the Li-Fi can be more suitable for indoor communication rather that outdoor because of the hurdles such as buildings, walls, and mountains. Alternatively, there is a need for careful design of Li-Fi communication system without changing the current communication system. The author Satyanarayana et al. [13] has proposed a two level architecture, where the level 1 has Li-Fi communication system and level 2 has the core communication system. However, the system is not very flexible for deploying in heterogeneous networks. In addition, the architecture is not scalable for large networks. To provide a solution to these problems, we propose a three level architecture which provides Li-Fi the flexibility and scalability for the current communication system.

The proposed architecture has three levels, see the Fig. 4. In the level 1, the terminal network is built up of Li-Fi systems, where the networking is carried out generally in homes, offices and indoor buildings. The main reason for using Li-Fi system in level 1 is because of the properties of the visible light spectrum. In addition,

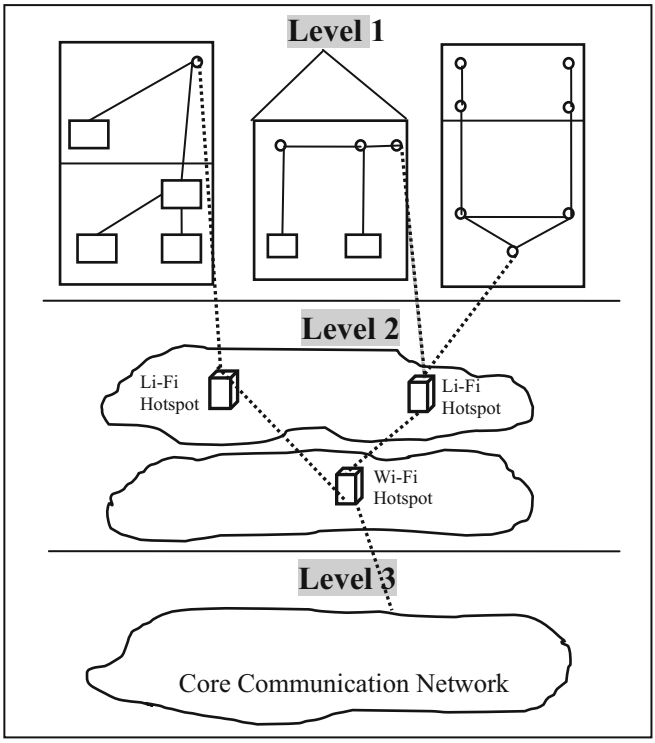


Fig. 4. Three level architecture

the survey says that 70% of the radio communication is used in homes, offices and in-building wireless communications [23]. Since the Li-Fi communication system is deployed at the level 1, the end users, such as mobile devices, will use visible light spectrum to replace the radio spectrum which solves wireless communication challenge: the radio spectrum scarcity.

The proposed architecture builds level 2 for two objectives. The first objective is to allow the Li-Fi home networks to connect to the outside world or internet. To achieve this, the Li-Fi hotspots are placed. These hotspots provide the flexibility of connecting the home networks to the outside world. The Li-Fi hotspot works like a base station to all the home networks. The second objective of level 2 is to transform the light signals into radio signals and hence the Li-Fi hotspots connect to the Wi-Fi. At this point, a logical line of boundary exists between the Li-Fi based system and the existing wireless communication system. The Wi-Fi is connected to the core communication network in the level 3, which is the backbone network in the communication system. The architecture does not change the core communication infrastructure because replacing the core communication network is very expensive.

The level 2 provides modularization of terminal networks from the core communication networks. That is, the level 2 avoids the structural changes of the core communication network with the changes in level 1 Li-Fi network. Therefore, the design is more flexible for deploying the Li-Fi in the current communication system.

The proposed architecture has the following advantages:

1. The architecture provides high data rates as it uses Li-Fi in the communication system.
2. It saves many radio spectrum channels to solve the problem of spectrum scarcity.
3. The architecture provides more flexibility since it has adopted the level 2. On the contrary, connecting directly from the Li-Fi to the core network will have limitations on the deployment.
4. It provides a healthy and environment friendly communication as the visible light is safe for humans compared to the radio signals.

4 Simulation

We have developed code for the simulation by considering all the required network parameters. In the simulation, though there is no restriction on selecting a specific network topology, we have considered the network shown in the Fig. 5. In this network, the Li-Fi is used at the terminals as they represent the home networks, hospitals, offices, and universities. The data transfer rates for Li-Fi and public networks are considered as 3 Gbps and 200 Mbps, respectively. To compare our model with the traditional network, we have used the same network topology by replacing the Li-Fi home networks with WLAN that has the data rate of 11 Mbps. To distinguish the WLAN in the graphs, the proposed architecture is named as TLALF which acronyms Three Level Architecture for Li-Fi.

The simulation is carried out by considering 10 connection patterns. In the simulation, we have randomly selected 10 nodes as source for the data transmission at one

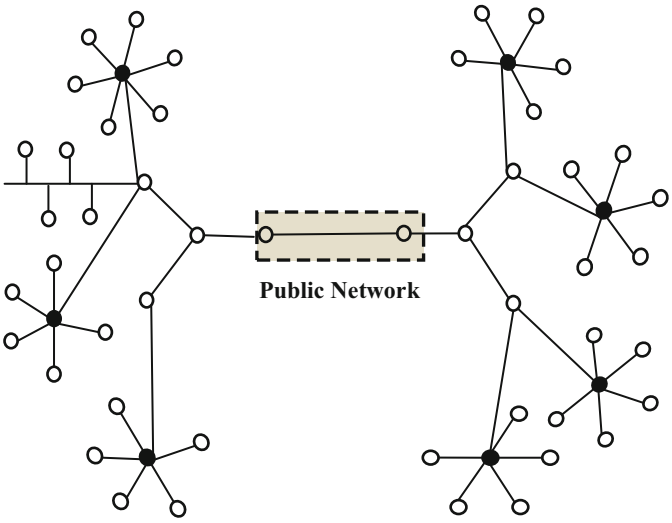


Fig. 5. Network topology for simulation.

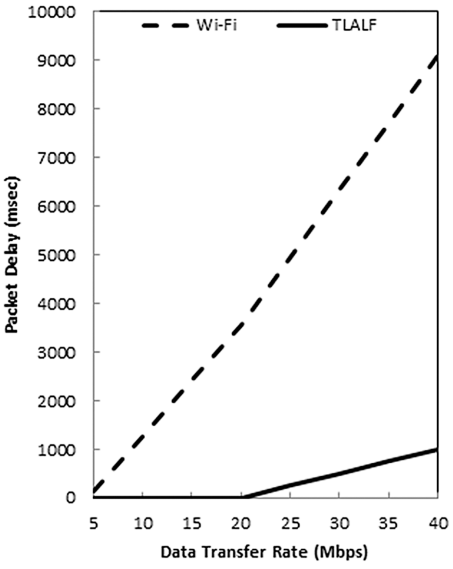


Fig. 6. Packet delay of TLALF and Wi-Fi

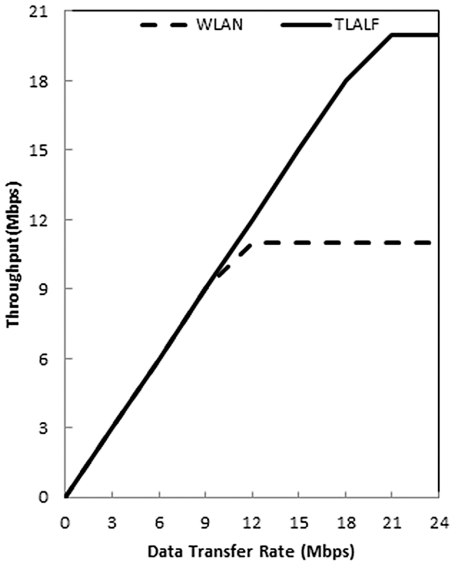


Fig. 7. Throughput of TLALF and WLAN.

end of the public network and 10 nodes are selected as destination for receiving the data at the other end. By considering the data transfer rate as control parameter for the simulation, we send the data from one end of the public network to the other end with

the data rates as 5, 10, 15, 20, 25, 30, 35, and 40 Mbps. The end-to-end packet delay is calculated. Similarly, the WLAN based network is chosen with the same data rate and the end-to-end packet delay is calculated. The results show that the network with Li-Fi based communication system has less end-to-end packet delay compared to the network with WLAN based communication system, see the Fig. 6. This is because the high bandwidth in Li-Fi conveys the packets with low delay, whereas the WLAN has the end-to-end packet delay higher than the TLALF based systems. The second experiment is for analyzing the throughput, which is defined as the received data in one second. We have chosen 10 connection patterns and the data is sent with the data rates of 3, 6, 9, 12, 15, 18, 21, and 24 Mbps in both TLALF based system and WLAN based system. From the graph, as in the Fig. 7, we say that the throughput for Li-Fi based system is higher than the WLAN based system. This is because the channel capacity for TLALF based system is higher than the WLAN based system.

5 Conclusion

Today's world faces the heat of radio spectrum scarcity due to the heavy usage of mobile devices and applications. To solve this problem, the researchers have invented an alternative technology named as Li-Fi which uses the visible light spectrum for data transfer. In this paper, a three level architecture is proposed for Li-Fi communication system. The proposed architecture has the benefits of high bandwidth and solves the spectrum bottleneck problem. The simulation results show that the proposed architecture for wireless communication has high data rates compared to the traditional radio networks. The proposed architecture can be applied to large scale networks comfortably than the small networks.

References

1. Mitol, J., Maguire, G.J.: Cognitive radio: making software radios more personal. IEEE Pers. Commun. **6**(4), 13–18 (1999)
2. Mitola, J.: Cognitive radio: an integrated agent architecture for software defined radio. Ph.D. Dissertation, Royal Institute of Technology in Sweden (2000)
3. Dimitrov, S., Haas, H.: Principles of LED Light Communications: Towards Networked Li-Fi. Cambridge University Press, Cambridge (2015)
4. Haas, H.: Wireless data from every light bulb. TED Website (2011). http://bit.ly/tedvlc
5. Rajagopal, S., Roberts, R., Lim, S.-K.: IEEE 802.15.7 Visible light communication: modulation schemes and dimming support. IEEE Commun. Mag. **50**(3), 72–82 (2012)
6. Khalid, A.M., Cossu, G., Corsini, R., Choudhury, P., Ciaramella, E.: 1-Gb/s transmission over a phosphorescent white LED by using rate-adaptive discrete multitone modulation. IEEE Photonics J. **4**(5), 1465–1473 (2012)
7. Haas, H., Chen, C.: What is LiFi. In: 41st European Conference on Optical Communication, pp. 1–3, Valencia (2015)

8. Tsonev, D., Videv, S., Haas, H.: Light Fidelity (Li-Fi) towards all-optional networking. In: Photonics West Conference on Broadband Access Communication Technologies VIII, SPIE 900702, Canada (2013). doi:10.1117/12.2044649
9. Orfanidis, S.J.: Electromagnetic Waves and Antennas. Rutger University Press, Piscataway (2008)
10. The Pioneer of Li-Fi Technology. http://www.oledcomm.com/home/parallax/01-parallax
11. Cossu, G., Khalid, A.M., Choudhury, P., Corsini, R., Ciaramella, E.: 3.4 Gbit/s visible optical wireless transmission based on RGB LED. Opt. Express **20**(26), B501–B506 (2012)
12. Azhar, A., Tran, T., O'Brien, D.: A gigabit/s indoor wireless transmission using MIMO-OFDM visible-light communications. IEEE Photonics Technol. Lett. **25**(2), 171–174 (2013)
13. Satyanarayana, D., Alex, M., Sathyashree, S.: An architecture for wireless communication using Li-Fi technology. In: Proceedings of the 8th International Conference on Latest Trends in Engineering and Technology, pp. 37–41, Dubai (2016)
14. Kahn, J.M., Barry, J.R.: Wireless infrared communications. In: Proceeding of IEEE, pp. 265–298 (1997)
15. IEEE 802.15.7 standard for Local and Metropolitan Area Network, Part 15.7.: Short-Range Wireless Optical Communication Using Visible Light. New York (2011)
16. Mesleh, R., Elgala, H., Haas, H.: Optical spatial modulation. IEEE/OSA J. Opt. Commun. Netw. **3**(3), 234–244 (2011)
17. Komine, T., Haruyama, S., Nakagawa, M.: Performance evaluation of narrowband OFDM on integrated system of power line communication and visible light wireless communication. In: International Symposium on Wireless Pervasive Computing, pp. 1–6, Phuket (2006)
18. Afgani, M.Z., Haas, H., Elgala, H., Knipp, D.: Visible light communication using OFDM. In: Proceedings of the 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, pp. 129–134, Spain (2006)
19. Armstrong, J., Lowery, A.J.: Power efficient optical OFDM. Electron. Lett. **42**(6), 370–372 (2006)
20. Dissanayake, S., Panta, K., Armstrong, J.: A novel technique to simultaneously transmit ACO-OFDM and DCO-OFDM in IM/DD systems. In: IEEE GLOBECOM Workshops, pp. 782–786, Houston (2011)
21. Saito, Y., Kishiyama, Y., Benjebbaour, A., Nakamura, T., Li, A., Higuchi, K.: Non-orthogonal multiple access (NOMA) for cellular future radio access. In: IEEE Vehicular Technology Conference, pp. 1–5, Dresden (2013)
22. Harald, H.: High-speed wireless networking using visible light. SPIE Newsroom (2013)
23. Chandrashekhar, V., Andrews, J., Gatherer, A.: Femtocell networks: a survey. IEEE Commun. Mag. **46**(9), 59–67 (2008)

Overhead Reduction for Route Repair in Mobile Ad Hoc Networks

Worrawat Narongkhachavana and Sumet Prabhavat(✉)

Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
57606007@kmitl.ac.th, sumet@it.kmitl.ac.th

Abstract. In Mobile Ad Hoc Networks (MANETs), routing overhead is a major problem since a bandwidth and an energy of mobile node are limited. Due to node's mobility and an unreliable wireless link, a path connecting between a source and a destination will break frequently. This causes a network-wide flooding of routing packets being invoked repeatedly, leading to high packet loss and network congestion, especially in a high mobility network. In this paper, we propose an efficient route repair method based on Query Localization technique. The routing packets, which propagate back to a source, are discarded to alleviate unnecessary rebroadcasting. Simulation results show that our proposed method can reduce routing overhead and MAC collision rate without sacrificing packet delivery ratio compared to existing protocols.

Keywords: Mobile Ad Hoc Networks · Route repair · Query Localization

1 Introduction

Mobile Ad Hoc Networks (MANETs) are groups of mobile nodes connecting via a wireless link. Since there is no centralized administration required, MANETs are designed to be deployed in distant areas such as a disaster area, a rural region, and a battlefield. Since nodes in MANETs can operate without fixed physical infrastructure, they can move independently during network operation.

In MANETs, a routing protocol is used for finding a path between faraway communication endpoints. Due to a small transmission radius, each node has to relay a packet for other nodes to enable multi-hop communication. In on-demand routing protocol, DSR [1] and AODV [2], routing packets are generated only when a source need to send data packets and the only path to the required destination is maintained. In proactive routing protocol, DSDV [3] and OLSR [4], a node periodically broadcasts routing packets and attempts to maintain paths to every node in the network. These conventional routing protocols are based on flooding mechanism. The routing packets are propagated over the network during the routing operation. Due to an unreliable wireless medium and a mobility of nodes, a connection between nodes will be periodically disconnected. This frequently triggers network-wide flooding of routing packets to repair the path to the destination. As the node's velocity increases, an enormous number of routing packets is generated due to route breaks, leading to the broadcast storm problem. This results in high network congestion and energy

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_21

consumption. Therefore, reducing overhead would lead to a better performance of communication in MANETs.

In this paper, we propose an extension to the conventional routing protocol, Query Localization [6]. We describe a method which makes route repair more efficient. We also use the number of hops to the destination to restrict forwarding nodes. The routing packets, which traverse back to a source, are discarded. This results in routing packets are directionally rebroadcasted toward the destination which can alleviate routing overhead and increase the success of end-to-end communications.

The rest of this paper is organized as follows. Section 2 describes related works. Our proposed model is described in detail in Sect. 3. Simulation results are discussed in Sect. 4. Section 5 summarizes the conclusions of our work.

2 Related Works

Ad hoc On-Demand Distance Vector (AODV) [2] is an on-demand routing protocol. First, a source initiates route discovery by originating Route Request (RREQ) packet when it has data packets to send and the route is unavailable. The RREQ packet contains a source address, a source sequence number, a destination address, a destination sequence number, broadcast ID and the number of hops to the source. The broadcast ID is used for preventing broadcast loops. The RREQ table is maintained in each node to record a broadcast ID of every received RREQ. When a node receives non-duplicated RREQ, it creates a reverse path back to the source in its routing table. Then, if there is an up-to-date routing information available, the node responds by sending Route Reply (RREP). The RREP packet contains the destination address, the destination sequence number and the number of hops to the destination. Otherwise, the node rebroadcasts the RREQ packet. When the destination receives the RREQ packet, it sends the RREP packet back along the reverse path to the source. A node, which receives the RREP packet, creates a forward path to the destination in its routing table. When the source receives RREP, the buffered data packets are sent along the recorded forward path.

When an intermediate node fails to forward a data packet to a next hop node, it assumes that a path to a destination is broken. The node then invalidates the corresponding routing entry in its routing table. The destination sequence number is increased by one. Then, the route maintenance process is started to repair the route. If the number of hops to the destination is no farther than MAX_REPAIR_TTL hops away, the node performs a local repair by originating a RREQ packet instead of the source. The data packet is buffered during the local repair process. Otherwise, a source repair is performed. The node originates and broadcasts a Route Error (RERR) packet to inform other nodes of the route breakage. A node, which receives the RERR packet, finds the corresponding route entry in its routing table. If there is a match, the node then marks the route entry as invalid, updates the sequence number and rebroadcasts the packet. After the source receives the RERR packet, it reinitiates the route discovery by sending a new RREQ packet.

In Expanding Ring Search (ERS) [5], the TTL field in the IP header is used for limiting the propagation of RREQ packets. A source set the TTL value in the IP header to TTL_START. If the node knows the previous number of hops to a destination, the TTL value is set to the old hop counts plus TTL_INCREMENT. When the route

discovery fails, the source increases the TTL value by TTL_INCREMENT and resends a new RREQ packet.

In [7], Extended AODV is proposed to reduce routing overhead from RREQ flooding. This work is based on AODV with ERS and assumes that node does not move quickly so that previous invalid routes could be usable again. When an intermediate node receives the RREQ packet, if a previous invalid path to the destination is available, the node forwards the packet to the next hop by unicast instead of the conventional broadcast. The unicast RREQ will be forwarded along the old invalid path until it reaches the destination. This can significantly reduce the broadcast storm of the RREQ packets. However, there is a high chance that the old invalid path may break which cause RREQ fails to reach the destination.

In [8], the number of forwarding messages is reduced by assigning tokens to each generated message. The message is also forwarded with some adjustable probability. In [9], an aggregation technique is applied to reduce the neighbor discovery overhead and the broadcast transmission is used to distribute data packets. The works in [10, 11] use a physical location information of nodes to control the propagation of a broadcast message.

In Query Localization (QL) [6], the assumption is a new path can be found within the local region of the previously used route. A RREQ packet is distributed no farther k hops from nodes in the old path. First, a source performs a conventional flooding of RREQ packet. Then, a destination responds by sending a RREP packet back along the reverse path to the source. When a node receives the RREP packet, it updates its routing table according to AODV rules. After that, the node records the source and the destination from the packet into an additional table.

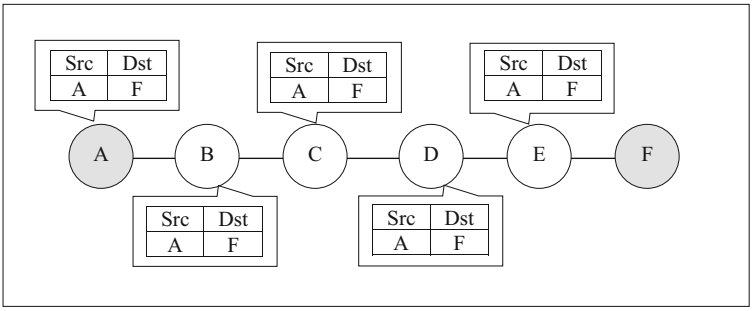
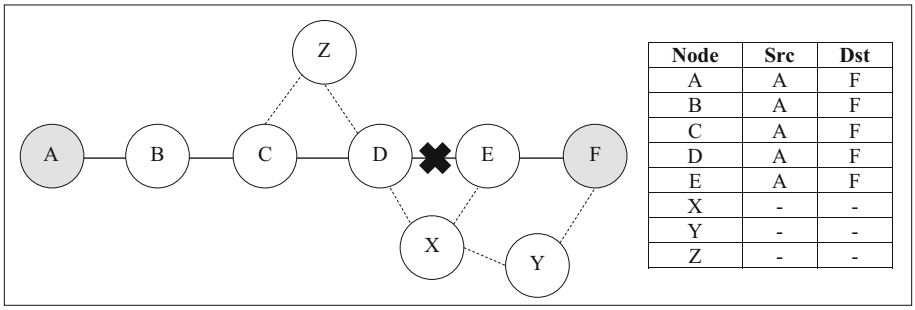


Fig. 1. Example topology and recorded information of QL

For example, Fig. 1 illustrates a path from source A to destination F. Each node records the source-destination pair (A and F) and assign an expiration time to the entry (due to the limited space, the expiration time column is omitted from the figure). When the route breaks and source A generates a RREQ packet, a counter (initially set to zero) and the predetermined value of k threshold are attached to the RREQ packet. If the packet is rebroadcasted by a node which has the unexpired entry of the corresponding source-destination pair (i.e. node B, C, D, and E), the counter is updated depends on the

selected method, Node locality and Path locality. In Node locality, the counter is reset to zero while remains the same in Path locality. Otherwise, the counter is incremented by one. When the counter of the received RREQ packet equals to or exceeds *k* threshold, the packet will be dropped.



Node	Src	Dst
A	A	F
B	A	F
C	A	F
D	A	F
E	A	F
X	-	-
Y	-	-
Z	-	-

Fig. 2. Example topology when an intermediate node (D) performs a local repair in QL

Figure 2 shows the example topology and a recorded source–destination pair of each node when node D fails to reach its next hop E. Node D performs a local repair. It then generates a RREQ packet, which the source address is set to itself, and the target destination address is set to node F. Normally, node D will perform the normal AODV local repair if it has not recently initiated a route discovery to destination F. This cause a lot of flooding packets. Otherwise, if QL is applied, node D will originate the RREQ packet with the counter and the threshold (set to 1 in this example). When node X and node Z receives the packet, it updates the counter to 1 and rebroadcasts the packet. However, the recorded source-destination entry in node E is A-F, not D-F. Node E will act as it was not in the previous active route. Therefore, node E and node Y will eventually drop the RREQ packet because the counter in the packet equals to the *k* threshold. From this situation, the RREQ packet originated from the intermediate node during a local repair may hardly to propagate to the destination.

3 Proposed Method

In this section, we introduce an extension to improve the performance of route repair in Query Localization (QL) [6]. We focus on reducing unnecessary RREQ packets which are rebroadcasted backward to a source. In addition, as we mentioned in the previous section, an intermediate node will perform a normal local repair if it has not recently performed a route discovery to the destination. This produces lots of routing packets. On the contrary, if QL is applied, the RREQ packet will be dropped earlier before it can reach the destination since QL use a source-destination pair to modify the counter while the source address of a RREQ packet in the local repair is set to the upstream node of the broken link. The problem might be solved by initially setting the *k* threshold higher. However, this causes more unnecessary RREQ packet.

In our proposed protocol, we do not use a source-destination pair to control the value of the counter. Instead, each node maintains a data structure, Ptable, containing a destination address, the number of hops to the destination and an expiration time as illustrated in Fig. 3 (due to the limited space, the expiration time column is omitted from the figure). When an intermediate node receives a RREP packet, it records the target destination of the RREP packet, the number of hops to the destination and assigns an expiration time into the table. Since it is not necessary to concern the source of the previous path that a node lie on, we use only a destination address to control the value of the counter of the RREQ packet. The number of hops is used for steering the RREQ packet outward from the source toward the destination.

When the source perform route discovery or the intermediate node performs a local repair, it originates RREQ similar to QL. In addition, it also attaches the previous number of hops to the destination from Ptable to the packet. When a node receives the RREQ packet, it checks if there is an unexpired entry of the corresponding destination exists in its Ptable. If there is no matched entry, this means that the node is not in the previous path to the destination. In this case, the operation is the same as QL. The node increments the counter and rebroadcasts the packet if the counter is less than the k threshold or discard the packet if the counter is equal to or exceed the k threshold. On the contrary, if there is an unexpired corresponding entry in the table, the node then compare the number of hops of the entry with the attached number of hops of the packet. If the number of hops of the entry is less than that in the packet, the counter is reset to 0 (Node locality) or remains the same (Path locality). Then, the node replaces the attached number of hops of the packet with the value from the entry and rebroadcasts the packet. Otherwise, the node discards the packet to prevent further rebroadcast backward to the source. These processes are explained in Algorithm 1.

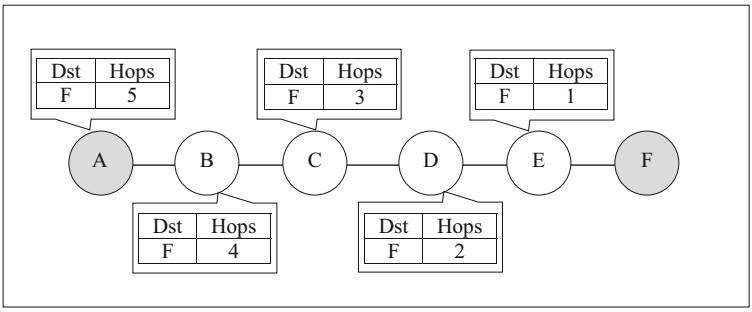
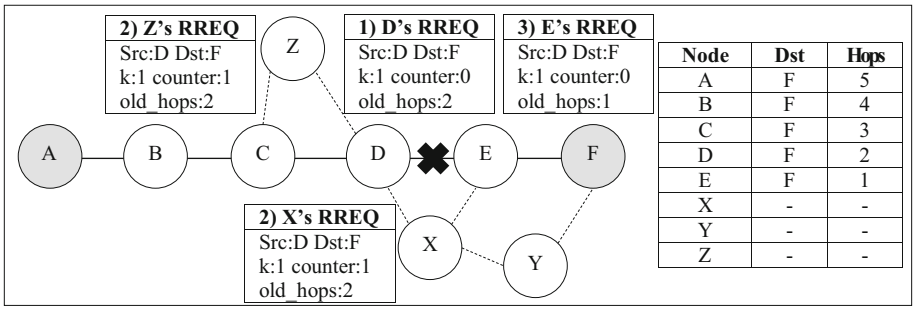


Fig. 3. The recorded information in Ptable after nodes receive a RREP packet



Node	Dst	Hops
A	F	5
B	F	4
C	F	3
D	F	2
E	F	1
X	-	-
Y	-	-
Z	-	-

Fig. 4. The action of nodes when an intermediate node (D) performs a local repair

```
Algorithm 1: Procedure Receive_RREQ (rq)
1:if rq.destination exists in Ptable and
  Ptable.expire_time > CURRENT_TIME then
2:  if Ptable.old_hops < rq.old_hops then
3:    rq.counter = 0 (Only for Node locality approach)
4:    rq.old_hops = Ptable.hops
5:    Rebroadcast rq.
6:  else
7:    Drop rq.
8:  endif
9:else
10: if rq.counter < rq.k_threshold then
11:   rq.counter++
12:   Rebroadcast rq.
13: else
14:   Drop rq.
15: endif
16:endif
```

Figure 4 shows a local repair operation of node D (the *k* threshold is set to 1). The recorded number of hops (variable "old_hops") to the destination F (which is 2 hops in this example) is attached to the originated RREQ packet. When node X and node Z receives the RREQ packet, it increments the counter by one and rebroadcasts the packet because it does not have a record of the corresponding destination and the counter is still less than the *k* threshold. Node C also receives the packet rebroadcasted from node D and latter from node Z. However, node C has a larger number of hops to destination F compared to that in the packet. Node C will eventually drop the packet. Node C also discards the duplicated RREQ packet from node Z. When node E receives the RREQ packet from node X, since it has an unexpired record of the corresponding destination, it then compares the recorded number of hops with the value that in the RREQ packet. Since the recorded number of hops is less than that in the packet, the node then modifies the counter (depend on which approach, Path locality or Node locality, is

selected), updates the attached number of hops to 1 and rebroadcast the packet. The Node locality approach is used in this example so that the counter of the packet is reset to 0 by node E. On the contrary, node Y will drop the RREQ packet from node X since it does not have the corresponding destination in its Ptable and the counter in the received packet is equals to the k threshold. Finally, the RREQ packet will arrive at destination F. This method can avoid flooding-based local repair, which occurs in QL.

In order to reduce storage utilization, the information in Ptable can be appended to the routing table of AODV. However, the previous number of hops should be recorded separately from the hop count field of AODV and must not be modified when the route is invalidated from route breakage or a RERR packet.

4 Performance Evaluation

In this section, the performance of routing protocols is evaluated by using Network Simulator 2 (NS-2). The simulation time is set to 1000 s. A warm-up period is also performed for 1000 s before the simulation begin sand the results are averaged from 20 experiment trials. Each node is equipped with an omnidirectional antenna providing transmission range of 100 m. We use IEEE 802.11 Distributed Coordination Function (DCF) as MAC layer with a fixed bit rate of 2 Mbps. Nodes move according to the Random Waypoint model in an area of 300×600 m^2. Pause time is set to zero. There are 20 sources sending 512 byte Constant Bit Rate (CBR) packets at the rate of 4 packets per second. The node's velocity is randomly selected from a uniform distribution between $v \pm 10\%$ m/s where v is set to 5, 10, 15, 20, 25, and 30. The following metrics are used to evaluate the network performance:

- Normalized routing overhead: the fraction of the total number of routing packets (RREQ, RREP, and RERR) and the total number of delivered data packets. Since the size of the RREQ packet of QL and our proposed protocol is larger than that in AODV, we also show the normalized routing overhead as the fraction of the total size of routing packets and the total size of delivered data packet.
- MAC collision rate: the average number of dropped frames at MAC layer due to a collision per second.
- Packet delivery ratio: the fraction of the number of data packets successfully delivered to destinations and the number of data packets sent from sources.
- End-to-end delay: the average amount of time which used to deliver data packets to the destination successfully.

We compare the performance of our protocol with AODV-ERS [5] and Query Localization (QL) [6]. QL and our protocol are implemented based on AODV library of NS-2. For QL, due to the limited space, we show only the result of Node locality approach. The k threshold is initially set to 1 as same as in [6]. The k threshold is incremented by one if route discovery fails and is decremented by one when route discovery success. The expiration time of each source-destination pair is initially set to 10 s. For our proposed protocol, the parameters are configured as same as in QL.

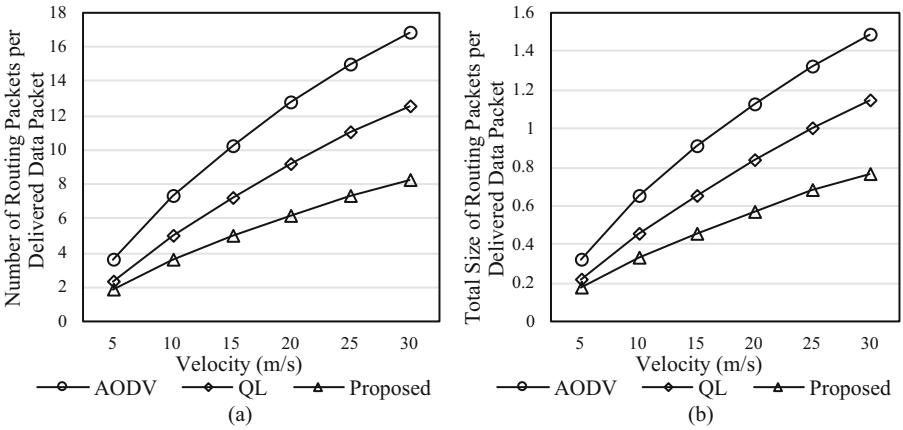


Fig. 5. (a) The number of routing packets, (b) The total size of routing packets

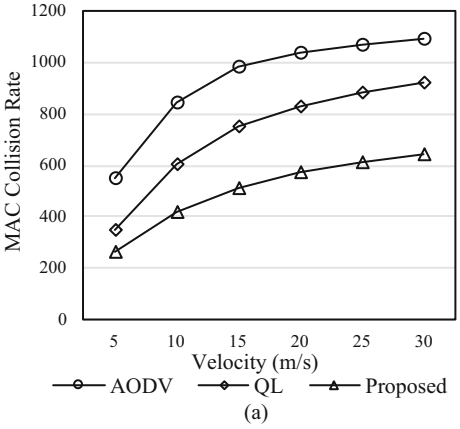


Fig. 6. MAC collision rate

Figure 5 shows the normalized routing overhead in a different node's velocity. The amount of routing overhead increases when the node's velocity increases since routes between sources and destinations break more frequently. Figure 5a shows the normalized routing overhead as the number of routing packets per delivered data packet. AODVhas the highest routing overhead since all RREQ are flooded throughout all node around sources. When paths break more frequently in the high mobility scenarios, routing overhead of AODV raises up rapidly. QL reduces routing overhead by performing directional flooding based on the previous old route instead of omnidirectional flooding. This greatly decreases the number of RREQ packets, results in less network-wide flooding. Our proposed protocol can further alleviate routing overhead more than QL. We enhance the route repair of QL by using a destination address and a previous number of hops instead of using a source-destination pair. This helps local

repair performs more successfully. The previous number of hops to the destination is also used in the route discovery in order to prevent RREQ from being rebroadcasted backward to the source. Compared to AODV, our method saves approximately 50% routing packets and 28% when compared to QL, while supporting high packet delivery ratio (Fig. 7a). Figure 5b shows the normalized routing overhead as the fraction of the total size of routing packets and the total size of delivered data packet. Even the size of the RREQ packet in QL and proposed protocol is gradually larger than the standard RREQ packet in AODV, the result shows that the total size of transmitted routing packets is still substantially less than that in AODV.

Figure 6 shows the MAC collision rate with a varied node's velocity. The collision increases when the node's velocity increases. AODV has the highest MAC collision rate in all node's velocity since there are many simultaneous transmission activities, which generate high routing overhead (Fig. 5). When the amount of routing overhead is lowered in QL, the MAC collision rate is also reduced. Our method can achieve and maintain low MAC collision rate because of the lowest generated routing overhead compared to other protocols. Comparing to AODV, the proposed protocol can reduce MAC collision rate up to 41% and up to 30% when compared to QL.

Figure 7a shows the packet delivery ratio with an increasing velocity. The delivery ratio decreases when the velocity increases. Since AODV generates a high amount of routing overhead and MAC collision, this leads to high packet loss. QL can reduce unnecessary rebroadcast so that the packet can be delivered more successfully. Since our protocol can further reduce unnecessary routing packets, we can decrease the number of packet loss from frame collision, results in high packet delivery ratio. Our proposed protocol can deliver data packets upto 6% higher than QL and 12% when compared to AODV.

Figure 7b shows the average end-to-end delay with different node's velocity. The end-to-end delay increases when the velocity increases. The effect of high routing overhead causes high latency to forward data packet in AODV. By alleviating routing overhead and network congestion, our work can achieve a low end-to-end delay. However, in very high mobility scenarios (25 to 30 m/s), all protocols have a similar delay since route breakage occurs frequently and cause route hardly to be established.

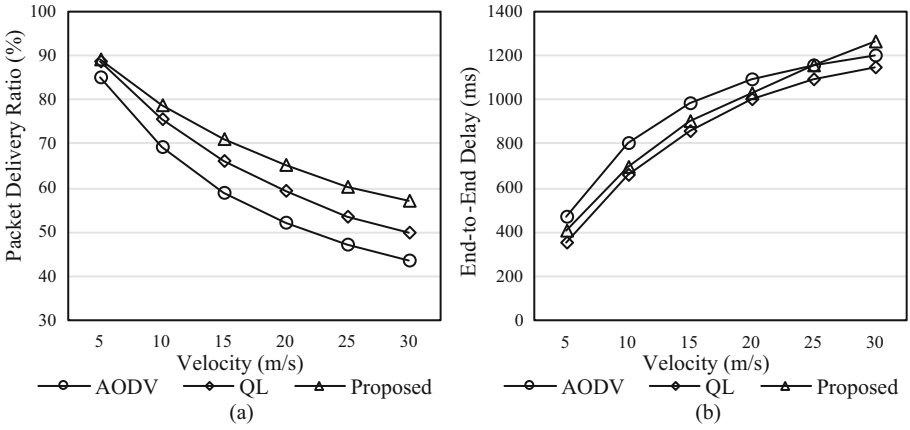


Fig. 7. (a) Packet delivery ratio, (b) End-to-end delay

5 Conclusion

In this paper, we proposed an extension to improve the performance of Query Localization [6]. Instead of being strict with the source-destination pair, our protocol allows any node in the previous active route with the lower number of hops to rebroadcast the RREQ packet. The packet will be propagated more efficiently in the routing operation. As evidenced by the simulation results, our method reduces routing overhead, while maintaining high packet delivery ratio.

Acknowledgements. This research project is approved by National Research Council of Thailand (NRCT) and is financially supported by King Mongkut's Institute of Technology Ladkrabang (KMITL).

References

1. Johnson, D., Hu, Y., Maltz, D.: The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4.RFC 4728 (2007)
2. Perkins, C., Belding-Royer, E., Das, S.: Ad Hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (2003)
3. Perkins, C.E., Bhagwat, P.: Highly dynamic destination-sequenced distance vector (DSDV) for mobile computers. In: Conference on Communications Architecture, Protocols and Applications (SIGCOMM), pp. 234–244. ACM, London (1994)
4. Clausen, T., Jacquet, P.: Optimized Link State Routing Protocol (OLSR). RFC 3626 (2003)
5. Lee, S.-J., Belding-Royer, E.M., Perkins, C.E.: Scalability study of the ad hoc on-demand distance vector routing protocol. Int. J. Netw. Manag. **13**, 97–114 (2003)
6. Castañeda, R., Das, S.R., Marina, M.K.: Query localization techniques for on-demand routing protocols in ad hoc networks. Wirel. Netw. **8**, 137–151 (2002)
7. Ochi, Y., Okazaki, T., Kinoshita, K., Tode, H., Murakami K.: An extended AODV routing for reduction of control messages in ad hoc networks. In: 4th IEEE Consumer Communications and Networking Conference, pp. 74–78. IEEE, Nevada (2007)
8. Narongkhachavana, W., Choksatid, T., Prabhavat, S.: An efficient message flooding scheme in delay-tolerant networks. In: 7th International Conference on Information Technology and Electrical Engineering, pp. 295–299. IEEE, Chiang Mai (2015)
9. Choksatid, T., Narongkhachavana, W., Prabhavat, S.: An efficient spreading epidemic routing for delay-tolerant network. In: 13th IEEE Annual Consumer Communications and Networking Conference, pp. 473–476. IEEE, Nevada (2016)
10. Thongthavorn, T., Narongkhachavana, W., Prabhavat, S.: A study on overhead reduction for GPS-assisted mobile ad-hoc networks. In: 2014 IEEE Region 10 Conference, pp. 1–5. IEEE, Bangkok (2014)
11. Thongthavorn, T., Narongkhachavana, W., Prabhavat, S.: Overhead reduction of location-aided gateway discovery protocols. In: 8th International Conference on Information Technology and Electrical Engineering, pp. 1–6. IEEE, Yogyakarta (2016)

A Dynamic Routing for Load Distribution in Mobile Ad-Hoc Network

Metha Rungtaveesak(✉), Noppawit Chartkajekaew, Thananop Thongthavorn, Worrawat Narongkhachavana, and Sumet Prabhavat

Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
metha.boat@gmail.com, noppawit_ice@hotmail.com,
{thananop, sumet}@it.kmitl.ac.th, 57606007@kmitl.ac.th

Abstract. Mobile Ad Hoc Network is a group of mobile nodes which can communicate with each other directly without infrastructure device. Most conventional protocols do not consider energy on route discovery. Route breaks may occur frequently since the energy of some intermediate nodes exhausts. This interrupts the packet forwarding. In this paper, we propose load distribution algorithm based on remaining energy of nodes to prolong network lifetime and load balancing. Simulation results show that our work can increase the number of remaining nodes in the network while maintaining high packet delivery ratio.

Keywords: MANET · Availability · Load distribution · AODV · Network lifetime

1 Introduction

Mobile Ad-Hoc Network (MANET) [1] is self-organizing nodes with a free movement, limited energy and based on mobile devices. MANET is an infrastructure-less net-work. Mobile node can communicate without any fixed infrastructure and centralized management. Communication between nodes in MANET relies on routing protocol. Most of routing protocol consider shortest path to select a routing path. In AODV [2], a source node selects path from route reply that come back first. However, AODV concerns nothing about an energy of the node. Even if a node has low energy, it still forward a packet until it runs out of the power. A mobile node then turns to 'dead' when there is no energy left. Since dead node cannot participate in the network, packets in the node will be dropped and source node will find the new path. In case of having too many source nodes sending a data packet through this dead node, the network will have a large amount of routing request from re-routing process, leading to network congestion.

Many proposed protocol attempt to decrease overhead from routing process such as using GPS (Global Positioning System) [3–5] and Aggregation technique [6, 7]. However, our work focused on availability improvement of a mobile ad-hoc network. Herein, we define the term 'availability' as the ability of every node in network is able to communicate with each other by using some communication path [8]. In Fig. 1,

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_22

node C in Fig. 1A is the common forwarding node to its destination. Therefore, node C will likely to suffer from high load then it will run out of power first. After that, node A is unable to connect with its destination node. Therefore, the network loses its availability. In contrast, in Fig. 1B, node B does not use the shortest path (node C) but it chooses node E instead. In this case, node C will not lose as much power as shown in Fig. 1A. This network will have more availability than Fig. 1A. Therefore, the Load-Balancing is important in MANET to share the load from one node to the others for saving the forwarding node that it is only one forwarding path to destination.

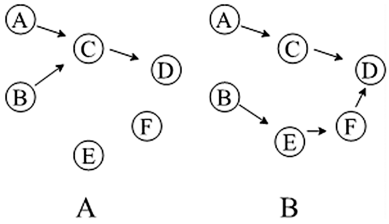


Fig. 1. Comparison between 2 groups of MANET that have different route selection algorithm.

2 Related Works

A routing protocol can be categorized into two groups: Proactive protocol and Reactive protocol [1]. Proactive or table driven routing protocols is based on Bellman-Ford algorithm, their routing methods are distance vector. A node will build a routing table before sending a packet. The table is always up-to-date when topology change occurred in the network. A disadvantage is an 'always up-to-date table'. It means more power is being used in order to make the routing information as fresh as possible. On the other hand, the reactive protocol builds the path only when a node needs to send a packet. Reactive protocol consumes less energy to establish the path when compared with proactive protocol but the disadvantage is it takes more time in routing process.

In mobile ad hoc network, some parts in the network that are farther away from active route may not be utilized while active nodes are suffering with high load. Therefore, a number of load balancing algorithms for mobile ad hoc network has been proposed, which offer the ability to divert the traffic from heavily loaded area to other areas that may lightly loaded or idle.

2.1 AODV

Ad-hoc On-demand Distance Vector (AODV) [2] is reactive routing protocol. It discovers routes only when needed. AODV uses routing tables to store routing information. When source node needs a route to its destination, it will look up for the routing path in its own routing table. If there is no established path, it will broadcast RREQ (route request) message to all other nodes around itself. AODV also uses sequence number to indicate the freshness of the information in both sending and

receiving. The higher sequence number means the information is newer. When RREQ arrived either the destination node or an intermediate node that has "fresh enough" routing information to the destination, that node will broadcast a RREP (route reply) packet back to the source. The RREP that arrive at source node first will be the path that selected as a communication path, while the rest will be discarded. The advantage is paths are created when needed and disadvantage: Path are selected without considering with remaining energy.

2.2 AOMDV

Ad-hoc on demand Multipath Distance Vector [9] is an improved version of AODV protocol which offers multiple path with loop-free [10] and link-disjoint [11] features. This means AOMDV can discover more than one path in route discovery process. An "advertised hop count", the highest hop count from source to destination, is attached to RREP by destination node to prevent a loop from forming up in the network. If a node receives a RREP with a hop count higher than advertised hop count, that alternative path will be discarded. In a route maintaining procedure, each update also provides loop-free and link-disjoint as well. The advantage is decreased the chance of path breakage with reserve route and disadvantage: Increased energy usage of nodes to provide reserve route.

2.3 CMMBCR

Conditional max-min battery capacity routing [12] is which combine Minimal total power routing (MTPR) and Min-Max battery cost routing (MMBCR) to increase the lifetime of the network. The node uses MTPR if its residual energy is greater than threshold. Otherwise, it uses MMBCR. This protocol chooses path by considering remaining energy of the node. It shares load form shortest path and increase availability of network.

The algorithm above explained as follows:

$R_j^c = \min c_i^t$, for $i \in$ route j

$R_j^c \geq \gamma$, for any $\text{route}_j \in A$

R_j is the lowest battery in the path to destination in path j at time t

γ is percent of threshold

A is all possible path to destination.

Minimal Total Power Routing (MTPR). [12] is routing metrics that determines the path from the summation of the remaining energy of each node. The summation of energy is attached with RREP. Then, source node chooses the paths that have minimum remaining energy from all possible path. Disadvantages of this process are it does not consider the remaining energy of each node in the path. Therefore, there is a chance that the selected path is formed with a node that has low energy resulting in path breakage because the node runs out of energy.

Minimum Battery Cost Routing (MBCR). [13] is routing metrics that determines the path by choosing the path that have minimum cost of all possible path. The cost can be calculated by using:

$$f_i(E_i) = \frac{1}{E_i} \tag{1}$$

E_i is energy of node n_i in percent

If node has high energy, the cost of the path is low. On the other hand, if node has low energy, the cost will be high. However, MBCR has a problem when the path has very low energy node mixed with very high-energy node but the sum of the path cost is still lower than another path cost, which supposed to be the selected as shown in Fig. 2.

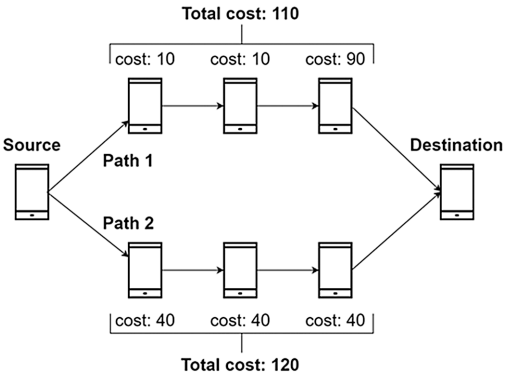


Fig. 2. Energy problem of MBCR.

Min-Max Battery Cost Routing (MMBCR) is developed from MBCR to fix problem in Fig. 2. First, it chooses the lowest remaining energy node (highest cost) from every node in the path as the path cost. Then it picks the lowest cost path from all possible paths. If the cost is equal, it will work the same way as MBCR. In Fig. 2, MMBCR will select path number two. Because the highest cost of path number two is 40, which is lower than 90 from path number one.

Advantage of CMMBCR is increase network lifetime and availability of node by choosing path with energy and disadvantage: Remaining energy of node in path is not balanced well.

3 Proposed Protocol

The conventional routing protocols discussed do not balance the load dynamically. AOMDV selects the lowest hop count routes to send packet from source to destination and keep higher hop count routes as a backup path. The problem of multipath is it uses more power than single path because a destination node needs to respond every RREQ packet with RREP as a candidate path which later selected by source node. Therefore,

AOMDV will run out of power faster than AODV. CMMBCR select route by considering energy of a node but this metric only applied to a routing process. When a node has low energy, it still forward packets until run out of power. The availability of network will decrease. On the other hand, the proposed protocol select the path by considering node's energy on both routing and data transmission process to maintains the most effective path in the network (Fig. 3).

Type	Special-Route Request	Hop Count
Broadcast ID		
Destination IP Address		
Destination Sequence Number		
Source IP address		

Fig. 3. Structure of special RREQ packet

3.1 Node Classification

Node is classified into three states. Green, Yellow and Red according to remaining energy of node

- Green state is the first state, where the power of mobile node is more than 50% of initial energy. This state means the node does not have energy problem. Node will forward all RREQ packet and data packet.
- Yellow state is the second state where the power of mobile node is between 30% and 50% of initial energy. This state means the node is going to have an energy issue. The node will discard the first RREQ it received. However, when it receives another RREQ from the same source node, it will forward RREQ with new header packet called Special-RREQ. In case of the data packets, node will forward all data packet.
- Red state is the last state where the power of mobile node is less than 30% of initial energy. This state means a node has critical energy problem. The node will start forwarding RREQ packet when it received RREQ 3 times from the same source. The forwarded RREQ also has special field in header as well as yellow state. Moreover, node will send a Lower Power Notification (LPN) packet back to the source node when the number of data packets, which already forwarded, reaches data packet limit.

The flowchart of node classification and RREQ process is shown in Fig. 4(A).

3.2 New Field in Header

New packet header in RREQ packet called Special-RREQ. It is a field in header of RREQ packet updated by yellow and red node only. When node receives RREQ packet with special-RREQ, it will forward the packet immediately without considering its current state. The purpose of special-RREQ is to ensure that a packet will make its way to the destination without being dropped by those yellow-red stated nodes in the path.

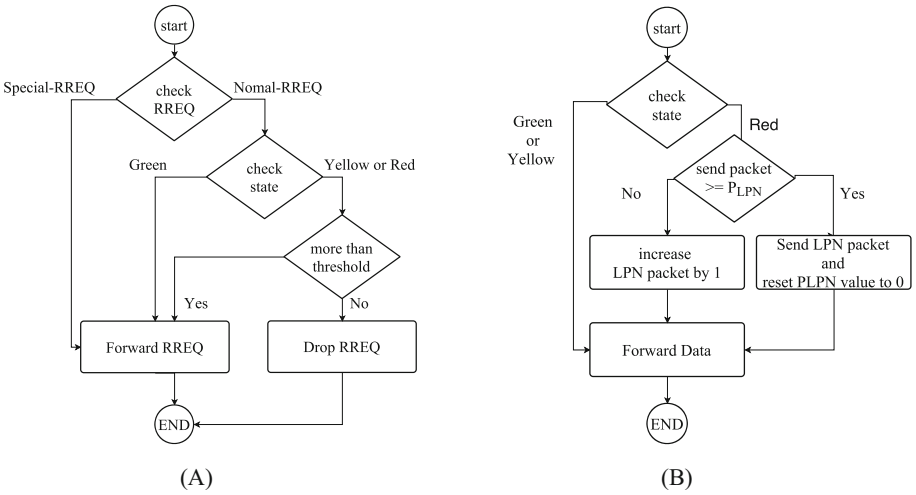


Fig. 4. (A) Flowchart of RREQ sending process, (B) Flowchart of data sending process

3.3 Notification Packet to Change the Direction of the Path

Lower Power Notification (LPN) is a packet that sent by intermediate node back to the source, telling the source to initiate a new route discovery process. A frequency of the LPN packet generation is calculated by using initial energy, current energy, and data packet limit. The data packet limit is the number of a data packet that can be sent before next LPN packet will be generated. when a node changes to red state, it will keeps forwarding data packet until it reaches data packet threshold which can be calculated from Eq. (2). After that, node will reset P_LPN counter to zero then send LPN packet back to the source node. The P_LPN value is depends on current energy of individual node. In forwarding node disjoint condition, a source node that send higher data packet will have more chance to be notified by LPN packet. The process flowchart is presented in Fig. 4B.

$$P_{LPN} = \begin{cases} \frac{E_c}{E_r} \times P_{Max}; & E_c \leq E_{red} \\ 0; & E_c > E_{red} \end{cases} \quad (2)$$

E_c = The ratio of current energy and initial energy.
E_r = the constant value which equal to energy percentage of a node at the moment when it turn to red state.
P_{Max} = the maximum number of data packets that can be sent before next LPN packet will be generate.
P_{LPN} = the number of data packets that can be sent before next LPN packet will be generate varies from node current energy.

4 Simulation Results and Analysis

In this simulation, the mobility model is Grid Mobility Model [14]. We place all nodes on a grid model to determine if the node is working as we programmed and to verify the path from source to destination is same as we expected. Simulation parameters are shown in Table 1.

Table 1. Simulation Parameters

Routing protocol	AODV, AOMDV, AOMDV-CMMBCR
Simulator	NS-2.35
Number of nodes	100
Dimension of simulated area	1000 × 1000
Initial energy	20 J
Simulation time	2000 s
Packet size	512 bytes
Source data pattern	4 packets/s
Traffic type	CBR

Figure 5 shows the performance of load distribution. Node's remaining energy is represented by color range from white (100%) to black (0%). If the network space contains a lot of black node, it means that the protocol cannot distribute the traffic load properly. The proposed protocol have less black node than other protocol but filled with many grey nodes instead. It can be concluded that the proposed protocol can shares the load to other part of network better than other protocols which makes the network have more availability.

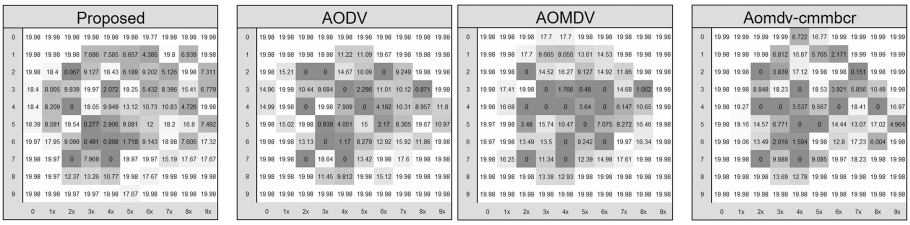


Fig. 5. Load distribution performance of proposed protocol and other related protocol.

4.1 Performance Evaluation

We evaluate routing protocols by using these following performance metrics:

- **Network Lifetime.** This metric represents a number of alive Node at particular time instant. Node is alive if its energy is more than 0.

- **First Node Dead.** This metric represents when the first node in the network run out of energy.
- **Energy Variance.** This metric represents energy distribution performance of the network.
- **Packet Delivery Ratio.** This metric represents success rate of sending packet from source node to destination node.
- **Average End-to-End Delay.** This metric represents average time that packet take to travel from source to destination node.

4.2 Simulation Results

Figure 6 shows number of remaining node in the topology at the end of simulation. The proposed protocol has more nodes left in the network than other protocols because the LPN packet that help source node re-routing the path to reduce usage of low energy node. More nodes remain in the network means more availability because a node will have more path to chooses for sending a data.

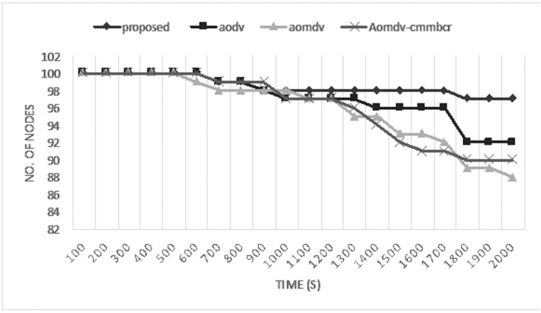


Fig. 6. Number of remaining nodes in the network at the end of simulation.

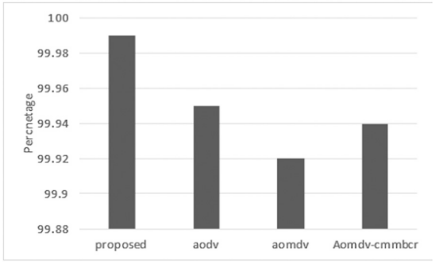


Fig. 7. Success rate of packet delivery in the network.

Figure 7 shows the packet delivery ratio of the network. It shows that all protocol can send packet at almost the same success rate. This means the proposed protocol can shares the load, make more availability to the network while it does not have any side effect to others performance of the network.

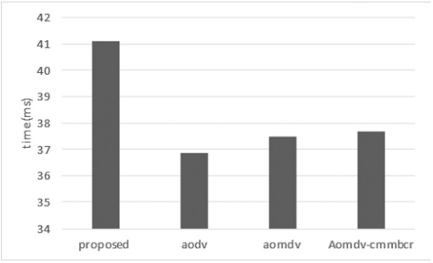


Fig. 8. Average end-to-end delay (ms)

Figure 8 shows the delay that the packet takes in order to travel from source to destination node in millisecond. The graph show that proposed protocol has longer delay than other protocol. Because of LPN packet, when a node has low power, it will send LPN packet to the source. Then, the sources rebroadcast a RREQ to find the new path which resulting in longer delay than the others.

5 Conclusion

In this paper, we propose a load-balancing routing protocol to improve the availability of MANETs. We can increase network lifetime and number of remaining node. Moreover, the mechanism of proposed protocol can be sent packet with same packet delivery ratio as others protocol. However, proposed protocol still has some disadvantages, such as the average end-to-end delay increase because the active path is not the shortest path. However, this defect is not effects on another improved performance metrics.

Acknowledgements. This research project is approved by National Research Council of Thailand (NRCT) and is financially supported by King Mongkut's Institute of Technology Ladkrabang (KMITL).

References

1. Macker, J.: Mobile ad hoc networking (MANET): routing protocol performance issues and evaluation considerations (1999)
2. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc on-demand distance vector (AODV) routing. No. RFC 3561 (2003)
3. Abrougui, K., Boukerche, A., Pazzi, R.W.N.: Location-aided gateway advertisement and discovery protocol for VANets. IEEE Trans. Veh. Technol. **59**(8), 3843–3858 (2010)
4. Thongthavorn, T., Narongkhachavana, W., Prabhavat, S.: A study on overhead reduction for GPS-assisted mobile ad-hoc networks. In: 2014 IEEE Region 10 Conference, TENCON 2014, Bangkok, pp. 1–5 (2014)

5. Thongthavorn, T., Narongkhachavana, W., Prabhavat, S.: Overhead reduction of location-aided gateway discovery protocols. In: 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, pp. 1–6 (2016)
6. Narongkhachavana, W., Choksatid, T., Prabhavat, S.: An efficient message flooding scheme in delay-tolerant networks. In: 2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE), Chiang Mai, pp. 295–299 (2015)
7. Choksatid, T., Narongkhachavana, W., Prabhavat, S.: An efficient spreading epidemic routing for delay-tolerant network. In: 2016 13th IEEE Annual Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, pp. 473–476 (2016)
8. Severinghaus, R., Tummala, M., McEachen, J.: Availability of ad hoc wireless networks of unmanned ground vehicles with group mobility. In: 2013 46th Hawaii International Conference on System Sciences (HICSS), pp. 5097–5105. IEEE (2013)
9. Rana, G., Ballav, B., Pattanayak, B.K.: Performance analysis of routing protocols in mobile ad hoc network. In: 2015 International Conference on Information Technology (ICIT), pp. 65–70. IEEE (2015)
10. Puri, S., Devene, S.R.: Congestion avoidance and load balancing in AODV-multipath using queue length. In: 2009 2nd International Conference on Emerging Trends in Engineering and Technology (ICETET), pp. 1138–1142. IEEE (2009)
11. Das, I., Lobiyal, D.K., Katti, C.P.: An analysis of link disjoint and node disjoint multipath routing for mobile ad hoc network. Int. J. Comput. Netw. Inf. Secur. **8**(3), 52 (2016)
12. Parissidis, G., Karaliopoulos, M., Baumann, R., Spyropoulos, T., Plattner, B.: Routing metrics for wireless mesh networks. In: Guide to Wireless Mesh Networks, pp. 199–230. Springer, London (2009)
13. Toh, C.K.: Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks. IEEE Commun. Mag. **39**(6), 138–147 (2001)
14. Chen, J.K., Chen, C., Jan, R.H., Li, H.H.: Expected link life time analysis in MANET under Manhattan grid mobility model. In: Proceedings of the 11th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, pp. 162–168. ACM (2008)

A Comparative Study of IXP in Europe and US from a Complex Network Perspective

Zhongyan Fan[1], Wallace K.S. Tang[1(✉)], Dong Lin[2], and Doujie Li[1]

[1] Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong
zyfanl99l@gmail.com, eekstang@cityu.edu.hk, lidoujie@gmail.com

[2] Future Network Theory Lab, Huawei Research Institute, Huawei Technology Co. Ltd, Kwai Chung, New Territories, Hong Kong
lin.dong@huawei.com

Abstract. A significant change in the Internet ecology due to the launch of Internet exchange points (IXPs) has been witnessed in recent years. The traffic exchange services of IXP make the traffic delivery between autonomous systems (ASes) not only lower in cost but also higher in efficiency. However, the business models of IXP are different in various regions, and hence their impacts would be different. In this paper, we investigate and compare their impacts in Europe and in US. The study is conducted, firstly, based on a bi-layered network framework which can characterize AS-IXP topologies properly. Secondly, the potential usage of IXP in routing service is investigated by using control exchange point (CXP). Our simulation results show that IXP industry flourishes better in Europe than in US. In addition, the IXP has high potential to be involved in routing service as it is effective and stable.

Keywords: Bi-layered network · Complex network analysis · Internet · Internet exchange point

1 Introduction

The Internet is a global system connecting billions of devices over the world via IP networks, managed by tens of thousands of autonomous systems (ASes) separately. The ASes and their mutual connections form the AS-level ecosystem, and can be deemed as a logical fabric of the Internet in a macroscopic view.

As a complex network, the scale-free property of AS-level topology was firstly discovered in [1]. Since then, many power-law models have been suggested to model the Internet topological features and characteristics. Examples include BA model [2], HOT model [3], PFP model [4], MLW model [5], Einsteinian model [6], just to name a few. The power-law feature in the Internet also leads to studies of other networking issues, such as the robustness [7], packet routing [8], traffic congestion [9], etc.

In contrast, the use of complex network analysis for Internet exchange point (IXP) is relatively little (e.g. [10, 11]), despite that it becomes more and more important. IXP is a network facility that enables the interconnection between multiple

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_23

ASes, primarily for the purpose of exchanging traffics. It not only reduces the operational costs of Internet Service Provider (ISP), but also improves the quality of service (QoS) [12]. Recent work also demonstrated the contributions of IXP in Internet flattening [11] and its role in new applications, such as software-defined network (SDN) [13, 14].

However, it is interesting to point out that the business model of IXP varies in different regions. In Europe, IXPs are generally public-facing infrastructures, mainly focusing on public multi-lateral peering business, which attracts a lot of ISPs due to the cheap price. As a consequence, the IXP business in Europe has flourished beyond other regions. On the other hand, in US, big IXPs are usually owned by commercial business entities that primarily provide private bi-lateral peering services.

The prime objective of this paper is to compare the impacts of IXPs in Europe and in US. Instead of focusing on the business models, the study is carried out from a complex network perspective, based on two scenarios: (i) Bi-layered network framework [11]; and (ii) CXP-based routing service [15]. As explained in [11], the bi-layered network is useful to investigate the interplay of IXPs and ASes, and to divulge the characteristics. While the CXP-based routing service becomes more feasible due to the fast development of IXP associations, it will be interesting to look into how this potential service would perform in Europe and US, respectively.

2 Overview of IXP

Conventionally, two ASes are connected by establishing bi-lateral business agreement, and it can be described by a dedicated link as shown in Fig. 1(a). However, dedicated link is inefficient and expensive. It imposes extremely high burden to ISPs, especially the small ones. On the other hand, IXP provides an alternative connection method in the inter-domain. It supports both bi-lateral peering and multi-lateral peering [16]. A bi-lateral peering via IXP only involves two ASes, which is similar to the dedicated link connection, except that IXP fabric is adopted. Multi-lateral peerings in IXP are technically supported by Route Server (RS), which directly reflects the BGP routes collected from participants to each other (See Fig. 1(b)). The RS participants of an IXP can be deemed as directly connected in logical layer [16]. Since most of IXP members now opt for being RS participants, it is assumed in our study that all IXP members of the same IXP are connected with each other either by the RS service or by bi-lateral peering.

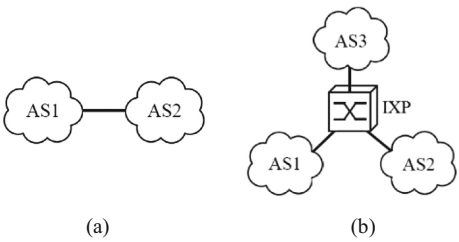


Fig. 1. (a) Dedicated link for AS-AS; (b) IXP interconnection

The recent growth of IXPs, both in their number and the number of their members, has greatly evolved the Internet ecology. Nowadays, there are more than 400 IXPs, serving about 40% ASes over 160 countries, managing a significant portion of traffics. IXPs bring significant benefits in different aspects, including cost reduction by shifting global transit to local transit, QoS improvement by reducing path latencies and packet loss, promotion of local contents, etc. [12–14].

3 Comparison of IXP Impacts Based on Bi-layered Network

3.1 Data Sources for the Study

The data used in our study are summarized as below.

- AS-level Internet Topology: The data set available in [17] is used to construct the AS-level topology. The original data is in a monthly basis, and they are merged for the whole year to build a relatively complete AS-level Internet topology.
- IXP Membership: Following the data collection methods proposed in [18], a series of IXP data sets is extracted from public databases and collected by several technical methods, including traceroute and target source routing.
- AS Location: The location of AS is obtained by mapping the AS number in the IP2LOCATION data set. It is remarked that an AS may contain different IP addresses registered in multiple areas, and hence an AS can map to both Europe and US topologies if it has some IP addresses registered in these regions.

Table 1 tabulates the basic statistics of the topological data. Although data incompleteness is inevitable, it can be observed that the obtained numbers of Internet elements (ASes and IXPs) and their interconnections (dedicated links and memberships) are sufficiently large. Thus, the two networks are considered to be representative.

Table 1. Basic statistics of AS-IXP networks in Europe and in US

Regions	#ASes	#IXPs	#IXP members	#AS-AS links	#IXP membership
Europe	19,433	290	10,273	152,359	30,126
US	14,581	284	5,533	58,388	15,789

Remark: An IXP appears in Europe/US topology if some of its members locate in Europe/US.

3.2 Bi-layered Network Framework

To investigate the impacts of IXPs in Europe and in US, the bi-layered network framework proposed in [12] is adopted. We consider each regional network (i.e. Europe and US AS-IXP) as a bi-layered network with two layers called AS layer and IXP layer. Figure 2 demonstrates an example of such a bi-layered network. G_{AS} and G_{IXP} are two undirected graphs defined in these two layers, respectively. We let $G_{AS} = (V_{AS}, E_{AS})$ where V_{AS} is the set of ASes and E_{AS} is the set of interconnections between the ASes. We also define $G_{IXP} = (V_{IXP} \cup V'_{AS}, E_{IXP})$, where is the set of IXPs, V'_{AS} is the

set of the IXP members, and E_{IXP} is the set of links describing the membership of IXP $AS'_i \in V'_{AS}$ can be considered as an image of the corresponding AS_i in V_{AS} The bi-layered network is then defined as $L(G_{AS}, G_{IXP}, E_{\text{int}})$, where E_{int} is the set of virtual links with zero weights, connecting IXP members who are in both layers. It is also remarked that the weight of links in E_{AS} and E_{IXP} are 1 and 0.5, respectively, since IXP members are deemed as directly connected in logical layer.

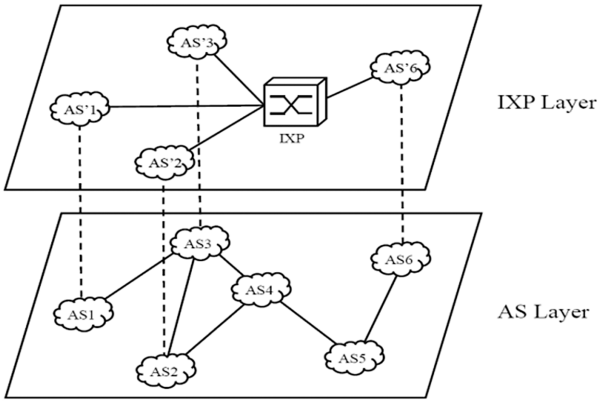


Fig. 2. An example of bi-layered network for AS-IXP topology

3.3 Simulation Results and Analyses

We firstly study the IXP business in Europe and US by performing numerical analysis onto the corresponding bi-layered networks as described in Sect. 3.2. For the ease of referencing, ASL and BiL represent the AS layer network and bi-layered network, respectively. It is noted that ASL corresponds to AS topology without any IXP.

Degree Distribution. Figures 3(a) and (b) depict the degree distributions of ASes in ASL and BiL for Europe and US, respectively. It can be observed that the distributions are similar. As shown in Table 1, the number of IXPs is small compared to the scale of the whole graph. Therefore, the influence of IXP can hardly be observed in this macroscopic view.

Distance Distribution. Figures 4(a) and (b) summarize the AS-AS distance distribution in ASL and BiL for Europe and US, respectively. It can be noticed that the distance between ASes in BiL is significantly shorter than that in ASL, indicating that IXP layer provides plenty of new shortest paths to AS-AS pairs.

Closeness. To illustrate who is benefited from IXP, the closeness of ASes in BiL versus ASL are plotted in Fig. 5. The result clearly shows that the closeness of all ASes are improved by introducing IXP layer (i.e. appearing in the left-hand side of the diagonal), and IXP members can be benefited more.

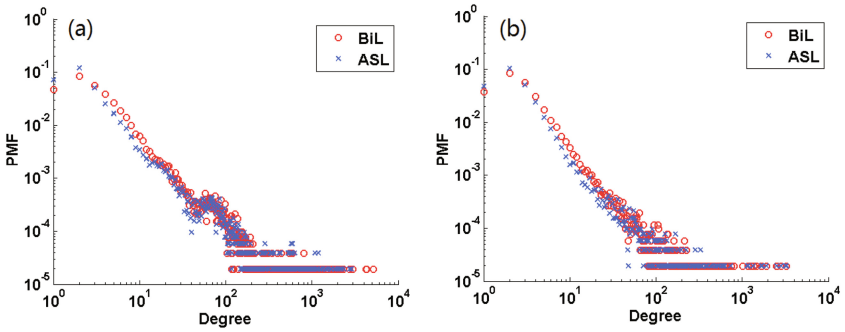


Fig. 3. Degree distributions of ASes in ASL and BiL (a) for Europe and (b) for US

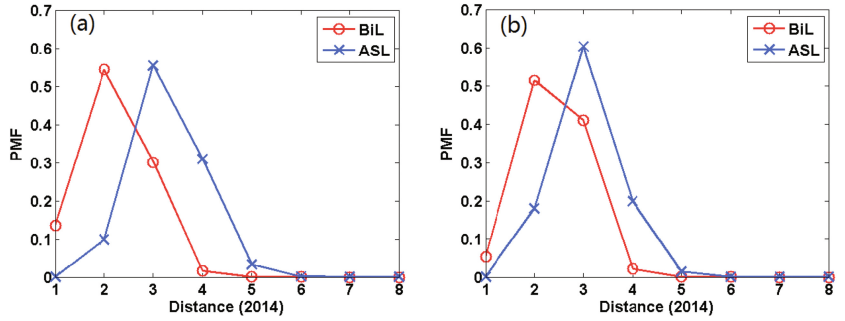


Fig. 4. Distance distribution in ASL and BiL (a) for Europe and (b) for US. The average path length in Europe are 3.28 (ASL) and 2.20 (BiL) and those in US are 3.05 (ASL) and 2.40 (BiL)

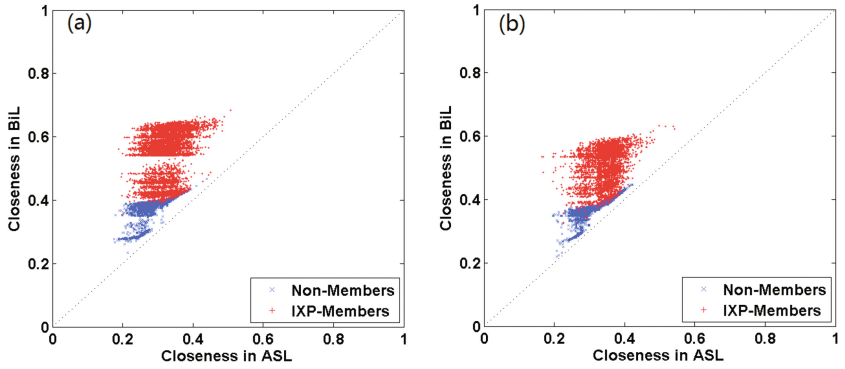


Fig. 5. Closeness distributions of ASes in ASL and BiL (a) for Europe and (b) for US. The average closeness of ASes in Europe are 0.31 (ASL) and 0.47 (BiL) and those in US are 0.33 (ASL) and 0.43 (BiL)

Although the distance distribution and the closeness can show that IXPs would make impacts onto the AS-level topology, they are unable to clearly reveal the involvement of IXPs. Therefore, to have a better understanding, we further consider the bi-layered metrics under the bi-layered network framework [11] as given in the followings.

IXP Layer Coverage. The layer coverage is defined as the ratio of IXP members over all the ASes in G_{AS}.

IXP Layer Facilitation (LF). The IXP LF of a source-destination (s-d) pair represents the ratio of the distance on IXP layer over the total distance. It is computed by:

$$LF(s,d) = \frac{1}{n_{sd}} \sum_{k=1}^{n_{sd}} \frac{\delta_k(s,d)}{D(s,d)} \tag{1}$$

where $s, d \in V_{AS}$ since only AS can be the source or the destination, n_{sd} is the total number of shortest paths from s to d, $\delta_k(s,d)$ is the distance on IXP layer for the k-th shortest path for (s, d) and $D(s, d)$ is the distance between s and d.

Layer Coordination (LC). The LC measures the cooperation intensity between AS layer and IXP layer in delivering traffics. It is defined as:

$$LC(s,d) = \frac{1}{n_{sd}} \sum_{k=1}^{n_{sd}} \frac{t_k(s,d)}{D(s,d)} \tag{2}$$

where $s, d \in V_{AS}$ and $t_k(s,d)$ is the number of transits from AS layer to IXP layer in the k-th shortest path.

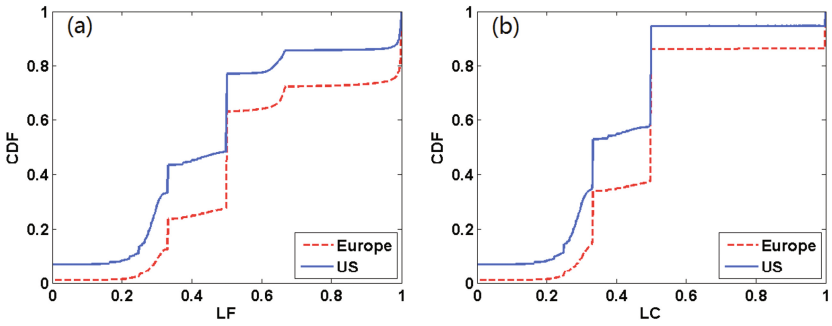


Fig. 6. CDF of (a) IXP LF and (b) LC for Europe and US bi-layered networks

Table 2. Summary of metrics calculated for Europe and US bi-layered networks.

Regions	Layer coverage	Average LF	Average LC
Europe	0.53	0.597	0.500
US	0.38	0.466	0.399

The computed results are given in Table 2. Figure 6(a) depicts the cumulative distribution function (CDF) of LF for the bi-layered networks of Europe and US. It can be observed that a large percentage of *s-d* pairs (99% to Europe topology and 93% to US topology) is affected by the introduction of IXPs when shortest path routing is assumed. Similarly, Fig. 6(b) depicts the CDF of LC. Based on the plot and the average values of LC in Europe (0.500) and US (0.399), a higher cooperation level between AS and IXP layers is observed in Europe. All these imply that IXPs are more influential in Europe than in US.

4 Comparison of Routing Performance Based on CXP

We further consider the routing performance that could possibly be enhanced by the inclusion of IXP. A recently proposed QoS routing service which uses a programmable switch point over IXP, named as Control eXchange Point (CXP), is assumed [15].

CXP can stitch pathlets collected from ISPs and provide QoS guaranteed routing in inter-domain, which is not available using current BGP protocol. In QoS scenario, routing in inter-domain will consider the cost (e.g. bandwidth and latency guarantee) of path rather than just considering hop count [19]. An illustrative example of CXP routing service is depicted in Fig. 7. All IXPs form a multigraph (involving multiple adjacencies) which is managed by a CXP controller, so that QoS optimal path between IXPs can be sought. Thus, the QoS-enabled path can be divided in 3 parts: (i) from source to IXP_a; (ii) from IXP_a to IXP_b; and (iii) from IXP_b to destination.

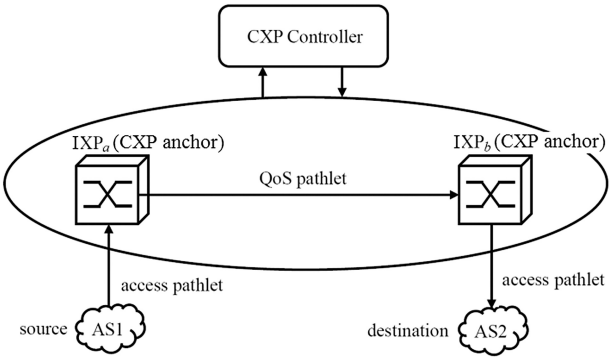


Fig. 7. An example of QoS-enabled routing service based on CXP

4.1 Methodology and Evaluation Metric

We simulate the CXP-based routing services in Europe and US based on the corresponding AS-IXP bi-layered networks. It is assumed that the k IXPs with largest degrees are assigned as CXP anchors. For any source-destination pair, (s, d) in the network, we can compute the QoS optimal path and its cost by the following procedures:

1. A CXP anchor, IXP_a (see Fig. 7), is selected so that the cost from s to IXP_a is the least. If there are more than one possible anchors, randomly pick one.
2. Similarly, a CXP anchor, IXP_b, is selected, which has the least path cost to d.
3. The QoS optimal path from s to d is then composed of 3 sub-paths: from s to IXP_a, from IXP_a to IXP_b, and from IXP_b to d. Each sub-path has the least cost, and their sum gives the cost of the QoS optimal path.

To quantify the performance of CXP routing service, an evaluating metric called path inflation (*PI*) is defined as below:

$$PI = \frac{1}{N_{AS} \times (N_{AS} - 1)} \sum_{s \neq d} \frac{\delta_{CXP}(s, d)}{\delta(s, d)} \tag{3}$$

where $s, d \in V_{AS}$, N_{AS} is the total number of ASes; $\delta(s, d)$ is the minimum cost (via path with total minimum cost) from s to d; $\delta_{CXP}(s, d)$ is the cost from s to d based on CXP routing service. Based on (3), $PI \geq 1$ and the best case is achieved when $PI = 1$

It is remarked that *PI* represents the routing efficiency in terms of QoS metric, reflecting the relative QoS compared to the global optimal QoS.

4.2 Simulation Results and Analysis

The performance of CXP-based routing service onto Europe and US AS-IXP bi-layered networks are compared in two cases described below.

Path Inflation vs. No. of CXP Anchors. Initially, path inflation is investigated with different number of CXP anchors. For simplicity, the costs of all links in E_{AS} are assumed to be the same ($cost_i = 1$ for $Link_i \in E_{AS}$). This condition also meets the scenario of optimal capability allocation [20]. It should be mentioned that, as stated in Sect. 3.2, the costs of IXP memberships are halved ($cost_i = 0.5$ for $Link_i \in E_{IXP}$).

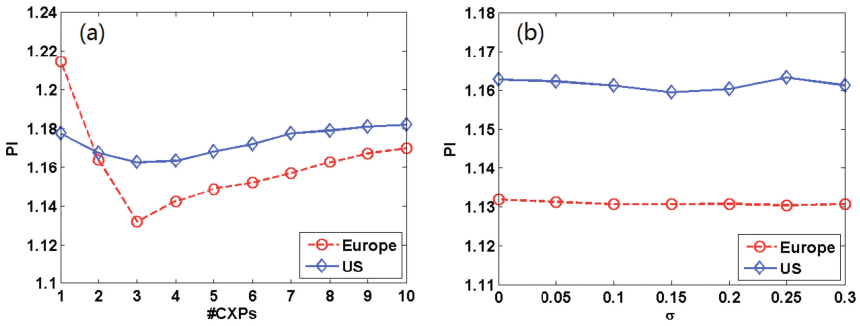


Fig. 8. *PI* vs. (a) no. of CXP anchors and (b) different σ's (averaged by 20 experiments)

As shown in Fig. 8(a), *PI* first decreases when the number of CXP anchors increases from 1 to 3, then it goes up with the increase of the number of CXP anchors.

In Europe or in US network, the best number of CXP anchors is three[1]. One possible reason is that, when the number of CXP anchors increases, some IXPs with lower centrality may also be selected as CXP anchors. In such case, the paths will even be longer. In addition, it is noticed that, in terms of path inflation, CXP-based routing service performs much better in Europe than in US, provided that more than one CXP anchors are employed.

It is remarked that the load of CXP anchor may vary a lot. For example, for the case with 10 CXP anchors in Europe, the largest three anchors serve more than 60% of the ASese, while each other anchor only manages less than 10% of the ASes. Similarly, about 24% ASes are served by a single anchor in US. The unbalanced load implies heavier duty or equivalently higher demand of resources in a few anchors. This issue needs to be managed so that such a routing scheme can be made possible.

Increment of Link Costs. Practically, the link costs vary because the levels of congestions are different. Thus, the cost of $Link_i$ with $Link_i \in E_{AS}$ is now assigned as:

$$cost_i = 1 + \varepsilon_i \tag{4}$$

where ε_i follows the half-normal distributions with some standard deviation σ. Again, the cost of $Link_i$ with $Link_i \in E_{IXP}$ is halved.

The number of CXP anchors is fixed to three which gives the best *PI*. Figure 8(b) depicts the resultant *PI* with different values of σ for Europe and for US AS-IXP networks. As one can see, *PI* maintains stable even when σ increases, implying that CXP routing service works well even under congestions.

As a summary, the QoS path searched by CXP-based routing service is close to the global shortest path that is current unachievable in inter-domain due to the extremely high overhead requirement. Also, it can be concluded that (i) there exists a global optimal value of the number of CXP anchors, which is three for both the Europe and US networks apparently; (ii) Such a performance is quite stable even deviating from the scenario of optimal capability allocation (i.e. $cost_i \neq cost_j$ with $Link_i, Link_j \in E_{AS}$ and $i \neq j$); and (iii) CXP-based routing service has higher potential in Europe than in US, matching the findings in Sect. 3.3.

5 Conclusion

In this work, the impacts of IXP in two regions, Europe and US, are investigated. The study is performed from a complex network perspective by using the bi-layered network framework and by applying CXP-based routing service. Based on the bi-layered network framework, it reveals that IXP has great impacts on Internet topologies both in Europe and US. Moreover, all the metrics indicate that the IXP industry is more flourishing in Europe. It is also noticed that a large percentage (93% of US and 99% of Europe) of source-destination pairs has at least one shortest path going through the IXP layer. It means that IXPs are good choices of routing service providers. This has been

[1] They are AMS-IX, LINX, DE-CIX for Europe; AMS-IX, Equinix-DC, Equinix-CHI for US.

further confirmed by the simulation results, for which the efficiency of CXP-routing service is demonstrated. The results show that IXPs would benefit the Internet in Europe more, if IXP-based routing scheme, such as CXP, is launched. The CXP-based routing service performs well and stable in both Europe and US, but it works better in Europe based on the current development.

Acknowledgements. The work described in this paper was partially sponsored by Huawei Innovation Research Program.

References

1. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. ACM SIGCOMM Comput. Commun. Rev. **29**, 251–262 (1999)
2. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**, 509–512 (1999)
3. Carlson, J.M., Doyle, J.: Highly optimized tolerance: a mechanism for power laws in designed systems. Phys. Rev. E **60**(2B), 1412–1427 (1999)
4. Zhou, S.: Understanding the evolution dynamics of internet topology. Phys. Rev. E **74**, 016124 (2006)
5. Fan, Z., Chen, G., Zhang, Y.: A comprehensive multi-local-world-model for complex networks. Phys. Lett. A **373**, 1601–1605 (2009)
6. Boguna, M., Papadopoulos, F., Krioukov, D.: Sustaining the internet with hyperbolic mapping. Nat. Commun. **1**, 62 (2010). doi:10.1038/ncomms1063
7. Fan, F.: Evaluating the AS-level internet models: beyond topological characteristics. Chin. Phys. B **21**(2), 028902 (2012)
8. Krioukov, D., Claffy, K.C., Fall, K., Brady, A.: On compact routing for the internet. ACM SIGCOMM Comput. Commun. Rev. **37**(3), 43–52 (2007)
9. Yurkevich, I.V., Stepanenko, A.S., Constantinou, C.C., Lerner, I.V.: Fluctuation-driven traffic congestion in a scale-free model of the internet. In: IEEE International Conference on Communications Workshops, pp. 1425–1428 (2013)
10. Gregori, E., Improta, A., Lenzini, L., Orsini, C.: The impact of IXPs on the AS-level topology structure of the internet. Comput. Commun. **34**(1), 68–82 (2011)
11. Fan, Z., Tang, W.K.S.: Unraveling the impacts of IXP in internet ecosystem using bi-layered network. Phys. A **456**, 327–339 (2016)
12. Di Lallo, R.: Is it really worth peering at IXPs? A comparative study. In: IEEE Symposium on Computers and Communication, pp. 421–426 (2016)
13. Chiesa, M., et al.: Inter-domain networking innovation on steroids: empowering IXPs with SDN capabilities. IEEE Commun. Mag. **54**(10), 102–108 (2016)
14. Lapeyrade, R., Bruyere, M., Owezarski, P.: OpenFlow-based migration and management of the TouIX IXP. In: IEEE/IFIP Network Operations and Management Symposium, pp. 1131–1136 (2016)
15. Kotronis, V., et al.: Stitching inter-domain paths over IXPs. In: 2nd Symposium on SDN Research (SOSR) (2016). doi:10.1145/2890955.2890960
16. Giotsas, V., Zhou, S.: Improving the discovery of IXP peering links through passive BGP measurements. In: IEEE International Conference on Computer, Communications Workshop, pp. 121–126 (2013)
17. Internet AS-level Topology Archive. http://irl.cs.ucla.edu/topology/

18. Augustin, B., Krishnamurthy, B., Willinger, W.: IXP mapped? In: 9th ACM SIGCOMM Conference on Internet Measurement Conference, pp. 336–349 (2009)
19. Di Sorte, D., Reali, G.: Minimum price inter-domain routing algorithm. IEEE Commun. Lett. **6**(4), 165–167 (2002)
20. Zhang, G.Q., Zhou, S., Wang, D., Yan, G., Zhang, G.Q.: Enhancing network transmission capacity by efficiently allocating node capability. Phys. A **390**, 387–391 (2011)

Message-Oriented Middleware for System Communication: A Model-Based Approach

Roland Petrasch(✉)

Department of Computer Science, Thammasat University, Bangkok, Thailand
roland.petrasch@cs.tu.ac.th

Abstract. Distributed systems with heterogenous platforms and communication components like IoT devices require message-oriented middleware (MOM). Protocol translation, message model handling, message queueing and conversion, security, transactional consistency, monitoring are examples for the features and aspects of MOM. This paper presents a model-based approach for the development of MOM components using the UML and UML Profiles for MOM and enterprise integration patterns. A model-to-model transformation is used for the preparation of the design model for code generation.

Keywords: Internet-of-Things · Process modeling · UML · Activity diagram · Model-driven-development · Model-driven architecture · Enterprise Integration Patterns · EIP · Message-oriented-Middleware · MoM

1 Introduction and Related Work

In the IoT era, message-oriented middleware (MOM) and machine to machine (M2M) communication play a crucial role: According to Gartner, in 2016, 6.4 billion of connected "things", i.e. IoT devices, worldwide will be in use – a rise of 30% compared to 2015 [1]. To meet the increasing demand for high-quality communication products, substantial research and development efforts have been made leading to significant advances in the discipline of communication technologies during the last decade [2], e.g. in the area of IT-security [3], or practical methodologies [4].

A plurality of IoT relevant norms, industrial (de facto) standards, and products exist, e.g. Advanced Message Queuing Protocol (AMQP) by OASIS [5] and ISO [6], Java Messaging Service by JCP [7] (JMS is part of the Java Enterprise API), and MQ Telemetry Transport (MQTT) by OASIS [8]. The integration of heterogeneous system components using these standards and communication products is possible via a middleware layer. Nowadays, concepts and implementations of MOM are well-accepted and well-established in practice. From a technological standpoint, MOM often comprise components like message and resource broker, transaction manager, persistence service/DBMS, request scheduler etc.

But the IoT paradigm also "raises a number of new challenges in the software engineering domain" [9]. Requirements, e.g. in the form of business or domain processes, need to transformed to architectural concepts and software design specification that take IoT aspects, e.g. mobility of IoT devices, security of IoT data, or performance, into account [10, 11]. New or advanced software architectures, modeling languages,

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_24

and modeling methods are helpful to take these new IoT aspects into account and reduce the complexity of the different components.

This paper addresses some of the challenges: It presents a model-based approach for developing middleware for IoT and Enterprise applications with a focus on enterprise integration patterns (EIP) [12]. Existing approaches focus on certain aspects of middleware, e.g. [13, 14], but they do not take advantage of UML, UML Profile, EIP patterns, and/or M2M transformations.

2 Development of MOM: A Model-Based Approach

2.1 Introduction to Model-Based Software Development

The general pattern for model-based development uses a PIM (platform independent model, e.g. a domain model, that is transformed into a PSM (platform specific model) if the meta-models of the PIM and the PSM are not identical (otherwise it is called a PIM-PIM transformation). The PSM serves as an input (PIM) for the next transformation(s). Transformation types are model-to-model (M2M), model-to-text (M2T), or text-to-model (T2M). Typically, the last step of a forward-engineering approach is the code-generation (model-to-text transformation), so that a PSI (platform specific implementation, i.e. code) is created (s. OMG's Model Driven Architecture [15]). A PIM as an input or source artefact for a transformation is considered platform independent, because it conforms only to its own meta-model (language specification) and is independent from the meta-model of the output or target model (PSM). Therefore, a PIM-PSM transformation involves two meta-models (source and target MM).

As the modeling language, the GPML (general purpose modeling language) UML was chosen [16]. The UML meta-model (language specification) is MOF-conform [17] and provides a lightweight extension mechanism: A UML Profile extends UML meta-model elements by defining a set of Stereotypes so that new domain ortechnological aspects can be included in the modeling. The possibility to create a DSML (domain specific language) for MOM/IOT (heavy-weight approach) was evaluated, but at the end, the existing UML diagrams were considered sufficient.

Figure 1 gives an overview of the approach used in this paper: Domain models, like business processes (UML activity diagrams or BPMN models) and domain class diagrams are used as a starting point (PIM). A UML Profile for MOM/IoT is developed and applied. The system or software architect then prepares the models for the model-to-model transformation: Certain model elements can be marked with Stereotypes. The transformation result is a first software design model (PSM), e.g. class or component diagrams. With the EIP Profile (application of enterprise integration patterns), design models are prepared for code generation (model-to-text transformation).

Different code generators for each target platform exist, e.g. Java Enterprise application, Apache Camel routes for middleware services (MOM). Generated code artefacts (PSI) are used for further manual programming.

The tool chain is based on the Eclipse modeling project [18]: Papyrus (UML Profiling and Modeling), QVTo as the Operational QVT implementation (model-to-model transformations), and Acceleo as the MOFM2T implementation (code generation).

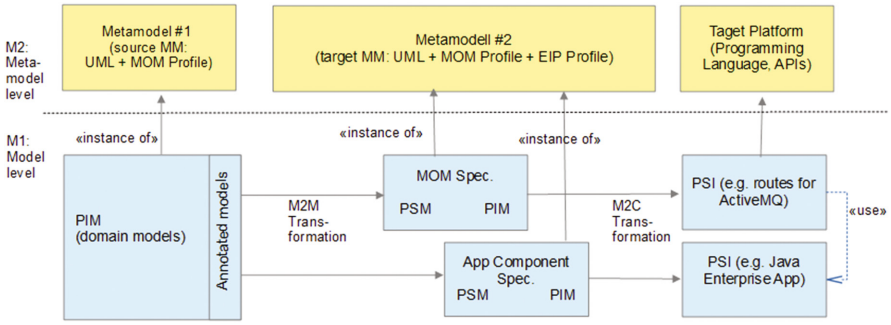


Fig. 1. Model-based software development approach

2.2 Example PIM (Domain Model)

To exemplify the approach, an example scenario from the manufacturing domain is used: the quality control (QC) of a production material. Figure 2 shows a detail of the domain class model for the quality control (test) for a production material that usually produces a QC result dataset. A material can be tested several times.

The process is modeled as a UML activity diagram (Fig. 3): The QC procedure is initiated by a control app that activates a transport robot. The material is delivered to the QC unit where the material quality check takes place. The result of the QC procedure is sent to the transport robot and the control app.

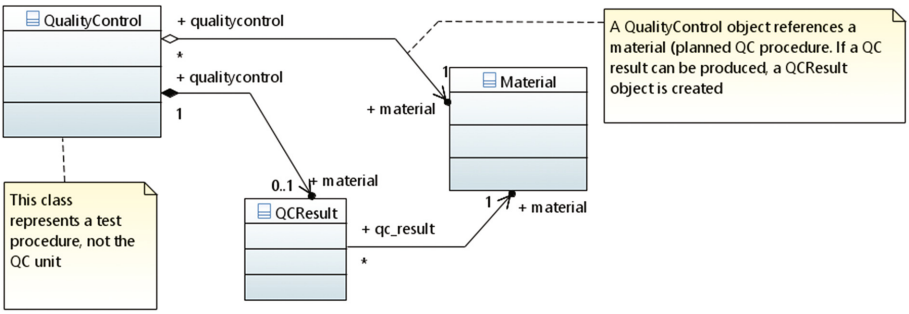


Fig. 2. Example domain class model for the production material quality control (class attributes and operations are omitted)

Data flow elements (data object, output/input PIN) in the process model can be linked with domain class elements. For example, the type specification attribute for the `qc_result` data flow is set to the class `QCResult` (Fig. 4). This model element connection is later analyzed and used for the transformation, e.g. class operations with the appropriate parameters can be generated.

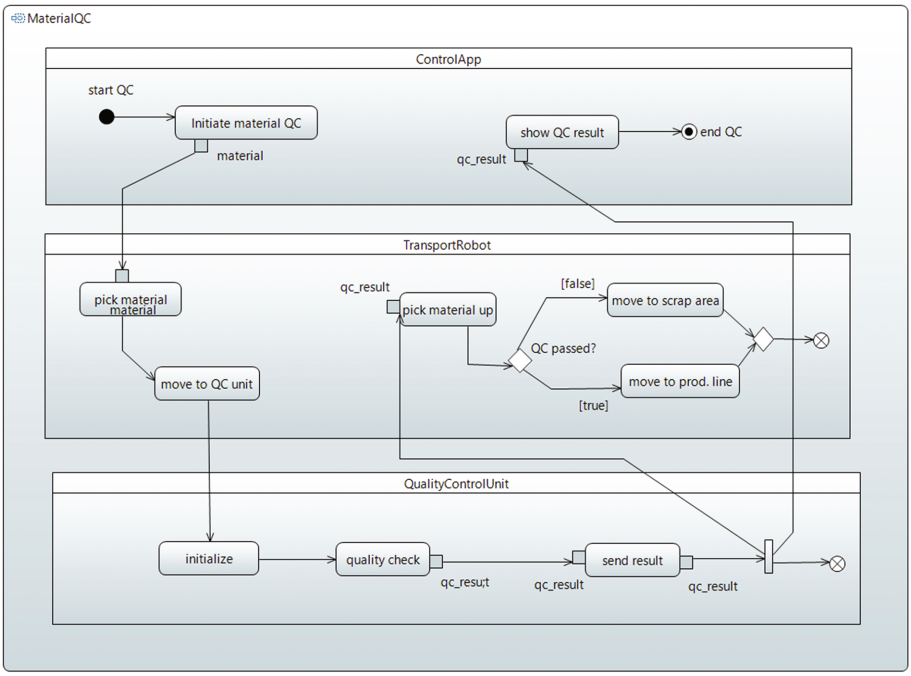


Fig. 3. Example process model for the quality control (simplified UML activity diagram)

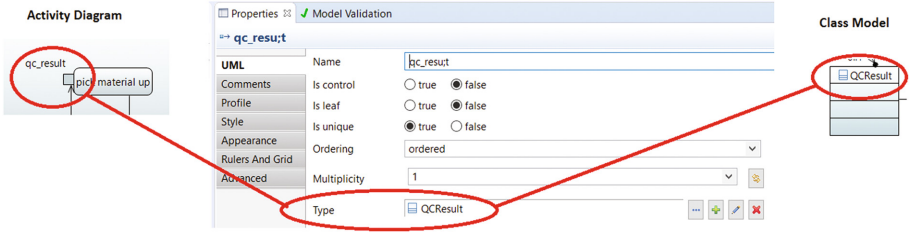


Fig. 4. Input PIN in the activity diagram with a connection to the domain class `QCResult`

This "weaving" of model elements across different model types is not only important for model transformations, but also for model validation (consistency checking). The following MOM Profile also uses this technique for the combination of behavioral and structural model elements as a preparatory step for the creation of design models.

2.3 MOM Profile Definition and Application

The MOM Profile is used for the UML activity diagrams and class models. It provides Stereotypes for mainly two different architectural components: Normal application software and MOM components. This differentiation is important for the M2M transformation. A detail of the MOM Profile is shown in Fig. 5. Process elements have a reference to the structural definition (class, package, or component).

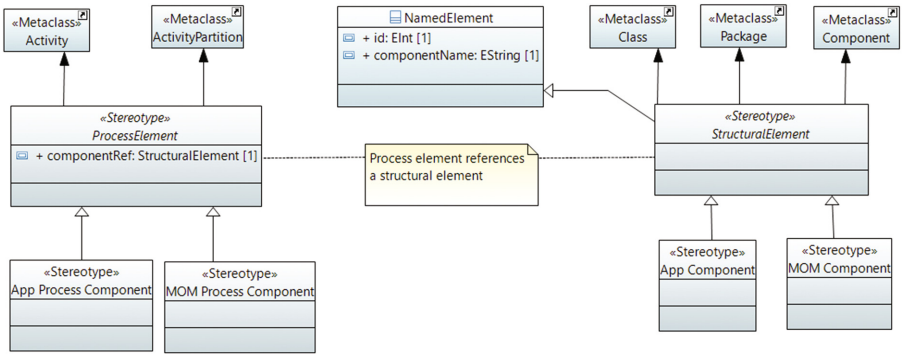


Fig. 5. Detail of the MOM Profile (simplified model)

Since partitions in an activity diagram can be interpreted as a structural element and used for M2M transformations, it seems questionable why an additional reference to an element of a structural model (class model, component diagram) is necessary. The reason is that an activity is a composite for a partition (that inherits from `ActivityGroup`). Therefore, a Partition cannot be "reused" in other activities (Fig. 6).

The application of the MOM Profile gives software architects the possibility to model architectural aspects while creating or modifying the class model: For the control app, a new package is introduced and – like the classes `RobotControl` and `QCControl` – marked with the Stereotype «App Component». The class `QCOntrol` is created and marked as a «MOM Component» (Fig. 7).

Also, the activity diagram is modified by applying the MOM Profile. Figure 8 shows the new partition for the MOM that acts as a message broker between the different system units, i.e. `ControlApp`, `TransportRobot`, and `QCControl` communicate with each other via the MOM `MaterialQCService`.

The last step before M2M transformations can take place is the creation of references between partitions in activity diagrams and classes or packages in the class model, so that different activity diagrams (partitions) can reference the same app or MOM component (structural elements). Figure 9 depicts the creation of this connection (or weaving) with the attribute (tagged value) of the Stereotype (the partition is on the left side, the package is on the right side, and the `componentRef` is in the middle).

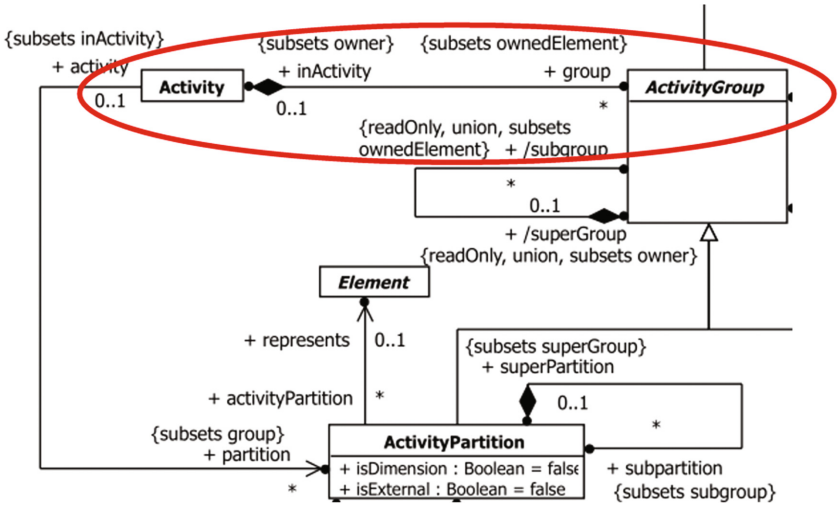


Fig. 6. Detail from the UML meta-model for activity diagrams [16, p. 404]

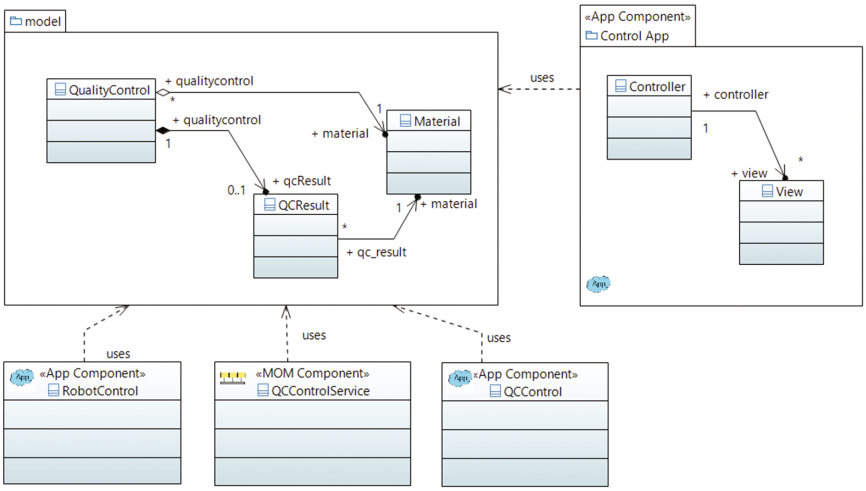


Fig. 7. Architectural class diagram with dependencies

2.4 Model-to-Model Transformation

The model-to-model transformation mapping needs to be specified before it can be implemented as an operational QVT transformation with QVTo. Rules for the mapping are formulated, e.g. a class in the source model marked with the Stereotype «App Component» will be transformed into a new controller and view class (Table 1).

Mapping functions are defined in operational QVT for classes that are marked with the Stereotype «MOM Component»: A package and a class with a consumer operation

Table 1. Detail of the M2M transformation mapping specification

Source model element	Target model element(s)	Remark
«MOM Component» class	(a) «MOM Component» classes (b) Package for MOM comp.	New package and classes for the component are created
«App Component» class	(a) Controller and view classes (b) Package for app comp.	The MVC pattern is used
Action (MOM partition)	Operations for MOM or app classes	References to the MOM classes/packages are used
Action (app partition)	Operations for controller class	References to the app classes/packages are used

are created for each MOM component (Fig. 10). The execution of M2M transformations lead to several software design models (PSM from the viewpoint of the domain model): For every component (app, MOM), a separate package is generated: App components are transformed into packages with classes for the view and controller (MVC) and MOM components are transformed to new packages with a handler for each communication channel (Fig. 8).

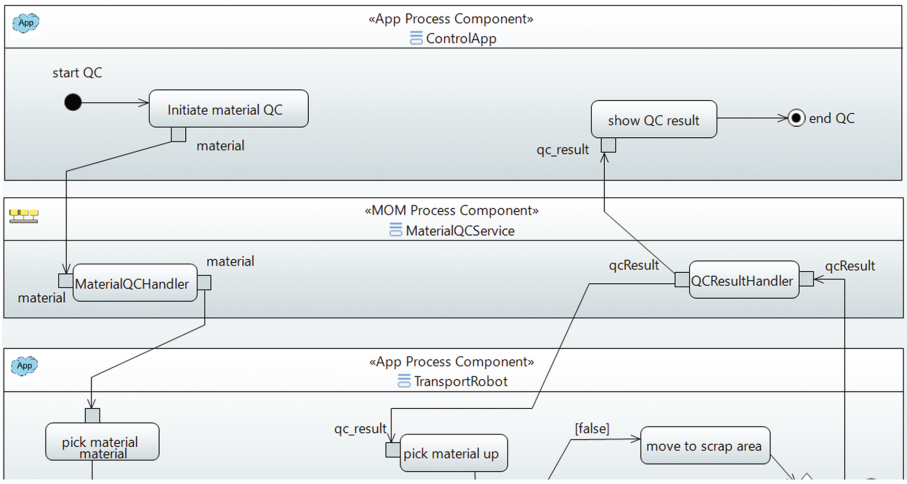


Fig. 8. Marked activity diagram with app and MOM process components

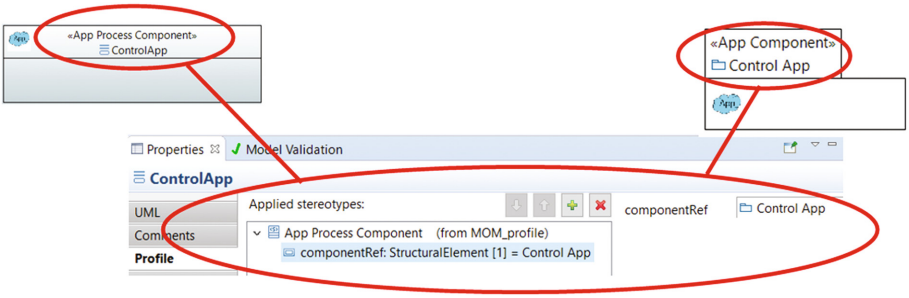


Fig. 9. Connection between marked partition and package

```
mapping Class :: transformMOMClass2ServicePackage() : Package
   when { not self.name.oclIsUndefined() and self.isStereotypeApplied(momComponentStereotype)}
{
    init { // initialization: set names for population
         var packageName := self.name + "Service";
         var qualifiedName := self.qualifiedName;
         var namesSpace := self.namespace;
   }
   population {
         object result:Package {
             name := packageName; // set the package name
             packagedElement += self -> map createMessageHandler(); // create service class
             packagedElement += self -> map createRouteBuilder(); // create route builder
             result.applyStereotype(momComponentStereotype); // apply MOM component to result package
```

Fig. 10. QVTo transformation specification for MOM classes (detail)

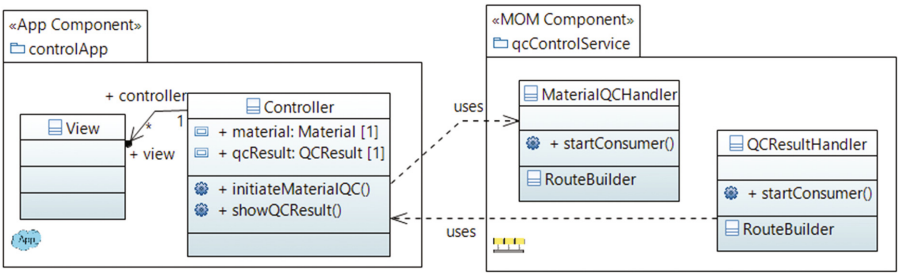


Fig. 11. Model-to-model transformation: generated class model

The following class model is a detail of the result of the M2M transformation (Fig. 11). This PSM represents a first version of a software design model.

2.5 EIP Profile Application and Code Generation

The software architect can manually modify design models concerning implementation aspects. In this case, implementation-oriented design decisions for the MOM service

Table 2. Icons and Stereotypes for the EIP Profiles

Icon [19]	Stereotype	Remark
	«Endpoint»	An endpoint is a component or an application
	«Channel»	A channel connects two or more endpoints
	«Message»	
	«Publish-Subscribe Channel»	Decouples producer and consumer. The publisher broadcasts a message to all subscribers

package are demonstrated: The cyclic dependency between the package `controlApp` and `QCControlService` will be eliminated (Fig. 11) by applying the enterprise integration pattern (EIP) Publish-Subscriber-Pattern which is part of the EIP Profile for UML class diagrams (Table 2).

The class `QCResultHandler` is marked with the Stereotype «Publish-Subscribe Channel» and the controller (package `controlApp`) becomes a subscriber. This leads to a unidirectional dependency of the package `controlApp` from the MOM service component in the package `qcControlService` (Fig. 12).

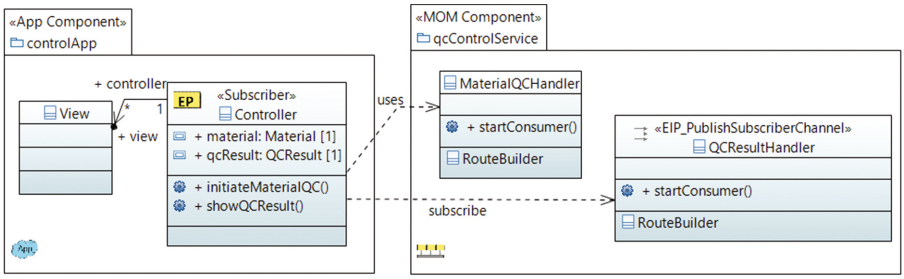


Fig. 12. Class diagram (detail) with applied "Publish-Subscriber Channel" EIP

```
[template public generateClass(pkg : Package, nspace : OrderedSet(Namespace))]
    [for (e:Element | pkg.ownedElement)]
        [if  (e.oclIsTypeOf(Class) or e.oclIsTypeOf(AssociationClass))]
           [let class : Class = e.oclAsType(Class)]
           [for (stereotype : Stereotype | class.getAppliedStereotypes())]
               [if (stereotype.name.equalsIgnoreCase('EIP_PublishSubscriberChannel'))]
                    [generateCamelPublishSubscriber(class, stereotype, nspace)/]
               [/if]
            [/for]
            [/let]
        [elseif (e.oclIsTypeOf(Enumeration))]
```

Fig. 13. Acceleo template for the code generation of Apache Camel routes (detail)

When the class model is ready for code generation (M2T transformation), a code generator for the app and MOM components is used: With the MOFM2T [20] implementation Acceleo, code generation templates for the PSI were developed. Figure 13 shows a detail of a template for the target platform (API) Apache Camel [21].

3 Conclusion

The model-driven approach for MOM development presented in this paper is a first proof-of-concept. The method consists of the following steps:

- Domain Modeling, e.g. creation of UML class model and activity diagram (PIM)
- MOM Profile application for domain model elements (PIM markup)
- Model-to-Model transformation (PIM to PSM): creation of software design models
- Design model modification and EIP Profile application for code generation
- Model-to-Text transformation (code generation: PSM to PSI)

The approach has proven useful, because most of the code for the MOM layer can be generated and it allows manual modifications of software design models. M2M reduces the complexity of the transformation specifications for the code generation: The UML (Profiles, activity diagrams, and class models) have been successfully used. In the future, more EIP patterns and model types will be included in the Profile.

References

1. Meulen, R. (Gartner): Gartner says 6.4 billion connected “things” will be in use in 2016. www.gartner.com/newsroom/id/3165317. Accessed 1 Feb 2017
2. Unger, H., Meesad, P., Boonkrong, S.: Recent advances in information and communication technology. In: Advances in Intelligent Systems and Computing, vol. 361. Springer, Heidelberg (2015)
3. Bergner, S., et al.: Networked IT-Security for Critical Infrastructures – The Research Agenda Of VeSiKi. www.itskritis.de. Accessed 14 Jan 2017
4. Slama, D., Puhlmann, F., Morrish, J., Bhatnagar, R.M.: Enterprise IoT: Strategies and Best Practices for Connected Products and Services. O’Reilly Media (2015)
5. Organization for the Advancement of Structured Information Standards (OASIS): OASIS Advanced Message Queueing Protocol (AMQP) v1.0, OASIS Standard (2012)
6. International Organization for Standardization: ISO/IEC 19464:2014 - Information technology - Advanced Message Queuing Protocol (AMQP) v1.0 specification (2014)
7. Java Community Process (JCP): Java Message Service (JMS) API. Final Release 1.1 (2002)
8. Organization for the Advancement of Structured Information Standards (OASIS): MQTT Version 3.1.1 Plus Errata 01 (2015)
9. Consel, C., Kabac, M.: Internet of Things: a challenge for software engineering. In: ERCIM - The European Research Consortium for Informatics and Mathematics, Special Theme. www.ercim-news.ercim.eu. Accessed 23 Jan 2017
10. Mukhopadhyay, S.C. (ed.): Internet of Things: Challenges and Opportunities. Springer: Smart Sensors, Measurement and Instrumentation 9 (2014)

11. Holler, J., Tsiatsis, V.: From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence, 1st edn. Academic Press, London (2014)
12. Hohpe, G., Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions, 1st edn. Addison-Wesley, London (2003)
13. Buckl, C., Sommer, S., Scholz, A., Knoll, A., Kemper, A.: Generating a tailored middleware for wireless sensor network applications. In: IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (sutc 2008)
14. Plšek, A., et al.: A component framework for java-based real-time embedded systems. In: Middleware 2008, ACM/IFIP/USENIX 9th International Middleware Conference (2008)
15. Object Management Group (OMG): Model Driven Architecture (MDA) - MDA Guide, rev. 2.0, document number ormsc/14-06-01 (2014)
16. Object Management Group (OMG): OMG Unified Modeling Language (OMG UML), version 2.5, document formal/2015-03-01 (2015)
17. Object Management Group (OMG): OMG Meta Object Facility (MOF) Core Specification. Version 2.5.1, document number formal/2016-11-01(2016)
18. The Eclipse Foundation: Eclipse Modeling project (2017). www.eclipse.org/modeling Accessed 3 Feb 2017
19. Hohpe, G.: www.enterpriseintegrationpatterns.com. Accessed 21 Jan 2017
20. OMG (Object Management Group): MOF model to text transformation language (MOFM2T). Version 1.0, document number formal/2008-01-16 (2008)
21. Apache Software Foundation: Apache Camel: Publish Subscribe Channel (2017). www.camel.apache.org/publish-subscribe-channel.html. Accessed 1 Feb 2017

Optimum Route Recommendation System to Escape Disaster Environment

Chayanon Sub-r-pa[1(✉)], Goutam Chakraborty[1], and Bhabani P. Sinha[2]

[1] Iwate Prefectural University, Sugo, Takisawa, Iwate 020-0693, Japan
chayanon.s@gmail.com, goutam@iwate-pu.ac.jp
[2] Indian Statistical Institute, Kolkata, India
bhabani@isical.ac.in

Abstract. In disaster environment such as Tsunami, people need to evacuate to safety shelter immediately. Using vehicle is the fastest way to evacuate. In case of emergency, instead of a specific destination, one needs to find route to a safety shelter, any one which could be accessed in shortest time. Existing navigation systems too can search a service instead of a specific destination. It calculates routes to nearby service points, and present a list of results to the user. The user has to take decision to select one from the list. In general people in the same area will get the same result from the system, and choose the shortest route, i.e., the nearest service point. In densely populated area, traffic congestion will appear in shortest route in a short time. Moreover destination accessible by the shortest route will quickly run out of service. It is better to choose different routes or different destinations from the beginning, by which traffic congestion could be avoided, and users will be distributed over several service points. In this paper, we proposed routing algorithm and navigation system to recommend optimum routes and destinations to users in a disaster environment. This navigation system can calculate and recommend routes considering multiple destinations and limited available resources at destinations, simultaneously.

Keywords: Intelligent transport system · Traffic control · Road network routing

1 Introduction

In an emergency situation such as Tsunami, most of the population from a wide area needs to escape to safe area. Figure 1 is the map of tsunami shelters and evacuation facilities (shown as green squares) near Sendai port, Miyagi prefecture of Northeast Japan. Orange and yellow areas were affected by Tsunami. Map includes list of evacuation and safety shelter. In general, a driver will choose the nearest safe destination via the shortest route. For densely populated area, traffic

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_25

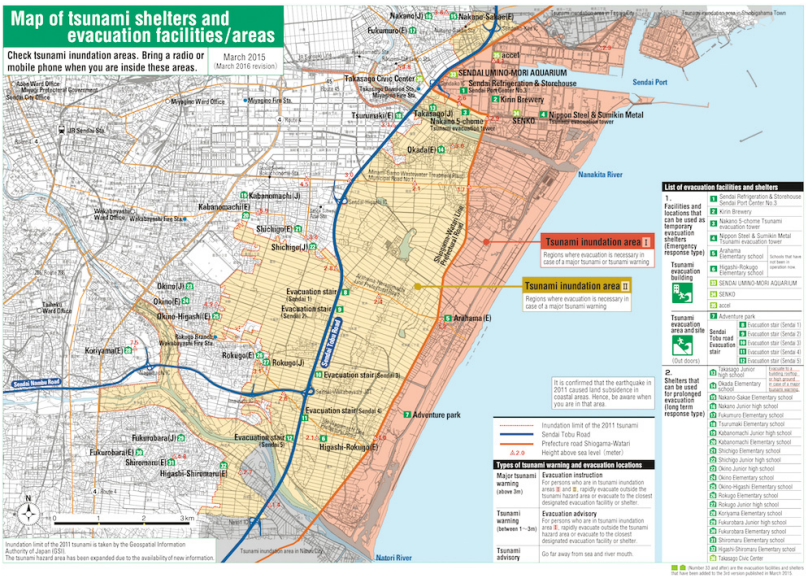


Fig. 1. Map of tsunami shelters and evacuation facilities/areas

congestion will occur on links approaching to the safe destination. Vehicles will be trapped in traffic or stop on the road itself, and eventually face disaster. If people, who attempt to evacuate to a safety shelter can get recommendation of route calculated based on road traffic and resource status at the safety shelter, they may not take the shortest route but still reach a suitable safe location in a reasonable time.

Existing navigation systems too can recommend routes to nearby service providers (like restaurants, or gas stations) instead of a specific destination. For a service inquiry, the system provides a list of results. One destination from the list, is to be selected by the user. In case of festival, for example, lot of people will select a parking lot nearest to the place of festival, creating congestion on the leading road. When they arrive the service may not be available any more. In general, the metric for optimization is shortest route, the distance calculated using map data. By psychological instinct, users choose destination by shortest distance, or shortest travel time. In case of disaster, destination and leading route that are easy to access during normal time will face heavy traffic. It will soon be congested. Immobile cars will face the disaster (like tsunami, or avalanche) on the road. It is safer to choose different routes, and/or different destinations from the beginning, even though the starting point is the same, to avoid traffic congestion and reach a service shelter where service is still available.

In this research, we propose a routing algorithm that computes multiple near-optimal routes, each to one of a few given destinations. As the routes are different, traffic is distributed to avoid congestion on the routes as well as at

the service points. Resources at different destinations are limited. The proposed route recommendation system considers the depleting resources at destinations, to avoid with higher probability a destination with scarce resources left. In our proposed algorithm, we find the near-optimal routes to multiple destinations based on a suitable extension of Yen's *k-shortest path algorithm* [15] which computes the near-optimal routes for a single destination. Traffic distribution is done for to minimize traffic congestion, and therefore travel time. The proposed algorithm has been simulated on real-map data with simulated traffic data close to real-life situation. Simulation parameters were similar to [10], and the results were encouraging.

2 Related Work

Road network is dense, and the navigation system with limited computing power and memory is incapable of using basic shortest path algorithms [2,6]. [4] proposed a system to improve shortest path algorithm to support large scale network with limited hardware resource, by using combination of heuristic methods, including hierarchical, bidirectional, and A* [5]. Hybrid algorithms are developed to reduce search space, and improve searching speed.

Dynamic Traffic Assignment (DTA) [3,12] was proposed, which led to choices between system-optimal or user-optimal route assignments. DTA was designed to help explain the basic concepts and definitions of DTA models and to address application, selection, planning, and execution of a DTA model. It also describes the general DTA modeling procedure and modeling issues that may be of concern to the model user.

During tsunami disaster on March 11th, 2011, on the pacific coast of North Japan, drowning was the main cause of death (92%), and more than fifteen thousand people lost their lives. In [10], various approaches were applied and experimentations done to understand evacuation behavior. They analyzed data of people who evacuated by different methods. The conclusion shown is that the average evacuation speeds of vehicles was 14 km/h and did not exceed 20 km/h. Many victims died on the road due to wrong decision about escape routes and means.

This paper is an extension of our previous work [13], where we considered destinations with infinite resources. Here, we consider more realistic environment with limited resources at each destination. Our routing algorithm is based on Yen's [15] k-shortest path algorithm. Yen's k-shortest path algorithm is a single source single destination shortest path algorithm with k routes to a specific destination - the first best route, the second best route, etc. We modified in several ways, Yen's algorithm to deliver routes of our routing problem. In simulation, we used parameters like vehicle speed using the work reported in [10]. The capacity of different shelters were manually collected by telephonic conversations.

3 The Proposed Scheme

In the beginning, let us define the notations. In this paper, all destinations, i.e., the set of service points are denoted as $D = |d_1, d_2, \ldots, d_n|$, where d_x is a destination, and number of destinations n is fixed. Number of destinations in a particular area is known from the map data. Recommended route includes a member of D, which is the destination node. We assume that information of the status of resource is available centrally, and is updated at periodic interval of every T minutes. Link cost metric, to calculate optimum route is travel time. Traffic density and consequently estimated time of travel on a road-segment is available, collected and made available by road-side equipments and communication infrastructure available. The underlying real-time information collection mechanism could be a part of the road network infrastructure (loop detector [1]), or VANET [14]), or crowd sourced using external service such as mobile phone [7] application.

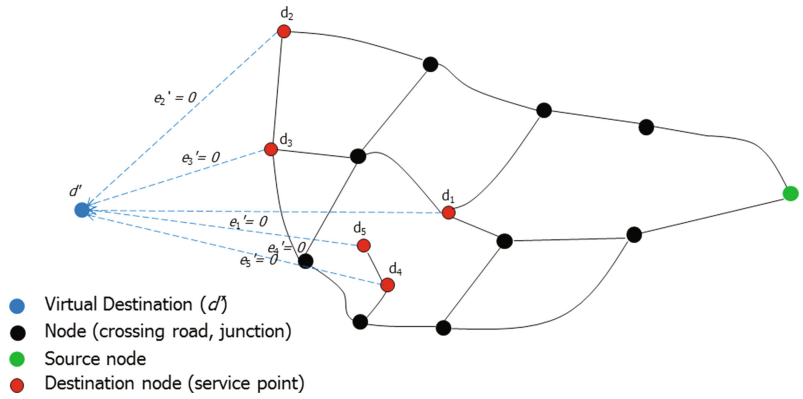


Fig. 2. An example of multiple destination network

Figure 2 is an example of multiple destinations environment, where vehicles from source node need to reach one of the five destinations. Similar situation does exist in daily life, such as looking for a gas station, a parking-lot, or searching for safety shelter in disaster environment. Existing navigation system offer such service, listing up nearby service points on the map. Then user has to take decision, selecting one from the recommended list. It is common that an user will select the closest destination without considering the cons like traffic on that route or the status of the resource availability at the target destination. This is true when one looks for a parking place near the festival ground. This is shockingly true in an evacuation environment, where many users from a locality are simultaneously trying to escape and take the same route to the nearest shelter. Sudden traffic increase, road completely blocked are very common under such circumstances. Diverting people on different routes to different service points is the only solution, a proper route recommendation system could achieve.

3.1 Multiple Destinations Routing Algorithm

To improve efficiency of shortest path algorithm with multiple destinations, we converted multiple destinations environment to a single destination problem by proposing an additional *Virtual Destination*. A new node, called as virtual destination (d') is added in the network. Virtual links are connected from all real destinations to d'. They are called as virtual links, $E' = |e'_{d_1}, e'_{d_2}, \ldots|$, where e_{d_i} is the link connecting between destination d_i to d'. At beginning we set cost of all virtual links equal to 0. In case of infinite resources available at the destinations, virtual link cost will always be zero. For limited resources at destinations, virtual link cost increases as the resource is depleted, increasing the total cost of the route via that destination, and thereby avoiding that destination for future traffic.

We explain that in Fig. 2. After applying virtual destination, we run k-shortest path algorithm based on Yen's *k-shortest path algorithm* [15], to virtual destination v'. Each source node, i.e., starting point of navigation, will get k-best routes to virtual destination. We can present route that route connecting to virtual destination, and d' predecessor node will be the real destination of that route.

However in contrast to our approach, if we would have merged all the destination points $\in D$ and replaced them by a single virtual destination d', we would have lost the identity of individual destinations. After running the Yen's algorithm on that graph, the results may contain unpractical route, touching two real destinations etc. We need to modify the algorithm result to discard some routes as discussed below.

Multiple Destinations Are in Same Road End: Normally we consider each destination as an individual destination (end point). In real road network, it is possible that the only way to reach one destination is go through some other destination. In our algorithm, we merge them and consider it as single destination with resource equal to the sum of the merged ones.

Route Contains More Than One Destination: While we find optimum routes to the virtual destination, as it is connected only via real destinations, at least one real destination will be included in the route. It possible that the i^{th}-best route ($i \neq 1$) to d' may include more than one real destinations. Vehicle following this route will stop when the first destination is arrived and need not continue to travel further. We delete such i^{th}-best route/s, that contain more than one destination.

3.2 Recommend Different Routes to Distribute Traffic

When the navigation system recommends k-shortest routes to users, in general user will choose the 1st-best from k-shortest routes. In normal situation, it is not necessary to consider 2nd-best route which would take longer time or lead

to a more distant destination. In emergency situation, it is different, because traffic increases at drastically fast. All selecting the 1st-best route will create congestion, whereby cars will be stalled on the road. Moreover, corresponding destination will be filled soon and later arrivals will find that service is no more available.

To prevent traffic congestion in 1st-best route, the traffic needs to be distributed over multiple near optimum routes, i.e., 2nd-best or 3rd-best routes. The traffic would be distributed over different destinations, and/or over different routes to the same destination, depending on the availability of the service. The goal of avoiding traffic congestion and distributing users over different service points will be achieved at the same time.

The proposed system will distribute vehicles in k-routes. The loading of different routes will depend on their respective costs. The probability that a particular route will get selected (by a user) depends on the calculated travel time for that route. The probability is inversely proportional to the travel time. The ith-best route will be selected out of k-shortest routes, with probability p_i, expressed as in Eq. (1), where c_i is cost of i^{th}-best path of all k-shortest paths.

$$p_i = \frac{(\sum_{j=1}^{k} c_j) - c_i}{(k-1)(\sum_{j=1}^{k} c_j)} \tag{1}$$

3.3 Traffic Distribution for Limited Destination Resources

With virtual destination and traffic distribution as explained in Sect. 3.2, traffic congestion in road network could be avoided. However this recommendation system do not consider that only a limited service is available at a destination. Vehicle may or may not get service when arrive at a destination. To recommend route considering depletion of resources at destinations, we proposed a new traffic distribution algorithm, called *dynamic virtual link cost*.

To explain dynamic virtual link cost, we give an example using Fig. 2 when $k = 2$, and destinations are parking lots with fixed spaces. Suppose the 1st-best route is a path to d_1 with cost = 3, and 2nd-best route is a path to d_2 with cost = 7. From Eq. (1), the system will distribute 70% of vehicles to d_1 and 30% of vehicles to d_2. Suppose, parking space at d_1 is 40, and that at d_2 is 60. There are 100 vehicles travelling from source node and follow recommended routes with traffic distribution. 70 vehicles will follow 1st-best route to destination d_1. But parking space at destination d_1 is only 40, which mean 30 vehicles will arrive at destination d_1 but will be refused. They need to re-route to another destination. If we consider only resources available at a destination and distribute vehicles considering only the space available (40 vehicles travel to d_1, and 60 vehicles travel to d_2), all vehicle will get the service when they arrive.

Figure 3 is a plot of the proportion of traffic distributed to d_1 and d_2. In this graph, traffic when distributed considering only travel time is denoted by point $\boldsymbol{A}$, and traffic when distributed considering only available resource at destination is denoted by point $\boldsymbol{B}$. The optimum point for traffic distribution, in a dynamic

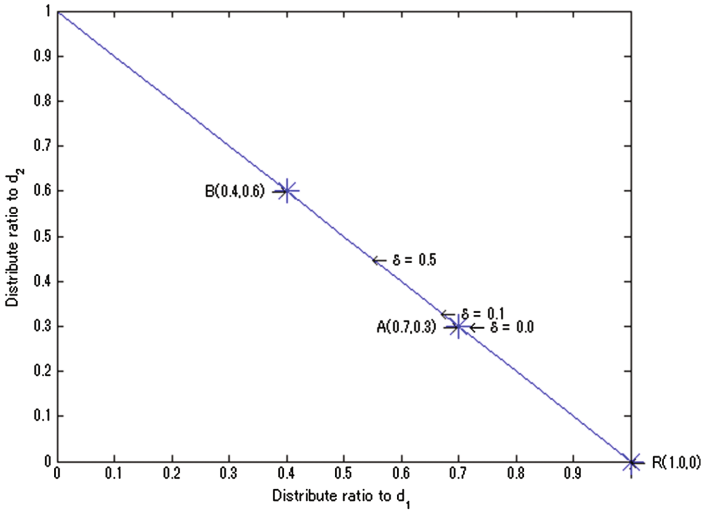


Fig. 3. Example of distribute ratio when $k = 2$, and k-closest destination is d_1, d_2

situation with limited resources, is somewhere in between, at which vehicles will smoothly take two different routes so that traffic congestion is avoided as well as the available resources are properly utilized (less U-turn at destination because of unavailability of resource). In other words, there are two optimization criterion: (1) minimum number of vehicles taking U-turn, (2) the longest time of travel for all vehicles to reach some destination is minimum.

Optimum ratio, point denoted by $\boldsymbol{X}$, is in between $\boldsymbol{A}$ and $\boldsymbol{B}$. In real situation, as the road traffic as well as the available resource at destinations vary with time, the optimum location of $\boldsymbol{X}$ need to be dynamically adjusted. As this is computationally complex, we did simulation with different fixed locations of $\boldsymbol{X}$ and analyze the results. For convenience, we define the position of $\boldsymbol{X}$ as ratio of distances from $\boldsymbol{A}$ and $\boldsymbol{B}$. Distance between $\boldsymbol{X}$ and $\boldsymbol{A}$ is defined by δ, where $\delta = 0$ means $\boldsymbol{A} = \boldsymbol{X}$, and $\delta = 0.5$ means $\boldsymbol{X}$ is at the middle between $\boldsymbol{A}$ and $\boldsymbol{B}$.

The new algorithm distribute traffic on k-shortest paths, using a ratio vector (where elements of the vector are rations for different routes) $\boldsymbol{X} = \{x'_1, x'_2, ...x'_k\}$. Virtual link cost is calculated according to Eq. (2). After calculating e'_i, the virtual link cost from destination d_i, we run k-shortest path to d' and calculate probability to select a route using Eq. (1). Here x'_i is the new distribution ratio to distribute traffic to d_i ($\boldsymbol{X} = \{x'_1, x'_2, ...x'_k\}$).

$$e'_i = -(((x'_i \times (k-1) \times \sum_{j=1}^{k} c_j) - \sum_{j=1}^{k} c_j) + c_i) \tag{2}$$

4 Experiment Setup and Results Analysis

We simulated the proposed routing algorithm in a disaster environment, where the whole population from affected area needs to be transported to safe shelters at the same time. We used the simulation package of Urban Mobility (SUMO)[9]. We downloaded road map from OpenStreetMap [8,11] and generate vehicle mobility to start movements with random starting point. Speeds of all vehicles were limited to 15–20 km/h, with uniform random distribution. This speed was adequate to evacuate from tsunami [10].

We used a part of the map affected by tsunami after the earthquake on March 11, 2011 Sendai Port, Miyagi, Japan, shown in Fig. 4. The map data used in this simulation cover about 4 km × 2 km. Safe shelters are marked in green with number. This map has 8 shelters. At each shelter service is limited by the number of parking-lot. All service space combined is equal to 2400 vehicles.

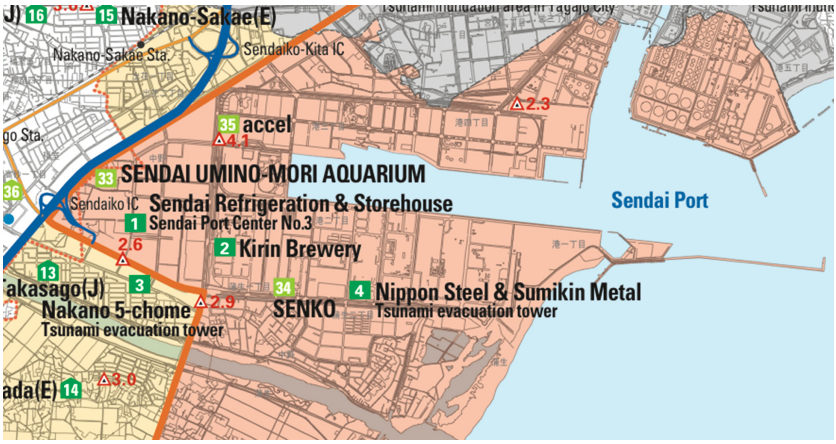


Fig. 4. Sendai port area

For each scenario, after vehicles join the road network it will travel to random directions at the beginning of simulation until first 10 min. Number of vehicles joining the network is a fixed number denoted by z. All vehicles will join the network in 30 min (1800 s). First vehicle joins the road network at $t = 0$. Next vehicle will join after $1800/z$ s of the previous vehicle. The vehicles move in haphazard directions towards diverse destinations, as it happens on an ordinary day. After 10 min from the beginning of the simulation, the tsunami alarm starts. Vehicles receive the alarm and also the route recommendation to a safe shelter by executing our proposed.

To get the average performance, this simulation is repeated 100 times with the same parameters but different start-location of vehicle and mobility. Simulations used virtual destination method to solve multiple destinations problem,

distribute traffic and dynamic virtual link cost. We compare results using different number of routes distribute (k) from $k = 1$ until $k = 5$. For dynamic virtual link cost proposed, we run simulation with different δ from $\delta = 0.0$ until $\delta = 0.5$.

4.1 Number of Vehicle Take U-Turn at Service Point

"Number of vehicle take U-turn at service point" is the number of vehicles, who follow the recommended route, but cannot get the service when they arrived at recommended destination. Those vehicles need to reroute for new recommended route to new recommended destination. If the system recommended route without considering resources available at destinations, many vehicles may need to re-route for next suitable destination. In disaster situation this will increase risk to encounter attack from disaster. Some destination may easy to access but have not enough resources. Increase virtual link cost following δ will improve this value, i.e., reduce U-turn.

Figure 5 shows the results of the number of vehicles need to reroute at destination. We group results by δ, results for each δ using different number of traffic distribution(k). From the plot it is evident that, with high δ the number of vehicles need to reroute at destination decreases.

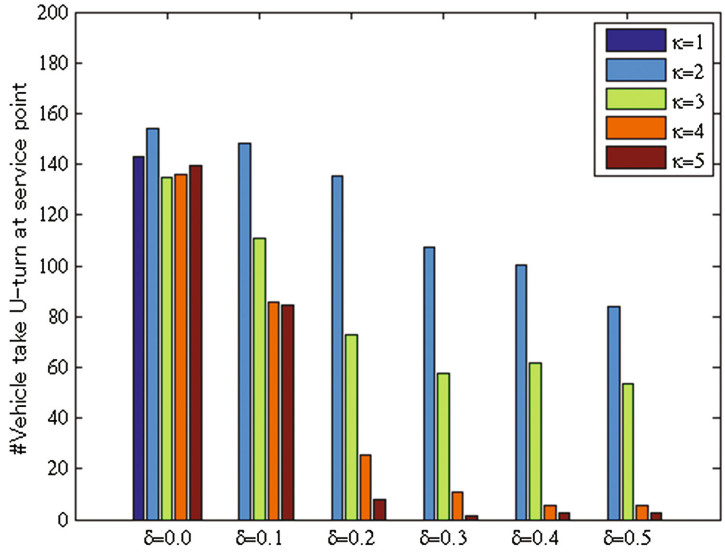


Fig. 5. Number of vehicle take U-turn at service point

We can conclude that by increasing k and δ we can improve the number of vehicle need to reroute at destination, i.e., reduce this number. This is true for all cars joining the road network at different points of time. But just increasing δ is not the best solution. We continue analyze to find optimum value of δ in terms of evacuation time.

4.2 Evacuation Time

Evacuation time is the time when last vehicle arrive at service point. This is the time required from starting of evacuate until all vehicles are in safe shelters. As common requirement, we need all vehicle evacuate to safety shelter as fast as possible. Figure 6 shows the results of evacuation time. High δ does not improve evacuation time. And most effective k value is $k = 2$, and $k = 3$.

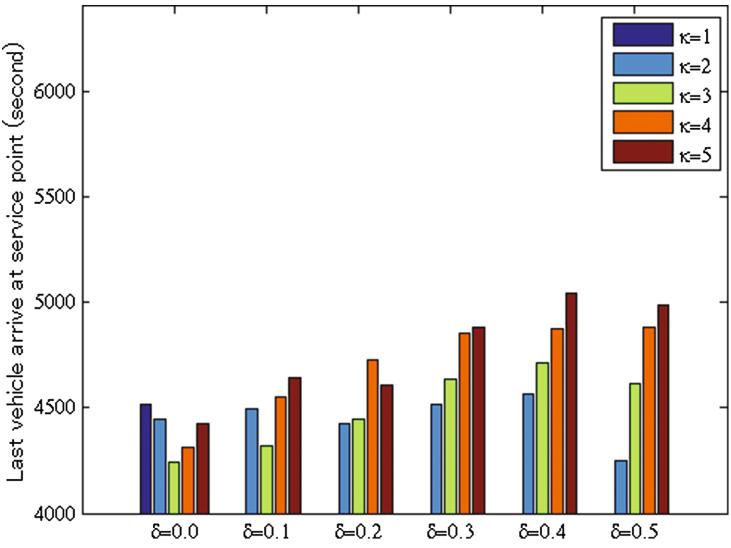


Fig. 6. Evacuation time (s)

We can conclude that increase δ will not improve evacuation time. And increase k more than 3 is also not improve evacuation time. However the best result, which is lower number of vehicle take U-turn at service point and evacuation time, is when using this proposed with k is 2 or 3, and δ is 0.1, or 0.2.

5 Conclusion and Future Work

In this paper, we have proposed an optimum route recommendation system, to recommend route to vehicle consider multiple destinations, traffic distribution, and destination resource limited simultaneously. The proposed algorithm can find multiple near-optimal routes to a number of target destinations. We proposed virtual destination for computing these multiple routes and eliminating those involving more than one destination points. After computing these multiple routes, the vehicles are distributed along different routes based on the travel time, and destination resource limited. The proposed algorithm has been simulated using real map data. The results confirm that distributing traffic along

multiple routes following our proposed algorithm improves parameters like evacuation time of the last vehicle, and number of cars to take U-turn when they reach a destination, due to unavailability of resource. The balance of setting the virtual link cost as the resource is depleted is important. Through simulation we have shown desirable values for δ, which depends on the road map, vehicles to accommodate and resources at different destinations.

References

1. Anderson, R.: Electromagnetic loop vehicle detectors. IEEE Trans. Veh. Technol. **19**(1), 23–30 (1970)
2. Bellman, R.: On a routing problem. Q. Appl. Math. **16**, 87–90 (1958)
3. Chiu, Y.C., Transportation Research Board, Transportation Network Modeling Committee, National Research Council (U.S.): Dynamic Traffic Assignment: A Primer. Transportation Research Circular. Transportation Research Board (2011). https://books.google.co.th/books?id=J7aEnQAACAAJ
4. Cho, H.J., Lan, C.L.: Hybrid shortest path algorithm for vehicle navigation. J. Supercomput. **49**(2), 234–247 (2009). http://dx.doi.org/10.1007/s11227-008-0236-7
5. Delling, D., Sanders, P., Schultes, D., Wagner, D.: Engineering route planning algorithms. In: Lerner, J., Wagner, D., Zweig, K.A. (eds.) Algorithmics of Large and Complex Networks: Design, Analysis, and Simulation, pp. 117–139. Springer, Heidelberg (2009). http://dx.doi.org/10.1007/978-3-642-02094-0_7
6. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**(1), 269–271 (1959)
7. Google: Google maps. http://www.google.com/maps/
8. Haklay, M., Weber, P.: Openstreetmap: user-generated street maps. IEEE Pervasive Comput. **7**(4), 12–18 (2008)
9. Krajzewicz, D., Erdmann, J., Behrisch, M., Bieker, L.: Recent development and applications of SUMO - Simulation of Urban MObility. Int. J. Adv. Syst. Meas. **5**(3–4), 128–138 (2012). http://elib.dlr.de/80483/
10. Lee, J., Hatoyama, K., Ieda, H.: Formulation of tsunami evacuation strategy to designate routes for the car mode - lessons from the three cities in Tohoku area, Japan. Proc. East. Asia Soc. Transp. Stud. **9**, 12 (2013). http://www.easts.info/on-line/proceedings/vol9/PDF/P41.pdf
11. OpenStreetMap: Openstreetmap. http://www.openstreetmap.org/
12. Peeta, S., Ziliaskopoulos, A.K.: Foundations of dynamic traffic assignment: the past, the present and the future. Netw. Spat. Econ. **1**, 233–265 (2001)
13. Sub-r-pa, C., Chakraborty, G., Sinha, B.P.: Route recommendation system to support multiple destinations and multiple routes to minimize road congestion. Int. J. Commun. Netw. Distrib. Syst. 17 (in press)
14. Uzcategui, R., Acosta-Marum, G.: Wave: a tutorial. IEEE Commun. Mag. **47**(5), 126–133 (2009)
15. Yen, J.Y.: Finding the k shortest loopless paths in a network. Manage. Sci. **17**(11), 712–716 (1971)

Comparative Study of Computational Time that HOG-Based Features Used for Vehicle Detection

Natthariya Laopracha(✉) and Khamron Sunat

Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen, Thailand
natthariya@gmail.com, Khamron_sunat@yahoo.com

Abstract. HOG produces a number of redundant and long features so that they affect to the detection rate and computational time. This paper studied the processes that HOG-based features were generated, selected, and used in vehicle detection and find one that takes the shortest time. There were five combinations of feature extractors and classifiers. Time spent in HV step, accuracy of detection and the false positive rate are considered together for making decision of which combination is the best. The experiments were conducted on GIT dataset. The experimental results showed that process which VHOG preceded ELM provided a little less accurate than HOG preceded SVM did. However, it did not only take shortest time in HV step but also provided the lowest false positive rate. Therefore, VHOG preceded ELM should be selected as a method for vehicle detection.

Keywords: Vehicle detection · Feature selection method · Histograms of Oriented Gradients (HOG) · Support Vector Machine (SVM) · Extreme Learning Machine (ELM)

1 Introduction

Intelligent vehicle detection systems have been developed because they are useful and are beneficial to both drivers and passengers. The developed systems have been applied to a number of applications, for instance, automatic-car systems, information traffic applications, safety and security systems, and the application for automatically finding car parks. Most systems were developed by using vision-based methods or sensor systems. Computer vision, compared to the traditional sensor systems, can ease the system build because the system can perceive information. In addition, information (the images) can be collected for further use, for example, as the evidence in intelligent security systems. The Scale Invariant Feature Transform (SIFT), which determines some significant points based-on their physical properties, and defines them as a set of key points [1]. Descriptive features can be, then, generated from the key points. SIFT is considered as rotational and scale invariant feature extraction technique. Therefore, SIFT can be applied and implemented for vehicle detection applications, such as, automatic car counting [2], and vehicle tracking by using key points [3]. The SIFT is

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_26

resistant to scale and rotation change but is not resistant to blurring and affine image. Thus, adding the color invariance to SIFT, a.k.a. CSIFT, can improve the performance [4]. Haar-like technique was also reported in literature that the method has been applied in a vast variety of computer vision applications. It extracts features by computing weights in white and black rectangles [5]. Haar-like feature has been also used in vehicle detection application by estimating a distance under challenging lighting conditions [6]. In addition, an adapted Haar-like feature has been used to improve efficiency of object detection [7] by dividing parts of vehicle images and then applying the technique to extract features. However, Haar-like not only is sensitive to illumination changes but also produces redundant features. Gabor filter [8] is one of the techniques that has been applied in vehicle detection systems. However, limited bandwidth and the massive size of features are its major drawbacks. Accordingly, Log-Gabor filter has been proposed later to alleviate the problems.

Recently, a common method used for extracting and representing object in computer vision applications is the Histograms of Oriented Gradients (HOG), which was proposed by Dalal and Triggs [9]. HOG is able to extract features in low quality images. HOGs have been widely used in the many applications such as face detection [10], moving text detection [11], and detection of copy-move image forgery [12]. In particular, HOG has been adopting and using in traffic flow monitoring systems [13], detection of engineering vehicles [14] and detection of cars in aerial image [15]. Vertical Histograms of Oriented Gradients (VHOG) describes images by determining shape or edge direction [16], which is computed in local regions. Then, the gradient orientations are grouped to be vector features in the same manner of HOG does. The difference between the two algorithms is that VHOG divides images vertically, whereas HOG divides images both horizontally and vertically. Therefore, HOG produces a larger number of features than VHOG does. Accordingly, VHOG consumed shorter time than HOG did. However, VHOG yielded a lower detection rate than HOG when the support vector machine (SVM) follows them [16]. The piHOG [17] is another method that can improve the efficiency of HOG-based features. However, it is prone to suffer from the computation burden.

Although, HOG provided a promising efficiency for vehicle detection, however, it usually produced a number of redundant and long features. Producing that kind of features does directly affect the application running in real time environment [17]. In vehicle detection systems, promptly response from the system is required to avoid the chance of accidents that can occur anytime. Therefore, time used to complete the task is very important in real time environment. To shorten HOG computation time, there are number of methods that have been proposed for reducing the redundancy of HOG features, VHOG is an example.

The objective of this paper is to study the processes that HOG-based features were generated, selected, and used in vehicle detection and to find one that takes the shortest time. There are several entities in our study; HOG, VHOG, feature selection methods, extreme learning machine (ELM), and support vector machine (SVM).

The rest of this paper is organized as follows. Section 2 introduces the flow of our study including the HOG computation. Section 3 represents the Experiment and analysis. Finally, Sect. 4 is the conclusion.

2 Methodology

Vehicle detection base on computer vision includes two steps (1) hypothesis generation (HG) and (2) hypothesis verification (HV) steps. The HG generates the locations of the vehicle. The HV removes the false detection in the classification stage. This paper focuses the HV because it affects to the time used for detecting vehicles in an image. Traditionally, the HV consists of two processes, i.e., (1) feature extraction and (2) classification methods. Most of vehicle detection systems use HOG for generating features and use SVM to classify. In general, HOG can extract image in several conditions, for example, various light conditions, complexity background environment, and low quality image. However, the features produced from HOG are long vector that affects to time computation of vehicle detection. This paper proposes a technique to reduce the computational time used in generating HOG. It is divided into three tasks; (1) VHOG that is used to reduce the number of HOG blocks by extracting feature in the vertical sections of an image [16] – i.e., vertical histograms of oriented gradients (VHOG), (2) feature selection methods and (3) classification by ELM. The VHOG reduces the HOG time used in the feature extraction step and can produce less number of features. In feature selection, some features are discarded, which contributes a smaller set of vector features. Feature selection method for feature reduction of HOG includes Correlation-based Feature Selection (CFS) [18], Sparse Multinomial Logistic Regression Method (SBMLR) [19]. According to the literature, SVM was popular but ELM used very little computation time, compared to SVM used. Therefore, we selected SVM and ELM to be the classifiers. The methodology of the proposed technique shows in Fig. 1, which comprise of two stages for vehicle detection; offline and real-time stages. Firstly, the feature selection method performs the optimal features selection in offline stage. Secondly, the resulting indices are used in the real-time stage. Features are learned by SVM and ELM. Then the results of learning are collected and used in the real-time stage. The real-time stage receives a sequence of images and clips the objects off. Each clipped object may be a vehicle image or a non-vehicle image. The features of images are then extracted by HOG and VHOG. The features produced by HOG are reduced. After that, the features are classified by ELM and SVM. The outcome is either vehicle or non-vehicle object.

2.1 Histograms of Oriented Gradients (HOG)

Dalal et al. [9] developed HOG descriptor to represent human shape in an image. The HOG describes image by determining shape or edge direction. It computes in local regions and groups the gradient orientations as vector features. HOG-based feature is one of the most popular feature representations for object detection because it is robust to environmental conditions; for example, light, noise, and objects color differences [9]. The HOG method is calculated as follows.

1. Compute the gradient of the given grayscale image (I). The method is applied by the one dimension mask in the horizontal and vertical directions with the following filter

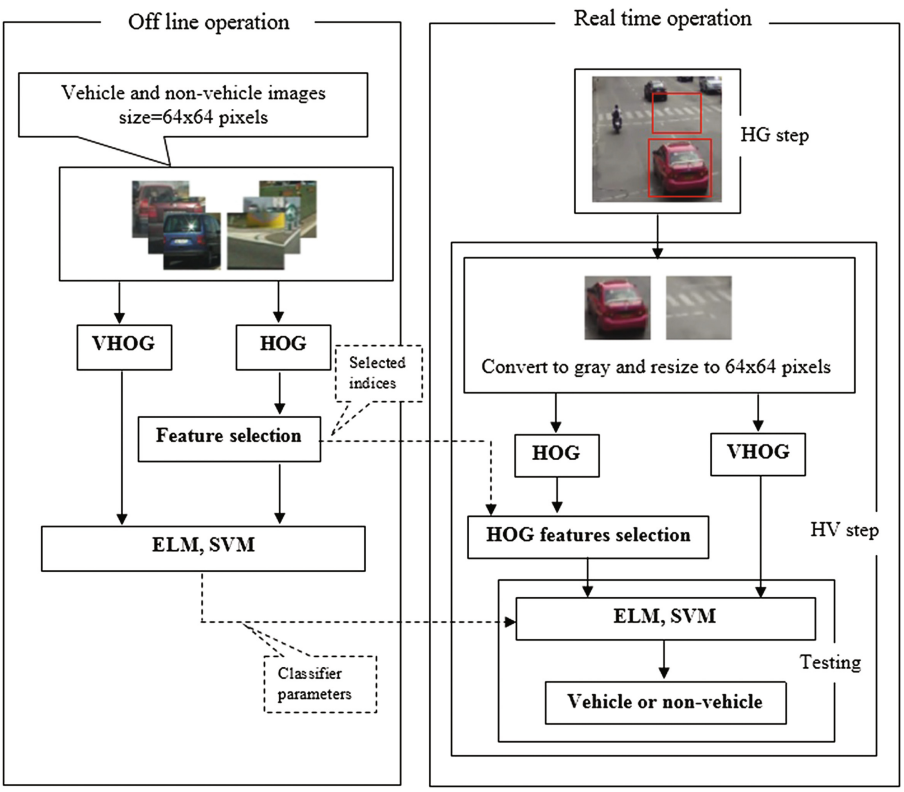


Fig. 1. Flow of our study

kernels in Eqs. (1) and (2). Then convolution operation with gray image (I) which shows in Eqs. (3) and (4).

$$M_x = [-1 \;\; 0 \;\; 1] \tag{1}$$

$$M_y = [1 \;\; 0 \;\; -1]^T \tag{2}$$

$$G_x = M_x * I \tag{3}$$

$$G_y = M_y * I \tag{4}$$

2. Calculate the magnitude of the gradient ($G(x, y)$) and the orientation of the gradients by Eqs. (5) and (6), respectively.

$$|G| = \sqrt{G_x^2 + G_y^2} \tag{5}$$

$$\theta = arctan \frac{G_y}{G_x} \tag{6}$$

3. Decompose the image into blocks; each block consists of cells as shown in Fig. 2.

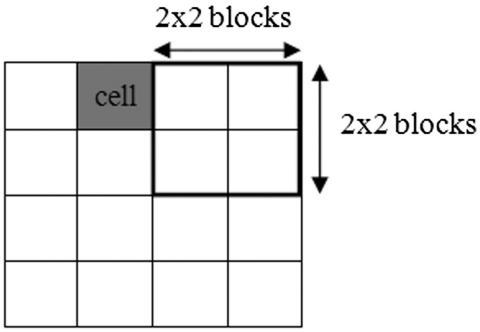


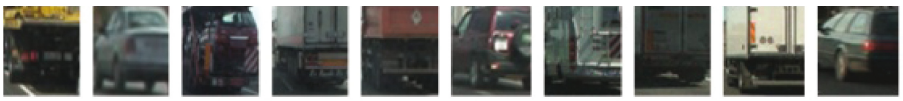
Fig. 2. Structure of HOG

4. Accumulate the histogram in each block to represent local histogram of each bin and add the gradient to that bin.
5. Collect the normalized histogram of each block. Then collect that histogram from every block to form the descriptors for object classification or detection.

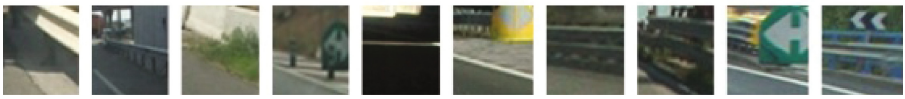
3 Experiment and Analysis

3.1 Database Preparation

The proposed method applied to GTI public data set [20]. The GTI is captured from videos, which comprises 4000 vehicle and 4000 non-vehicle images. The size of images is 64 × 64 pixels and the images are divided into four regions, i.e., far, front, left and right. Non-vehicle of GTI consists of roads, street lines, traffic signs, and road barriers. The samples of GTI dataset are shown in Fig. 3.



(a) vehicle images



(b) non-vehicle images

Fig. 3. GTI dataset

3.2 Experiment Setup

This paper aims to study the time reduction of HOG-based features used in vehicle detection. The parameters of the HOG affect time computation, which consist of the size of the block (b) and the number of cell and orientation bins (β). We use two different sizes of the block; $b = 4$ and $b = 8$. For each block size, the number of cells is 2×2 and β is 9.

In feature selection, Correlation-based Feature Selection (CFS) [18] and sparse multinomial logistic regression incorporating Bayesian regularization using a Laplace prior (SBMLR) [19] are used as the feature selection methods in the study. The CFS provided good performance, compared with the other feature selection methods [21]. SBMLR is very fast technique for feature selection. Both of CFS and SBMLR run on Matlab environment connecting to Weka library [22]. There are 30 distinct sets of train-test data for model evaluation. Each set of the train-test data is constructed from the GTI dataset by using the 50% 'holdout' cross validation. Features are classified by SVM and ELM. ELM takes shorter running time than SVM.

3.3 Feature Representation Experiments and Analysis

The HOG can produce high accuracy but may not applicable for implementing in real time environment because the HOG composes of high dimension that affects to classification time. The high number of dimensions of HOG presents redundant features. Some redundant features affect to discriminative ability because those features of vehicle and non-vehicle are similar values. Figure 4 shows how do the features from the different extractors distribute. As can be seen from Figs. 4(b) and (c), VHOG and FsCFS can discriminate vehicle from non-vehicle clearer than HOG; Fig. 4(d), SBMLR produce more overlapping of vehicle and non-vehicle features than VHOG and FsCFS do.

3.4 Performances Experiments and Analysis

This paper studies time reduction of HOG-based features used in real time application because time is very important in vehicle detection system. VHOG and feature selection methods are compared with the original HOG. Then, SVM and ELM are used in the classification process for time and accuracy comparison in the HV step. Figure 5 shows time comparison of HOG, VHOG, SBMLR and FsCFS, in HV step. The process that HOG preceded SVM consumes the longest time. The process that VHOG preceded ELM is the fastest. FsCFS and SBMLR are very effective feature selection methods. The dimension of HOG features is reduced so that the HOG-(FsCFS or SBMLR)-SVM take shorter time than HOG preceded SVM takes. Using $b = 4$ consumes longer time than $b = 8$ because the number features of $b = 4$ is higher than that of $b = 8$. ELM alone is faster than SVM. However, both SBMRL and FsCFS classified by ELM are not the fastest process, either. Because these feature selection methods need additional time to perform the feature selection before classified by ELM.

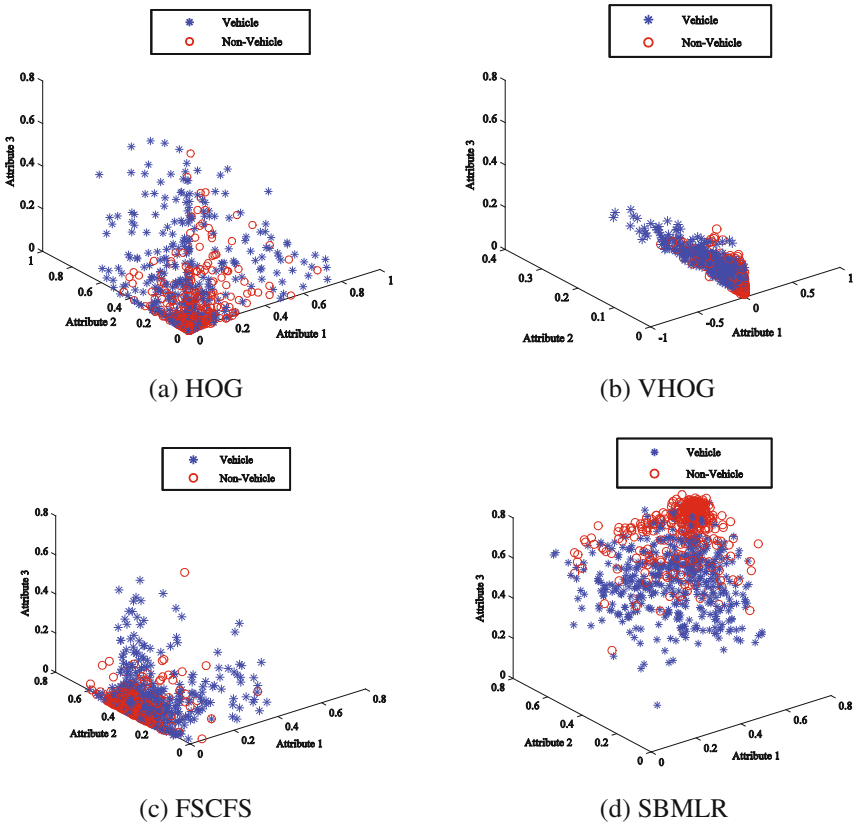


Fig. 4. Scatter plots of vehicle and non-vehicle features produced form different feature extractors and feature selectors: (a) HOG, (b) VHOG, (c) FSCFS, and (d) SBMLR

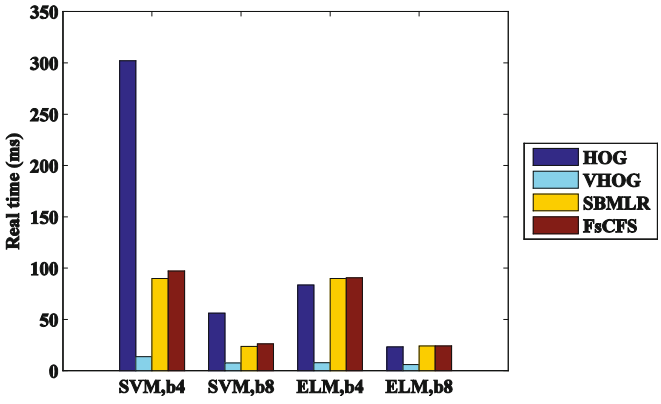


Fig. 5. Time comparison of HOG, VHOG, SBMLR and FsCFS

Figure 6 shows the comparison of accuracies produced from difference b and classifiers. The process that HOG preceded SVM provides the highest accuracy, whereas the processes that VHOG, SBMLR, or FsCFS preceded SVM and the processes that HOG, SBMLR, or FsCFS preceded ELM produced less accuracy. The process that VHOG preceded ELM also provides a high accuracy but it is not as high as that provided by HOG preceded SVM. The effective b for SVM is 4 and the effective b for ELM is 8. Thus, the two classifiers require different block size.

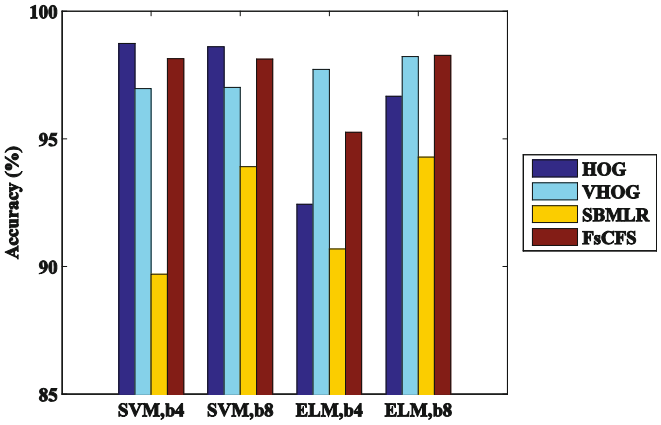


Fig. 6. Accuracy comparison of HOG, VHOG, SBMLR and FsCFS

Table 1 shows the accuracies of the classification, HOG produces the best accuracy (Acc) when it is followed by SVM. However, HOG preceded SVM used the longest time in training (TT), testing (TS), and HV (THV) steps. For safety, the fault positive rate (FPR) must be as low as possible. The process that VHOG preceded ELM provides a little less accurate than HOG preceded SVM. However, VHOG preceded ELM does not only provide the lowest FPR but also take shortest time in HV step. Therefore, VHOG preceded ELM is a good methods for vehicle detection.

Table 1. Comparison best accuracy

No.	Method	Acc (%)	FPR	TPR	TT (ms)	TS (ms)	THV (ms)	b
1	HOG-SVM	98.74	0.0135	0.9884	612.88	221.76	301.96	4
2	HOG-SVM	98.61	0.0149	0.9867	227.04	35.98	56.08	8
3	HOG-FsCFs-ELM	98.27	0.0118	0.9774	75.14	0.0020	24.05	8
4	VHOG-ELM	98.22	0.0112	0.9759	82.92	0.0018	5.90	8
5	HOG-FsCFs-SVM	98.14	0.0167	0.9792	62.35	8.98	97.09	4

4 Conclusion

This paper focuses on time reduction of using HOG-based features in vehicle detection. There were five combinations of feature extractors and classifiers. To decide that which combination is the best, four measures must be considered together; time spent in HV step, accuracy of detection, and the false positive rate. The experiments were conducted on the GIT dataset. The experimental results showed that the process with VHOG preceded ELM provided a little less accurate than HOG preceded SVM. However, it did not only take shortest time in HV step but also provide the lowest FPR. Therefore, VHOG preceded ELM should be selected as a method for vehicle detection.

References

1. Mo, G., et al.: A method of vehicle detection based on SIFT features and boosting classifier. J. Converg. Inf. Technol. **7**(2), 328–334 (2012)
2. Moranduzzo, T., Melgani, F.: Automatic car counting method for unmanned aerial vehicle images. IEEE Trans. Geosci. Remote Sens. **52**(3), 1635–1647 (2014)
3. Yang, S., et al.: On-road vehicle tracking using keypoint-based representation and online co-training. Multimed. Tools Appl. **72**(2), 1561–1583 (2013)
4. Wu, J., et al.: A comparative study of SIFT and its variants. Meas. Sci. Rev. **13**(3), 122–131 (2013)
5. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. Comput. Vis. Pattern Recogn. **1**, 511–518 (2001)
6. Rezaei, M., Terauchi, M., Klette, R.: Robust vehicle detection and distance estimation under challenging lighting conditions. IEEE Trans. Intell. Transp. Syst. **16**(5), 2723–2743 (2015)
7. Park, K.Y., Hwang, S.Y.: An improved Haar-like feature for efficient object detection. Pattern Recogn. Lett. **42**, 148–153 (2014)
8. Sun, Z., Bebis, G., Miller, R.: ON-road vehicle detection using gabor filters and support vector machines. In: Proceedings of 14th International Conference on Digital Signal Processing, vol. 2, pp. 1019–1022 (2002)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 886–893 (2005)
10. Tan, H.L., Yang, B., Ma, Z.M.: Face recognition based on the fusion of global and local HOG features of face images. IET Comput. Vis. **8**(3), 224–234 (2014)
11. Khare, V., Shivakumara, P., Raveendran, P.: A new histograms oriented moments descriptor for multi-oriented moving text detection in video. Expert Syst. Appl. **42**(21), 7627–7640 (2015)
12. Lee, J.C., Chang, C.P., Chen, W.K.: Detection of copy—move image forgery using histograms of orientated gradients. Inf. Sci. **321**, 250–262 (2015)
13. Pham, H.V., Lee, B.R.: Front-view car detection and counting with occlusion in dense traffic flow. Int. J. Control Autom. Syst. **13**(5), 1150–1160 (2015)
14. Liu, X., et al.: Detection of engineering vehicles in high-resolution monitoring images. Front. Inf. Technol. Electron. Eng. **16**(5), 346–357 (2015)
15. Moranduzzo, T., Melgani, F.: Automatic car counting method for unmanned aerial vehicle images. IEEE Trans. Geosci. Remote Sens. **52**(3), 1635–1647 (2014)

16. Arrospide, J., Salgado, L., CamPlani, M.: Image-based on-road vehicle detection using cost-effective histograms of oriented gradients. J. Vis. Commun. Image Represent. **24**, 1182–1190 (2013)
17. Kim, J., Baek, J., Kim, E.: A novel on-road vehicle detection method using πHOG. IEEE Trans. Intell. Transp. Syst. **16**(6), 3414–3429 (2015)
18. Hall, M.A.: Correlation-based Feature Selection for Machine Learning. Ph.D. Thesis, University of Waikato, Hamilton (1999)
19. Cawley, G.C., Talbot, N.L.,Girolami, M.: Sparse multinomial logistic regression via Bayesian l1 regularisation (2007)
20. GTI Vehicle Image Database. Image Processing Group at UPM (2011). Accessed 18 Nov 2013
21. Chuang, L.-Y., et al.: A hybrid feature selection method for DNA microarray data. Comput. Biol. Med. **41**(4), 228–237 (2011)
22. Frank, E., et al.: Data mining in bioinformatics using weka. Bioinformatics **20**, 2479–2481 (2004)

Natural Language Processing

Android IR - Full-Text Search for Android

Mario Kubek(✉), Robert Schweda, and Herwig Unger

Chair of Communication Networks, FernUniversität in Hagen, Hagen, Germany
{mario.kubek, herwig.unger}@fernuni-hagen.de,
robert.schweda@googlemail.com

Abstract. Modern mobile devices such as smartphones concentrate information from various sources that provide textual contents, mainly in the form of e-mails, short and instant messages, web documents and social network posts. While the respective apps make it especially easy to intuitively consume and create such contents, the analysis of large amounts of natural language text on mobile devices is still uncommon, although their hardware is mostly powerful enough to carry out this task. This paper presents with *Android IR* a first solution for effective and power-saving full-text search on Android devices. Its features and working principles are described in detail. Furthermore, the app's performance is evaluated using real-world text documents.

Keywords: Mobile search · Full-text search · Natural language processing · Android · SQLite · Mobile secretary

1 Introduction

Mobile technologies play an increasingly important role in everyday life. As an example, smartphones, tablets and even smartwatches are naturally used in both private and business sectors and can be regarded as centres of personal information processing as these portable devices concentrate diverse data and information streams and make them instantly and intuitively accessible by means of mobile applications (apps) running on them. While being able to also handle multimedia data with ease, these apps are mostly designed for the consumption and (to a lesser extent) the creation of textual contents. Thus, natural language text, be it in the form of e-mails, short and instant messages, web documents or social network posts, is the most important means to convey information. And its importance is constantly rising: for instance, the number of active users of the instant messaging service WhatsApp [1] amounted to about 1 billion in February 2016. On 2016's New Year's Eve, 63 billion messages [2] have been sent using this service. Also, the data traffic volume for private web usage and sending e-mails in 2016 reached 9170 PetaByte per month in 2016 [3]. These high numbers suggest that the automatic separation of important from unimportant or even unsolicited information (one could simply speak of data in this case, too) is not only a central problem today, but will significantly grow in the near future. As this classification heavily depends on the particular recipient's–it could be a person or institution alike– interests and information needs, these parameters must be taken into account when designing and training classifiers for this purpose. To the same extent, it must be

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_27

ensured that (especially valuable) information and knowledge (already transformed information) is not lost and can be retrieved with high reliability, a task that usually involves (at least once) the extraction, analysis and indexing of text from documents in a variety of formats such as HTML, PDF, DOC(X) and plain text. These four processes can be effectively and efficiently carried out by standard open-source tools and libraries available, the most prominent ones being written in Java and provided by the Apache Software Foundation (https://www.apache.org/), e.g. *Apache Tika*, *Apache OpenNLP* and *Apache Lucene*. They are traditionally applied in text analysis pipelines [4] andoptimised to run on servers, desktop computers and laptops.

By this time, however, the hardware of modern smartphones is powerful enough to handle these tasks as well. Even today's mid-range smartphones are often equipped with more than 4 GB of RAM, fast multi-core processors and internal memory of 32 to 64 GB. Moreover, they are usually powered by a strong, rechargeable battery with 4000 mAh or more while at the same time their power-consumption is decreased by power-saving features of the operating system (OS) they run on.

Despite these facts, current mobile devices along with their installed apps mostly act as an intuitive terminal to request from or send data to remote services or servers, which usually carry out the (supposed to be) computationally expensive tasks. The analysis of large amounts of unstructured natural language text or other kinds of data on such devices is still uncommon. Economic interests play a significant role at this. For instance, companies can only gather insight from customer-related information when it is stored in their warehouses.

However, it is sensible to offer text processing solutions such as an integrated full-text search component that can run directly on mobile devices, especially smartphones. Several reasons for this assertion can be given:

- As stated above, these devices concentrate information from different sources.
- It can be enriched with context-related information provided by analysing data from the devices' sensors or adding relevant metadata from images on them to it.
- This unique composition of information as well as an in-depth analysis of it is likely of value to the user as she or he put it on or downloaded it to the mobile device in the first place.
- When the analysis can take place directly on the mobile device, the user's privacy is maintained as the information on her/his device does not have to be propagated to a possibly untrusted third party using an unsecured connection in order to use an analysis service.

Therefore, this paper introduces the Android app*Android IR* which represents a first solution for effective, power-saving and completely integrated full-text search for Android devices. The next section presents a short overview of existing approaches and solutions in the field of mobile information retrieval which also explains why Android is the currently best suited OS to develop mobile IR apps for. In Sect. 3, the components, features and working principles of *Android IR* will be described in detail. Afterwards, its performance is evaluated in a number of experiments. Section 5 outlines planned extensions to *Android IR* which will turn it into a powerful "mobile secretary" with the goal to improve the user's personal information management (PIM)

by integrating, analysing and semantically connecting information from the mentioned sources. Section 6 concludes the paper and summarises the presented developments.

2 Mobile Information Retrieval

When it comes to mobile information retrieval (IR), in literature, a common point of view on this topic seems be that mobile devices only act as "intelligent" search masks. As an example, Flora et al. [5] use the term "context awareness" to explain this view: the respective search context is augmented and enriched by the many features of the devices such as the integrated location functions. Practical examples for this approach are Apple's question answering system *Siri*, Microsoft's analogon *Cortana*, the music recognition app *Shazam* or one of the many location-based web services like *Foursquare*. A direct connection between the Android OS and mobile IR is presented in [6], which deals with natural language processing on Android devices, too. However, also in this case, the focus is put on the implementation of a library to invoke remote web services to analyse natural language text. The book "Pocket Data Mining" [7] covers the highly interesting topic of distributed data mining on mobile devices and profoundly discusses the problems that come along with local (and mobile) data processing. In addition, it motivates these approaches and solutions by relevant real-world scenarios.

However, as indicated in Sect. 1, these approaches either neglect the fact that modern mobile devices are capable to handle large amount of data on their own or focus on the (nevertheless important) analysis of structured data only. However, the autonomous processing of unstructured textual data on mobile devices is relevant, too.

For this purpose, the Android OS is particularly suited as a target platform as

- the Android OS largely consists of open-source software and dominated the smartphone OS market with 86.8% share in Q3 of 2016 [8],
- Android apps are usually written in Java, making it possible to integrate existing Java-based libraries for natural language processing with minor adaptations and without having to completely reimplement them for the mobile application scenario and
- Android natively offers application programming interfaces (APIs) for parsing HTML files and with the library *SQLite* (https://www.sqlite.org/) an efficient solution for the persistent storage of mobile application data. A useful extension to *SQLite*, *SQLite FTS* (Full Text Search) (https://www.sqlite.org/fts3.html), enables the creation and query of inverted (word) indexes and can be used in Android as well, making it perfectly suited for the task at hand to create a functional full-text search solution for Android.

So far, none of the mentioned approaches and practical solutions offer a holistic method to handle natural language text, from its extraction to its retrieval (to say nothing of semantic analysis), right on the users' devices. All apps use remote servers to process queries and analyse textual contents. For iOS, the app *Spotlight* exists that at least enables the search for contacts, addresses, appointments, music and e-mails. With *WhereDat - Enterprise Search*, a similar solution for Android is available. Taken by themselves, these

features are useful. However, a future challenge is to combine, semantically connect and correlate this information in order to actually transform these reactive solutions into "intelligent" assistants that autonomously and proactively search for, prepare and present needed information. The herein presented app *Android IR* is just a first, yet important, step towards this goal.

3 The Mobile App *Android IR*

In order to realise the mobile full-text search app *Android IR*, the selection of appropriate existing software components to handle text documents on mobile devices was necessary as it is not feasible to rewrite the functionalities of tools. For this purpose, several applicable native (preferred) and third-party software libraries have been investigated and evaluated regarding their applicability in the presented mobile scenario. This section will therefore introduce some of them. Emphasis is put on the indexing and searching of text on Android devices.

3.1 Software Components

MIME-Type Identification and Text Extraction. The first important task in automatic text processing is the extraction of textual data from given documents in various formats. As it should be possible to extract text from plain text documents as well as from PDF- and HTML-files within the app, the library *Apache Tika* has been found suitable to detect the correct MIME-type of those files. However, as one goal during the design of the software was to reduce third-party dependencies, Android's own class android.webkit.MimeTypeMap has been found reliable and finally selected for this task.

In order to extract text from HTML-files, Android's native class android.text.Html is a suitable solution. However, the selective access to specific DOM-elements and metadata in them is not possible. The well-known library *Jsoup* (https://jsoup.org/), however, offers this functionality and is compatible with Android. Moreover, in direct comparison with Android's native HTML-parser, *Jsoup*has shown a higher performance (the average throughput has been measured for both parsers and a number of given files) when analysing HTML-files larger than 10 kB.

More complicated is the extraction of text from PDF-files as their structure usually does not reflect the actual visible layout when displaying them. This discrepancy is addressed by a number of scientific publications such as [9] and determines the complexity of respective parser libraries. For the herein discussed task, the libraries *Android Apache PDFBox*, a port of *Apache PDFBox* for Android with no incompatible dependencies, and *iText* (http://itextpdf.com/), whose Android compatibility is explicitly advertised, are possible candidates. In tests, it was found that *Apache PDFBox* is unsuited for an application in *Android IR*. The main reason for this assessment is that this library often caused application crashes due to high memory consumption (OutOfMemoryException). Furthermore, its overall performance when

analysing PDF-files was much lower than *iText's*. Therefore, *iText* has been selected for application in *Android IR*.

Analysing, Indexing and Searching Text. After the successful text extraction (typically on document level), the next common, yet optional, step of language-dependent text analysis is carried out. Respective software components first try to segment text into smaller chunks (called tokens) such as sentences, words and phrases. Then, stop words, usually short function words that carry no meaning, are removed as well. However, if it necessary to be able to search for phrases with stop words, this step must be skipped. Afterwards, the step of base form reduction and/or lemmatisation is executed which is backed by the so-called part-of-speech tagging (POS tagging), which assigns a lexical item (usually a word) to a lexical class such as noun, verb, adjective etc. These steps make it possibly to select particular classes and sequences of lexical items to be indexed. A well-known tool for these purposes is *Apache OpenNLP*. After the analysis phase, the indexing step takes these items and persistently saves them for later retrieval. Usually, a so-called inverted index is created in this process. During retrieval (search), this index is queried using keywords and the results (usually the documents that contain them) are ranked in descending order according to their relevance to the query and finally presented to the user.

The Java-based library *Apache Lucene* is a holistic solution to develop efficient IR-applications that also offers analysis functions such as stop word removal, base form reduction and query expansion for a variety of languages. It is possible to make use of this library in Android applications as well. Here, *Lucene* 4.7 is the latest usable version as starting from *Lucene* 4.8, file-based operations rely on the Java NIO 2 API which is missing under Android.

However, as indicated in the previous section, the full-text search component *SQLite FTS* is natively included in Android. Besides its capability to search for single words (terms) and n-grams of arbitrary length with optional wildcard characters, multiple search terms can be concatenated using Boolean expressions as shown in Fig. 1.

Although *SQLite FTS* offers the possibility to include external analysis methods by defining own tokenisers, base form reduction for the English language using the Porter stemming algorithm is the only further analysis function included. Even a function for stop word removal is missing and must be manually added if needed. Therefore, its capabilities are very limited compared with and in contrast to *Lucene's*. However, in tests, *SQLite FTS* outperformed *Lucene* during search to a great extent and has therefore been selected for usage in *Android IR*[1].

3.2 Further Features

Besides the optimisation of the important index and search functionalities, power-saving features have been integrated as well which are helpful in conjunction

[1] Interested readers can download *Android IR* (16.8 MB; installation of apps from unknown sources must be allowed in security settings) from: http://www.docanalyser.de/androidir.apk.

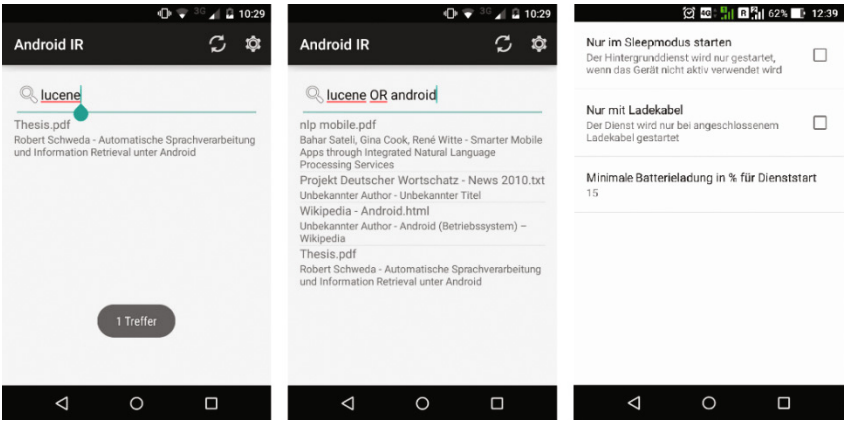


Fig. 1. Screenshots of *Android IR*: search result lists and energy options

with Android's own power management features. As it can be seen in Fig. 1, the user can activate energy options to only start the indexing service (which runs in the background even when the app's main activity is not visible) when the device is in sleep mode or connected to a charger. Furthermore, the service is only allowed to run when the current battery level is above a given threshold (default: 15%). In this context, the option to set the maximum number of pages of a PDF-file to be extracted (in menu "text extraction") is helpful, too. Even more, in general settings, the user can specify the folders to be indexed as well as the maximum file size and the maximum processing time (for text extraction and indexing) in seconds allowed for one file. These options are, nevertheless, also intended to keep the local index small, whose maximum size can be adjusted in menu "indexing". Here, the user can also delete the entire index and select or deselect the option for stop word removal which is activated by default.

4 Experimental Evaluation

In order to evaluate the libraries' performance, a fixed test environment had to be established to make the final decision on which libraries to finally use in *Android IR*. For this purpose, a special benchmark app (see Fig. 2) has been developed with all candidate libraries included.

All tests have been performed on devices with a freshly installed operating system and activated flight mode in order to minimise power consumption of possibly installed applications, services and CPU-intensive network connections such as incoming software updates. Moreover, the device screen has been turned off during the tests and an initially fully charged battery has been used (no charger was connected) in order to measure the app's power consumption.

The test datasets consisted of 1459 HTML-files (331.5 MB) from the German *Wikipedia* (http://de.wikipedia.org) and *Projekt Gutenberg* (http://gutenberg.spiegel.de) as well as of 178 PDF-files (899.1 MB) consisting of converted articles from the

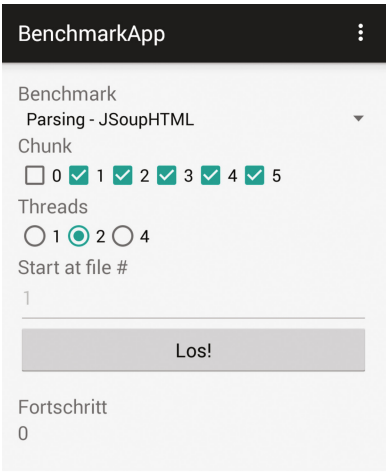


Fig. 2. Screenshot of benchmark app

German *Wikipedia*. Furthermore, 1676 text files (797.5 MB) from the *Projekt Deutscher Wortschatz* (http://wortschatz.uni-leipzig.de) consisting of online news articles. Three unmodified "Google Nexus" test devices with 2 GB of RAM have been selected. Particularly, the LG Nexus 4 (with Android 4.4.4 and upgraded to Android 5.0.1), the HTC Nexus 9 (with Android 5.0.1) and the Samsung Nexus 10 (with Android 4.4.4) have been used. For a full description of all tests conducted during the development of *Android IR*, the reader is pointed to [10].

Text Extraction. As a first result discussed herein, the authors found out that the library *Jsoup* for text extraction from HTML-files is much more performant than Android's HTML-parser. To come to this conclusion, the processing time and throughput in [kb/s] have been measured on a Nexus 10 while using 1 thread only. This data is given in Table 1.

Table 1. Average processing time per file and throughput of *Jsoup* and *Android HTML*

Jsoup		Android HTML	
Average processing time in (s)	Throughput in (kb/s)	Average processing time in (s)	Throughput in (kb/s)
0,95	**220,54**	3,38	63,59

As indicated above, the library *Android PDFBox* mostly failed to successfully finish its task of extracting text from PDF-files due to unexpected program crashes caused by high memory consumption, whereas *iText* in 78% of all cases succeeded. However, also *iText* sometimes could not finish this task. The main reason were problems with the file structure and content causing the respective files to be skipped.

Indexing and Searching Text. In order to compare the performance of *Lucene* and *SQLite FTS* during indexing and searching text documents, the same hardware and settings as in the previous tests have been used.

In Table 2, it can be seen that during indexing, the average processing time per file in [s] is much lower for *SQLite FTS*. Accordingly, its throughput is much higher than *Lucene's*. However, the size of *Lucene's* index created is much smaller (only about 1/4 of the original data), whereas the size of the *SQLite FTS* index is about 1,3 times larger than the original amount of data. This is due to the redundant storage of text in both indexed and full-text form. The activated function of stop word removal can, at least, reduce the index size by about 30%.

Table 2. Indexing: average processing time per file and throughput of *Lucene* and *SQLite*

SQLite FTS		Lucene	
Average processing time in (s)	Throughput in (kb/s)	Average processing time in (s)	Throughput in (kb/s)
0,72	**480,04**	2,94	124,72

Even more interesting is the performance of the search capabilities. As Table 3 shows, *SQLite FTS* greatly outperforms *Lucene* on all test devices and all Android versions. The search performance on Android 5.0.1 is also higher than on Android 4.4.4 due to the new runtime environment Android Runtime (ART) and improved garbage collection. Because of these results, the libraries *Jsoup*, *iText* and *SQLite FTS* are used in the final app *Android IR*.

Table 3. Searching: average processing time per file when searching with *Lucene* and *SQLite FTS* on the Nexus test devices

	N4 (4.4.4)	N4 (5.0.1)	N9	N10
Lucene	2479	972	598	2410
SQLite FTS	30	**22**	**22**	33

Further Results. Another significant performance improvement can be obtained by a concurrent execution of the mentioned operations using more than one threads [10]. However, it is sensible to limit the maximum number of threads to the number of CPU cores available to keep the device responsive.

Furthermore, the overall processing time for indexing all available HTML- and PDF-files along with the power consumption on the three test devices is given in Table 4. Here, it is noteworthy that the devices running on Android 5.0.1 could index all files much faster than the Nexus 10. However, as this device is a tablet equipped with a 9000 mAh battery, the power consumption amounted to only 4% despite the large processing time. The Nexus 9 used the less time for this task and the battery level decreased by only 3%.

Table 4. Indexing: overall processing time for all HTML- and PDF-files and power consumption on the Nexus test devices

	Nexus 4 (5.0.1)	Nexus 9	Nexus 10
Processing time	35 min	**26 min**	1 h 17 min
Power consumption	13%	**3%**	4%

5 Extending *Android IR*

All of the previous results clearly show that information retrieval and the handling of text files is both efficiently and effectively possible on modern Android devices. This is why current works are dedicated to extend *Android IR* to turn it into a powerful "mobile secretary" which can autonomously combine, analyse and correlate textual data from various sources in order to act as a helpful personal information manager (PIM), e.g. extract appointments and remind the user of them. For doing so, it is necessary to be able to correctly identify named entities such as mentioned persons, organisations, locations as well as date- and time-related data. The already mentioned library *Apache OpenNLP* would be good choice for this task. Its applicability as part of a mobile app is currently being investigated. Furthermore, the usage of text mining methods to improve the app's search functionalities would be beneficial. As an example, state-of-the-art graph-based methods for term clustering [11] as well as local search word extraction and query expansion [12] could be applied which mostly work on word or term level.

However, although it seems to be natural to represent information on entities and their relationships using graphs [13], the logical need to persistently store and efficiently traverse them directly on the mobile device presents a problem. Up to now, there are no graph database systems such as Neo4j (https://neo4j.com/) for mobile platforms available. As the demand for autonomous data analysis solutions on mobile devices will definitely grow in the future, the development of such systems is necessary, not only in the context of IR solutions. From the technological point of view, it is clearly possible (and sensible) to realise efficient and effective mobile search solutions that are backed by both flexible natural language and text mining tools as well as local database systems which do not necessarily have to be relational ones. In this regard, the app *Android IR* is just a first, nevertheless important, step towards a holistic search solution for mobile devices.

6 Conclusion

This paper introduced *Android IR*, a mobile app for effective and power-saving full-text search on Android devices. Its software components and features have been described in detail and experimentally evaluated. Also, extensions to this app have been discussed. Further research will be conducted on how to turn *Android IR* into a powerful assistant for the mobile and personal information management by making use of state-of-the-art text analysis methods and tools.

References

1. Statista: Daten und Statistiken zu WhatsApp (2016). https://de.statista.com/themen/1995/whatsapp/
2. Novet, J.: Facebook says people sent 63 billion WhatsApp messages on New Year's Eve (2017). http://venturebeat.com/2017/01/06/facebook-says-people-sent-63-billion-whatsapp-messages-on-new-years-eve/
3. Statista: Monatliches Datenvolumen des privaten Internet-Traffics in den Jahren 2014 und 2015 sowie eine Prognose bis 2020 nach Segmenten (in Petabyte) (2016). https://de.statista.com/statistik/daten/studie/152551/umfrage/prognose-zum-internet-traffic-nach-segment/
4. Wachsmuth, H.: Text Analysis Pipelines: Towards Ad-Hoc Large-Scale Text Mining. Springer, Cham (2006)
5. Tsai, F.S., et al.: Introduction to mobile information retrieval. IEEE Intell. Syst. **25**(1), 11–15 (2010)
6. Sateli, B., Cook, G., Witte, R.: Smarter mobile apps through integrated natural language processing services. In: Mobile Web and Information Systems: 10th International Conference, MobiWIS 2013, pp. 187–202. Springer (2013)
7. Gaber, M.M., Stahl, F., Gomes, J.B.: Pocket Data Mining: Big Data on Small Devices. Springer, Cham (2014)
8. International Data Corporation: Smartphone OS Market Share, 2016 Q3 (2016). http://www.idc.com/promo/smartphone-market-share/os
9. Ramakrishnan, C., et al.: Layout-aware text extraction from full-text PDF of scientific articles. Sour. Code Biol. Med. **7**(1), 7 (2012)
10. Schweda, R.: Automatische Sprachverarbeitung und Information Retrieval unter Android. Master's thesis, FernUniversität in Hagen (2015)
11. Biemann, C.: Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In: Proceedings of the HLT-NAACL-06 Workshop on Textgraphs-2006, pp. 73–80. ACL, New York City (2006)
12. Kubek, M., Unger, H., Loauschasai, T.: A quality- and security-improved web search using local agents. Intl. J. Res. Eng. Technol. (IJRET) **1**(6) (2012)
13. Efer, T.: Text mining with graph databases: traversal of persisted token-level representations for flexible on-demand processing. In: Autonomous Systems 2015, Fortschritt-Berichte VDI, vol. 10, no. 842, pp. 157–167. VDI-Verlag, Düsseldorf (2015)

Sequentially Grouping Items into Clusters of Unspecified Number

Maytiyanin Komkhao[1], Mario Kubek[2], and Wolfgang A. Halang[2(✉)]

[1] Faculty of Science and Technology,
Rajamangala University of Technology Phra Nakhon, Bangkok, Thailand
maytiyanin.k@rmutp.ac.th
[2] Faculty of Mathematics and Computer Science, Fernuniversität in Hagen,
Hagen, Germany
{mario.kubek,wolfgang.halang}@fernuni-hagen.de

Abstract. When run, most traditional clustering algorithms require the number of clusters sought to be specified beforehand, and all clustered items to be present. These two, for practical applications very serious shortcomings are overcome by a straightforward sequential clustering algorithm. Its most crucial constituent is a distance measure whose suitable choice is discussed. It is shown how sequentially obtained cluster sets can be improved by reclustering, and how items considered as outliers can be removed. The method's feasible applicability to text analysis is shown.

Keywords: Clustering · Number of clusters · Distance measures · Sequential clustering · Single-linkage · Reclustering · Outlier removal · Text analysis

1 Introduction

Clustering is successfully used in exploratory pattern analyses, in data mining, machine learning and in pattern classifications to build concise models of large datasets. The effect of clustering is to group individual items in such a way that the values of their corresponding feature vector components have high similarity to one another within the same cluster, but are rather dissimilar to the components' values in other clusters. An abundance of clustering algorithms has been devised [8], of which the classical and most widely used ones are *k-Means* and, although a classifier by its nature, k-Nearest-Neighbours (*k-NN*).

Most clustering algorithms including k-Means and k-NN require to specify and fix right from the very beginning the number of clusters to be generated for a given dataset. This is too serious a restriction for important application areas such as general recommender systems, because it necessitates visualisation of

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_28

the underlying datasets and intervention by human experts prohibiting recommender systems to be offered on a continuous basis and automatically operated in an unattended mode. Hence, to adequately build cluster models reflecting the characteristics of given settings, the number of model elements must be adjustable and, thus, employed clustering algorithms are only suitable if they can determine the number of a model's clusters themselves.

Indeed, hierarchical clustering—both agglomerative [11] and divisive—is able to dynamically determine the number of clusters modeling a given dataset. It suffers, however, from another drawback impairing its applicability for many practical purposes, viz. that a set of items to be clustered must be available for processing in its entirety. In contrast to this, the items considered by recommender systems are added one by one, and such systems are expected to be operational all the time and permanently available as web services.

Although incremental clustering algorithms such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN), a faster variant of it considering the density of databases [7], or one for information retrieval purposes based on hierarchical agglomeration and considering the maximum cluster diameters regardless the employed distance function [4] do process one item at a time, they adhere to a priori specified numbers of clusters.

To eliminate these two weaknesses of clustering methods, in this paper a heuristic algorithm will be presented, which is able to continuously form cluster models of sequentially arriving items with the number of clusters being adjusted when need be. The algorithm is based on a graph-theoretical and a point-density interpretation of the feature vectors' locations.

Both in proper clustering and in matching feature vectors with the constituents of cluster models, the measures for distance or similarity, respectively, are also quite decisive elements. Since sequential clustering with not a priori prescribed numbers of clusters is the topic of this paper, we do not specify certain distance functions here, but discuss some aspects to be considered in making suitable choices and give a recommendation.

Efficiency and accuracy of modelling also depend highly on how well a model's clusters capture the intrinsic characteristics of the underlying dataset in feature space, and whether this representation is free of redundancies. To this end, outliers may be removed from the input data if the latter are known to be susceptible to noise or errors such as measured values. Therefore, in the sequel it will also be considered how to remove outlying feature vectors and, based on this, how to obtain higher-density and lower-volume clusters.

2 Measuring Distances and Similarity

A plentitude of applications requires to determine the distance of items in feature spaces of any kind. Often distance measures are employed to find item agglomerations (clusters) which are, in turn, used to form classifications of objects and behavioural models of certain phenomena. Items are called most similar, when a distance between them is minimised. When items are similar to a certain

extent, they are often grouped into a common cluster, and dissimilar ones are grouped into different clusters. Employing the metrics induced by the vector norms $\|\cdot\|_m,\ m = 1, 2, \infty$ silently assumes that the component spaces are more or less equal, and that the attributes are totally unrelated.

In most application domains, however, neither the classical metrics are feasible to measure distance nor are the items' attribute spaces similar or, at least, numeric. On the contrary, the components of multi-dimensional feature spaces may be as heterogeneous as continuous set of numbers, discrete sets, Boolean sets, fuzzy sets, structures, graphs, or even continuous functions such as spectra and many others more. Consequently, suitable distance measures can only be selected on the basis of sound knowledge of the particular application domains and of the real semantics of the data [2, p. 26], and utmost care must be exercised when combining different and mutually independent quantities with different physical dimensions in a single arithmetic expression finally giving rise to just a single number. Moreover, in most cases it will be impossible to find an optimal distance measure.

According to the large variety of attribute types and scales, distance measures must be chosen very carefully. Preprocessing may be needed to transform the characteristics of natural phenomena into feature spaces where a notion of distance can be defined. Generally, the physical dimensions of the different attribute spaces should be transformed to similar scales. A distance function $d\ :\ F \times F \rightarrow \mathbb{R}_+$ on a feature space F must have the properties $d(x, x) = 0$, $x \in F$ and $d(x, y) = d(y, x),\ x, y \in F$, but does not need to be a metric, i.e. the triangular inequality $d(x, y) + d(y, z) \geq d(x, z),\ x, y, z \in F$ is not required to be fulfilled. A suitable distance function does not even have to be continuous.

An empirical study [3] has revealed that for feature spaces with numerical components the Manhattan metric expresses the notion of distance rather well. In comparison to the other metrics based on the vector norms $\|\cdot\|_m,\ m > 1$, for many applications it leads to results of the same or even higher quality, but requires less computational effort. Hence, it is advisable to structure knowledge-based distance functions similar to the Manhattan metric by taking absolute values of two items' attribute differences, but then normalise and multiply them with positive factors to express different weighting of the component spaces in the subsequent summation yielding just one number as distance. For non-numerical attributes, component differences must be defined analogously, e.g. as 1 or 0 when discrete values coincide or not.

3 A Heuristic Algorithm for Sequential Clustering

To form cluster models $\mathbf{M}$ of data points with the number of resulting clusters not set a priori, particularly in application domains where the sets of data points are not available at one time, but the data points are arriving one by one and the cluster models are to be built incrementally, we suggest to employ the following heuristic algorithm, which determines appropriate numbers of clusters itself, i.e. it comprises preclustering as an integral part. The algorithm works sequentially

on a set of data points or feature vectors **F**, either available upon its initialisation or growing by arriving new feature vectors joined with the set. Comparing a feature vector under consideration with known clusters, the algorithm either associates the vector with the cluster matching best, called the winning cluster, already existing in the corresponding model and updates the cluster's parameters accordingly, or it inserts the vector into the model as a new cluster.

Initialisation: Given an input vector f_1, which may be selected randomly in **F** for $|\mathbf{F}| > 1$, let the cluster $\{f_1\}$ form the model **M** initially.
Loop: Execute for any newly arriving or for all further feature vectors $f \in \mathbf{F}$:

1. Calculate the membership of f in all clusters of **M**.
2. Determine the winning cluster as the one for which f assumes the highest membership value.
3. **If** the value of f's membership in the winning cluster does not exceed a given threshold,
   then merge f with the winning cluster,
   else extend the model by a new cluster containing just f ($\mathbf{M} := \mathbf{M} \cup \{f\}$).

The decisive aspect of this algorithm is determining a feature vector's membership in clusters. For this, a distance measure as discussed in the last section will be used. It still remains a design choice to which point in a cluster the distance from a vector is considered. One could, for instance, compare the vector's distances to the centroids of a model's clusters.

Another choice [2, pp. 298–307] is to decide on cluster membership—as in step (3) above—based on the vector's distances to its nearest neighbours in each cluster, respectively, which is called single-linkage method in the literature. Experience revealed that the straightforward and very simple algorithm above works quite satisfactorily with this choice. Clusterings generated are also rather robust with respect to the distance measure and the threshold selected, because not the absolute distance values are crucial, but only their order. The algorithm cannot only recognise circular or ellipsoidal clusters, but also quite differently shaped ones, e.g. with branches, curves or elongate, and two members of a cluster are always connected by a path fully contained in the cluster. The method thus emphasises connectivity of items rather than their similarity.

The last-mentioned property follows from a graph-theoretical interpretation of this kind of clustering. Let a complete graph be formed with the elements of a data set to be clustered as vertices, and the edges between any two vertices weighted by their distance. Removing from the complete graph all edges weighted by distances exceeding a given threshold yields a subgraph whose connectivity components, i.e. the sets of vertices linked by edges contained in the subgraph, are identical with the clusters produced by the algorithm.

4 Outliers and Reclustering

In order to model the intrinsic characteristics of given sets of feature vectors with as low redundancy as possible, model-constituting clusters should be rather

densely filled with data points and should have clear boundaries. Although sequential clustering according to the single-linkage method is able to cope with clusters of a large variety of shapes, it can also lead to the undesirable property of connecting rather dissimilar agglomerations of homogeneous items by chains of intermediately located items and placing them into common clusters. Since sequential clustering is unsuitable to recognise outliers and to form compact clusters all the time, it is advisable to postprocess item sets in an integral way.

An approach to do so is based on a probabilistic interpretation, which considers feature vectors of items as observations of a mixed population constituted by several overlapping populations, the sum of whose single unimodal distribution densities is a multimodal distribution density, i.e. has several local maxima. Under the condition, that the single populations are sufficiently separated, it is assumed, that the local maxima characterise the regions in feature space where the single populations are concentrated, i.e. where clusters are expected.

Based on this interpretation, the method proposed in [12] is able to detect clusters of very complex shapes—just like sequential clustering as discussed above. According to the method those locations in feature space are searched, where a given data set exhibits local point concentrations with higher densities than in the respective vicinities. The search works by iteratively translating with a small step-size all feature vectors towards regions of higher point density. By this process the vectors gradually approach the local maxima. Merging into a single cluster all feature vectors thus arriving in the neighbourhood of a certain location, an exhaustive and disjoint clustering of the data set is produced, with the number of these clusters derived from the characteristics of the data set, but not specified a priori.

Let the set $\mathbf{F} = \{f_1, ..., f_n\}$ of n feature vectors with m dimensions and a distance function $d : F \times F \rightarrow \mathbb{R}_+$ be given. With the variance σ the gradient used in the above-mentioned translation of an $f \in \mathbf{F}$ is defined as

$$\nabla f = \frac{1}{(2\pi)^{m/2}\sigma^{m+2}} \sum_{i=1}^{n} (f_i - f) \cdot exp\left[-\frac{d(f, f_i)^2}{2\sigma^2}\right] \tag{1}$$

The scaling parameter σ shapes the Gaussian distributions occurring in this expression. It has to be selected carefully as it determines number and contents of clusters. A clustering appearing "natural" for a certain data set may be sought running the above method [12] for a variety of σ values. When the number of clusters remains constant over a relatively wide range of σ, such a clustering reflecting the data set's intrinsic properties is assumed to be found.

As Gaussian functions are idealised, but computationally demanding approximations of real distributions, and since their values are negligible short distances away from their centres, in the method of [12] we replace Gaussian by other bell-shaped and very similar looking curves, namely B-splines. With

$$(x)_+^k := \begin{cases} x^k, & x \geq 0 \\ 0, & x < 0 \end{cases}, \quad k \in \mathbf{N}$$

they are defined by

$$b_k(x) = \frac{1}{k!} \cdot \sum_{i=0}^{k+1} (-1)^i \cdot \binom{k+1}{i} \cdot \left(x + \frac{k+1}{2} - i \right)_+^k, \; x \in \mathbf{R} \tag{2}$$

In general, it will not be necessary to employ B-splines of high degree k, but the bell-shaped ones of lowest degrees will suffice, i.e. the cubic ($k = 3$) or even the just once continuously differentiable quadratic ($k = 2$) B-spline (cp. Fig. 1).

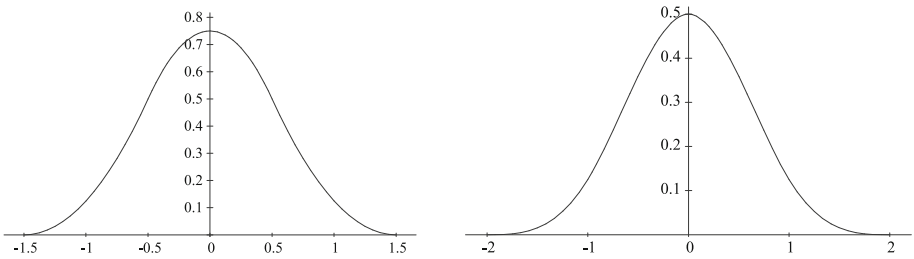


Fig. 1. B-splines of degrees 2 and 3

The method described in [12] lends itself for outlier removal as, after running it, the clusters with low point density are simply removed. This yields a more concise disjoint, but non-exhaustive clustering of the original item set.

Another approach to outlier removal is to eliminate all items from this set in whose neighbourhoods lie only very few, if any, other items. The resulting subset can then be reclustered by the sequential single-linkage method above. The clusters obtained will not contain any dissimilar agglomerations of homogeneous items connected by chains of intermediately located items anymore.

5 Case Study: Clustering Text Documents

The heuristic algorithm presented above can be regarded as a generic solution to group arbitrary kinds of data objects. It performs its task without relying on the number of clusters to be generated as an input parameter, which is usually guessed by an experienced human domain expert. The advantage of not having to estimate this parameter is especially beneficial when information on the heterogeneity or homogeneity of data objects is insufficient. This is particularly true for the domain of automatic text analysis. For instance, a book such as conference proceedings could cover a variety of major and minor topics with each one having its own domain-specific terms. It would be—even depending on the granularity required—hard to estimate the correct number of such topics.

Traditionally, data objects to be clustered are given as vectors with weighted features, making it easily possible to determine their distance or similarity in Euclidean space by applying standard measures such as Euclidean distance or

cosine similarity. In text processing, however, it is also very common to represent them (mostly terms or documents) and their semantic relationships by graphs. For instance, the nodes in so-called co-occurrence graphs usually represent terms, and the (usually undirected) weighted edges indicate semantic relationships between them as well as their significance. Normally, an edge is only drawn when the respective terms co-occur frequently, e.g. on sentence level. In order to determine the significance of co-occurrences using a weight function $g(w_a, w_b)$ for any two co-occurring words w_a and w_b, measures such as the Dice coefficient [5], the Poisson collocation measure [10] or the log-likelihood ratio [6] can be applied. A distance $d(w_a, w_b)$ between w_a and w_b is then defined by

$$d(w_a, w_b) = \frac{1}{g(w_a, w_b)} \tag{3}$$

The distance d of any two nodes (terms) w_a and w_b in a fully connected co-occurrence graph G is obtained by computing the shortest path between them:

$$d(w_a, w_b) = \sum_{i=1}^{k} d(w_i, w_{i+1}) \;\rightarrow\; minimum \tag{4}$$

with $d(w_a, w_b) = \infty$ in case of a partially connected co-occurrence graph.

Unsupervised graph-based clustering approaches such as the Chinese Whispers algorithm [1] can efficiently find useful clusters of semantically connected terms (topics) in co-occurrence graphs. This algorithm relies on a label propagation technique and—just like the algorithm presented in the previous section—does not require a pre-set number of clusters/topics for this purpose. However, in order to achieve the same result on the document level, and taking into account the valuable term relations found in co-occurrence graphs, a new measure to determine the semantic distance between text documents is needed.

By analogy with the physical centre of mass of complex bodies consisting of several single mass points, it was shown in [9] that text documents can be represented by their centroid terms found in a preferably large and topically well-balanced co-occurrence graph G (which acts as reference corpus). In order to determine the centroid term of a document D using G, the distance $d(D, t)$ between a given term $t \in G$ and D containing N words $w_1, w_2, .., w_N \in D$ reachable from t in G must be computed by

$$d(D, t) = \frac{1}{N} \sum_{i=1}^{N} d(w_i, t) \tag{5}$$

Thus, $d(D, t)$ returns the average length of the shortest paths between t and all words $w_i \in D$ that can be reached from it in G. Note that—differing from many methods found in the literature—it is not assumed that $t \in D$ holds. The term $t \in G$ is called the centre term or *centroid term of* D when $d(D, t)$ is minimal. Thus, the semantic distance ζ between any two documents.

$$\zeta(D_1, D_2) = d(t_1, t_2). \tag{6}$$

D_1 and D_2 with their respective centroid terms t_1 and t_2 in G can be expressed as $\zeta(D_1, D_2) = d(t_1, t_2)$. The centroid terms obtained this way generally represent their documents very well. This distance measure is also able to detect a similarity between topically related documents that, however, do not share terms or have only a limited number of terms in common. The cosine similarity measure (when relying on the bag-of-words model) would not be able to accomplish this. Generally, the 25 most frequent words of a medium-sized document, such as a Wikipedia article, are sufficient to properly determine its centroid term. Thus, it is very well possible to determine suitable centroid terms of short documents or even search queries.

As a precondition to successfully apply the sequential clustering algorithm on text documents, it must be examined first whether this new distance measure for documents can actually find pairs of semantically close documents. Also, there are some noteworthy and general remarks to be made when doing so:

1. the data objects to be clustered (the text documents) are represented by only one feature (the centroid term),
2. the distance measure operates on a non-Euclidean space (the Euclidean metric cannot be applied, because the data points are not assigned a coordinate in a multi-dimensional space) and
3. for the co-occurrence graph generated, the triangular inequality does not hold (unequal node distances).

Thus, the aim of the following experiment was to show that most of a reference document's k closest neighbours according to the centroid distance measure share its topical category. The experiment was carried out 100 or 200 times, respectively, for all documents in the following two datasets, whereby each document in these sets was used as reference document. The datasets consist of on-line news articles having appeared between September and November 2015 in the German newspaper "Süddeutsche Zeitung". Dataset 1.1 contains 100 articles covering the topics 'car' (25), 'money' (25), 'politics' (25) and 'sports' (25); dataset 1.2 contains 200 articles on the same topics with 50 documents for each topic. The articles' categories (tags) were manually set by their respective authors. On the basis of these assignments (the documents/articles to be processed act as their own gold-standard for evaluation), it can easily be found out how many of the k nearest neighbours of a reference document according to the centroid distance measure share its topical assignment (needless to say that these topical tags were not considered by the distance measure). The desired result is that this number is as close to $k = 5$ or $k = 10$, respectively, as possible. For this purpose, the fraction of documents with the same topical tags was computed. Furthermore, linguistic preprocessing was applied on the documents to be analysed, whereby stop words were removed and only nouns (in their base form), proper nouns and names were extracted. In order to build the undirected co-occurrence graph G (as reference for the centroid distance measure) using all documents of these two

datasets[1], co-occurrences on the sentence level were extracted. Their significance values were determined using the Dice coefficient [5].

As an interpretation of Table 1, for dataset 1.1 and $k = 5$, on average the centroid distance measure returned 3.9 documents with the reference document's topical assignment first. For $k = 10$, on average 7.6 documents shared the reference document's tag. In both cases the median is even higher. Similar results were obtained for dataset 1.2.

Table 1. Average number of documents sharing the reference documents' category with their $k = 5,\ 10$, respectively, most similar documents

	$k = 5$		$k = 10$	
	Aver. no. of doc.	Median	Aver. no. of doc.	Median
Dataset 1.1	3.9	5	7.6	9
Dataset 1.2	3.9	5	7.5	9

These good values indicate that it is indeed possible to identify semantically close documents with the centroid distance measure. Furthermore, the measure is able to group documents with the same topical tags. Its application in classification systems considering nearest neighbours seems therefore beneficial. The findings further suggest that the centroid distance measure can successfully be applied in document clustering methods, too. Although they represent documents, centroid terms are basically nodes in the co-occurrence graph G used. This means that graph-based clustering algorithms applied on G are inherently able to return both term clusters and document clusters at the same time.

Having said this, the heuristic clustering algorithm presented here—maybe due to its graph-theoretical background outlined above—is well-suited to be used in conjunction with the centroid distance measure, too. In doing so, however, there are two questions remaining to be answered:

1. How can the membership value be calculated?
2. How to set the threshold to assign a document to the cluster with the highest membership value (or to create a new cluster)?

To answer the first question, the membership values for all existing clusters can be computed by determining, in G, the average distance between a document (centroid term) and all centroids existing in a cluster during the algorithm's execution. The threshold's (in the sense of a distance) value, however, is the most important factor to influence the size (number of assigned documents) and the overall number of clusters generated. A high value will likely lead to few large clusters, whereas a low value will cause the generation of many small clusters. In an interactive document clustering solution, this value could be the only input

[1] Interested readers may download these datasets (1.3 MB) from http://www.docanalyser.de/cd-clustering-corpora.zip.

parameter needed from the users. With respect to implementation, one might think of a graphical element such as a slider by which users can easily adjust this value, causing the algorithm to recompute the clusters afterwards.

In fully unsupervised settings, however, this threshold must be determined automatically. For this, several approaches may be sensible. A fixed, semantically motivated, threshold for all centroid terms could be used. In the given graph-based setting, one could speak of a node's 'radius', in which it is likely to find similar documents. Another option is to make this threshold individually dependent, e.g. on the nearest neighbours of a centroid term. The average distance from the nearest neighbours to this term is a good indication for an actual cluster membership. In order to prevent a bias towards only these nearest neighbours, an additional factor should be multiplied with this average distance value to increase the mentioned 'radius' and, in doing so, be able to put more related documents in the same cluster. Future research will investigate these computation options in detail.

Furthermore, when using the given graph-based setting, a model M can initially be filled with two clusters each containing one of the two most distant centroid terms of the so-called antipodean documents in the co-occurrence graph G. Owing to their distance, it is very likely that they are topically unrelated, especially when G is large. For the remaining documents, the cluster assignment and creation can be carried out according to the algorithm presented.

6 Conclusion

Clustering is a means of exploratory pattern analysis and classification aiming to build concise models of large item sets. Most clustering algorithms' property to require the number of clusters sought to be specified beforehand, and all considered items to be present upon clustering, contradicts the exploratory nature of this process and constitutes a serious drawback for many practical applications, which are to operate automatically and continuously. Therefore, it was shown that these shortcomings can be overcome by a heuristic, straightforward and very simple, albeit rather powerful sequential clustering algorithm. After a larger number of items has been collected, it becomes possible to improve cluster models generated sequentially. For this purpose, a feasible algorithm determining an appropriate number of concise clusters by itself, and two approaches for removing items considered as outliers from the models were presented. As measures for distance and similarity are the factors most decisive for the success of clustering algorithms, it was advocated for founding them on domain knowledge and data semantics. As a case study the feasibility of applying a centroid distance measure and the sequential clustering algorithm to find and group semantically similar documents in text analysis was shown.

Acknowledgement. This work was supported by Rajamangala University of Technology Phra Nakhon.

References

1. Biemann, C.: Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In: HLT-NAACL 2006 Workshop on Textgraphs, pp. 73–80. Association for Computational Linguistics, Stroudsburg (2006)
2. Bock, H.H.: Automatische Klassifkation. Vandenhoeck & Ruprecht, Göttingen (1974)
3. Breuer, D.: Abstandsmaße für die multivariate adaptive Einbettung. MSc Thesis, Fernuniversität in Hagen (2014)
4. Charikar, M., Chekuri, C., Feder, T., Motwani, R.: Incremental clustering and dynamic information retrieval. SIAM J. Comput. **33**(6), 1417–1440 (2004)
5. Dice, L.R.: Measures of the amount of ecologic association between species. Ecology **26**(3), 297–302 (1945)
6. Dunning, T.: Accurate methods for the statistics of surprise and coincidence. Comput. Linguist. **19**(1), 61–74 (1993)
7. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231. AAAI Press (1996)
8. Estivill-Castro, V.: Why so many clustering algorithms - a position paper. ACM SIGKDD Explor. Newsl. **4**(1), 65–75 (2002)
9. Kubek, M., Unger, H.: Centroid terms as text representatives. In: ACM Symposium on Document Engineering, pp. 99–102. ACM (2016)
10. Quasthoff, U., Wolff, C.: The Poisson collocation measure and its applications. In: 2nd International Workshop on Computational Approaches to Collocations, Vienna. IEEE (2002)
11. Rasmussen, E.: Clustering algorithms. In: Frakes, W.B., Baeza-Yates, R. (eds.) Information Retrieval: Data structures and Algorithms, pp. 419–442. Prentice-Hall, Upper Saddle River (1992)
12. Schnell, P.: Eine Methode zur Auffindung von Gruppen. Biometrische Zeitschrift **6**, 47–48 (1964)

Word2Vec Approach for Sentiment Classification Relating to Hotel Reviews

Jantima Polpinij(✉), Natthakit Srikanjanapert, and Paphonput Sopon

Intellect Laboratory, Faculty of Informatics, Mahasarakham University, Mahasarakham, Thailand
jantima.polpinij@gmail.com, nsrikanjanapert@gmail.com, paphonput@gmail.com

Abstract. In general, the existing works in sentiment classification concentrate only the syntactic context of words. It always disregards the sentiment of text. This work addresses this issue by applying Word2Vec to learn sentiment specific words embedded in texts, and then the similar words will be grouped as a same concept (or class) with sentiment information. Simply speaking, the aim of this work is to introduce a new task similar to word expansion or word similarity task, where this approach helps to discover words sharing the same semantics automatically, and then it is able to separate *positive* or *negative* sentiment in the end. The proposed method is validated through sentiment classification based on the employing of Support Vector Machine (SVM) algorithm. This approach may enable a more efficient solution for sentiment analysis because it can help to reduce the inherent ambiguity in natural language.

Keywords: Sentiment classification · Natural language · Word2Vec · Support Vector Machine

1 Introduction

Sentiment classification is to identify the emotional tendencies of the short messages that is to classify users' emotions into positive, negative, and neutral [1–4]. In the last decade of the 20th century, sentiment classification has attracted increasing research interest [1–4]. This is because e-commerce appears in society and social media provide a novel way to gather real time data in large quantities directly from users/customers. Therefore, very large amounts of information are available in on-line documents today. As the result, sentiment classification has become a significant research area [1, 4–6]. This is because it can help businesses quickly to classify and organize such as on-line reviews of goods and services. Also, it helps businesses to handle and have correct customer feedback.

In the previous study, most of the existing researches concentrated on the extraction of lexical features and syntactic features [7], while the semantic information of words are ignored [8–10]. Therefore, this work introduces a new method similar to word expansion or word similarity task, where this approach helps to discover words sharing the same semantics automatically. We apply Word2Vec technique to group the words having similar meaning into a same concept or class, and then the Cohen's Kappa

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_29

Statistics will be used to estimate it polarity of each concept. The proposed concept is validated through sentiment classification based on the employing of Support Vector Machine (SVM) algorithm. Through the experiment, we validate the effectiveness of our concept, by which we have performed a preliminary exploration of the sentiment analysis of hotel reviews in this paper.

The structure of this paper is as follows. Section 2 describes literature review. Section 3 describes our method. Experimental results are presented in Sect. 4. Finally, conclusions are made in the last section.

2 Literature Review

2.1 Sentiment Classification

Definition: Sentiment classification is a task of sentiment analysis (or opinion mining). It uses of natural language processing (NLP) and computational techniques to automate classification of sentiment from typically unstructured text [1–4]. It attempts to sort documents according to their subject matter (e.g., sports, politics, entertainments, books), where sentiment classification can describes the items in some detail and evaluate them as good/bad, preferred/not preferred, favorable/unfavorable [3–6]. Simply speaking, sentiment classification is a task to label a document according to the positive or negative polarity of its opinion [1].

Advantages: Sentiment classification would also be helpful in business intelligence applications and recommender systems [11]. This is because it can help businesses quickly to classify and organize such as on-line reviews of goods and services. Also, it helps businesses to handle and have correct customer feedback [1, 4–6]. Therefore, sentiment analysis is widely applied to reviews and social media for a variety of applications in order to get the real user/customer need. This information is very important for improving of product and service [11].

2.2 Related Works

Sentiment classification is a task of classifying n target item in a document to positive (good or favorable) or negative (bad or unfavorable) class. Previous researches mainly treated three kinds of target items: a word, a sentence and an overall document to positive or negative. Although sentiment classification is required and it is helpful in business, most existing tasks concentrate only the syntactic context of words but ignore the sentiment of text [7–9]. As this, many researchers correctly pay attention to study on the sentiment of text. Some of them have been proposed as follows.

Polpinij et al. [4] have presented a methodology, called concept-based sentiment analysis (C-SA). The main mechanism of the C-SA is Msent-WordNet (Multilingual Sentiment WordNet), which is used to prove and increase the results accuracy of sentiment analysis. By using the Msent-WordNet, all words in opinion texts having similar sense or meaning will be denoted and considered as a same concept. Indeed, concept-level

sentiment analysis aims to go beyond a mere word-level analysis of text and provide novel approaches to sentiment analysis that enables a more efficient solution from opinion text. This can help to reduce the inherent ambiguity and contextual nature of human languages. In final, the proposed methodology has been validated through sentiment classification.

In Tang et al. [8], they mentioned that most existing algorithms for learning continuous word representations typically only model the syntactic context of words but ignore the sentiment of text. Then, the word embedding learned by traditional sentiment classification are not effective enough for Twitter sentiment classification. This is because the traditional method typically only model the context information of words so that they cannot distinguish words with similar context but opposite sentiment polarity (e.g. good and bad). They address this issue by learning sentiment specific word embedding (SSWE), which encodes sentiment information in the continuous representation of words.

Zhang et al. [12] they have proposed a feature extraction method based on word embedding for this problem. They train word embedding by Word2Vec and model supplied by Stanford NLP Group. Also, prior statistical knowledge and negative sampling have been proposed and utilized to help extract the feature sub-space. They evaluated their model on WordNet synonym dictionary dataset and compare it to word2vec on synonymy mining and word similarity computing task, showing that their method outperforms other models or methods and can significantly help improve language understanding.

2.3 Related Theories

Word2Vec: Word2vec [4, 13] is one of the most successful ideas of modern statistical NLP, where it can associates words with points in space. Then word meaning and relationships between words are encoded spatially. As this, it can be seen that Word2Vec consists of two mains concepts. Firstly, similar words should be closer together, where their *"meanings"* are similar, denoted as *word* $w \rightarrow vec[w]$. Suppose $vec[good] = (0.1, -1.4)$, it means the word '*good*' point itself in space as a position vector (See in Fig. 1).

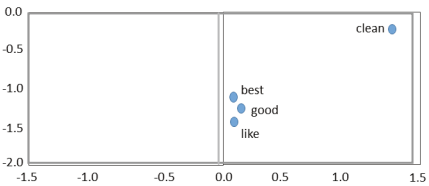


Fig. 1. A position of *vec*[*good*]

Secondly, the vector between the points of two words presents the word relationship. It means *the same word relationship* ⇒ *same vector*.

To learn the meaning of a word, suppose the word '*good*'. It can be calculated by the probability $P(w|good)$. Then, the word w nearby is considered. Simply speaking, Word2Vec learns from input text by considering each word w_0 in turn, along with its context C. The Word2Vec is an efficient combination of the *continuous bag-of-words* (*CBOW*) [13] and *skip-gram* (*SG*) [13] for computing vector representations of words. CBOW utilizes a window of word to predict the middle word, while SG uses a word to predict the surrounding ones in window.

Today, the Word2Vec can be used to significantly improve and simplify many NLP applications.

Cohen's Kappa Statistics: Cohen's kappa statistic (κ) is a measure of agreement between categorical variables X and Y [15, 16]. This work applies this technique to estimate the polarity of each specific sentiment word. Kappa is calculated from the observed and expected frequencies on the diagonal of a square contingency table [15]. Suppose that there are n subjects on whom X and Y are measured, and suppose that there are α distinct categorical outcomes for both X and Y. The formula to calculate Cohen's kappa for two raters can be:

$$\kappa = \frac{p_0 - p_e}{1 - p_e} = 1 - \frac{1 - p_0}{1 - p_e}$$

where

p_o is the relative observed agreement among raters.
p_e is the hypothetical probability of chance agreement.

If the raters are in whole agreement then $\kappa = 1$, there is no agreement among the raters other than what would be expected by chance (as given by p_e), $\kappa \leq 0$. Therefore, reliability happens when your data raters give the same score to the same data item.

3 Preliminary: A Corpus of Specific Sentiment Words

This section describes the process of collecting all specific sentiment words with their polarity. The method of the corpus development consists of three main steps. The overview of this method can be shown as Fig. 2.

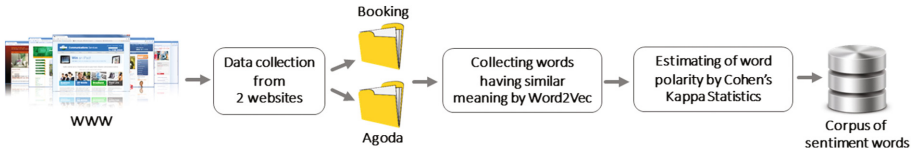


Fig. 2. The method overview for developing of the corpus of specific sentiment words

Firstly, it commences with the data collection. We collect textual hotel reviews relating to hotel from two main websites: www.agoda.com and www.booking.com. We have 20,000 textual reviews per website (10,000 positive reviews and 10,000 negative reviews), and each review should be at least 50 words.

The second step is to provide some specific sentiment words. In this initial work, 30 single words for positive sentiment and 30 single words for negative sentiment are provided, and then these words are used to find other words that may have similar meaning by the use of the Word2Vec technique. We use 10,000 textual reviews per website (5,000 positive reviews and 5,000 negative reviews) to learn sentiment specific words embedded in textual reviews. All similar sentiment words will be grouped as a concept. By using the Word2Vec, it can easily add a word to the vocabulary corpus. It is noted that the new published tool Word2Vec of Google is used to train word embeddings, where its training is extremely efficient [17]. Suppose we want to find other words related to the concept '*good*' though the use of 30 hotel reviews from two websites (Agoda and Booking). By using the Word2Vec tool, it returns the words '*like*', '*best*' and '*ok*' into the concept '*good*' because the tool predicts that these word are similar to '*good*'.

Finally, the sentiment polarity (*positive* or *negative*) estimation of each word is done by applying of the *Cohen's kappa statistic* (κ). Suppose the words related to the concept '*good*' is analyzed based on the use of 94 reviews from *Agoda* and *Booking* websites. Each sentiment word will be read and counted, represented as the matrix. In general, $0 \leq \kappa \leq 1$, although negative values do occur on occasion. Cohen's kappa is ideally suited for nominal (non-ordinal) classes. Weighted kappa can be calculated for tables with ordinal classes.

Consider the example in Fig. 3. We concentrate only *positive* and *negative* sentiment. Therefore, each polarity can be calculated as follows.

$$Polarity_{\text{positive}} = a/(a+b+c+d)$$
$$Polarity_{\text{negative}} = d/(a+b+c+d)$$

		www.booking.com	
		positive	negative
Agoda	positive	a	b
	negative	c	d

		www.booking.com	
		positive	negative
Agoda	positive	61	2
	negative	6	25

Fig. 3. The matrix of sentiment counting

Finally, each concept and its similar words will be represented in a XML format. An example is shown as Fig. 4.

```
<concept_no> 0001
  <concept_word> good
      <concept_syn> best, ok, ... </concept_syn>
      <concept_polarity_pos> 0.64893617 </ concept_polarity_pos>
      <concept_polarity_neg> 0.26595744 </ concept_polarity_neg>
  </concept_word>
</concept_no>
```

Fig. 4. An example of sentiment word with polarity score

4 The Validation of the Proposed Concept

The proposed concept is validated through sentiment classification based on the employing of the SVM algorithm [18]. The SVM is chose in this work because this algorithm is efficient although using a small features.

Before sentiment classifier is built, the training set must be tokenized through the use of the specific sentiment word corpus. After tokenizing process, all words (concepts) represented in a structured *bag of words* (*BOW*). We obtain $w = (w_1, w_2, \ldots, w_k, \ldots, w_v)$, where v is the number of unique concepts within the collection. By the use of the corpus in the stage of tokenization, the size of BOW is quite small (Fig. 5).

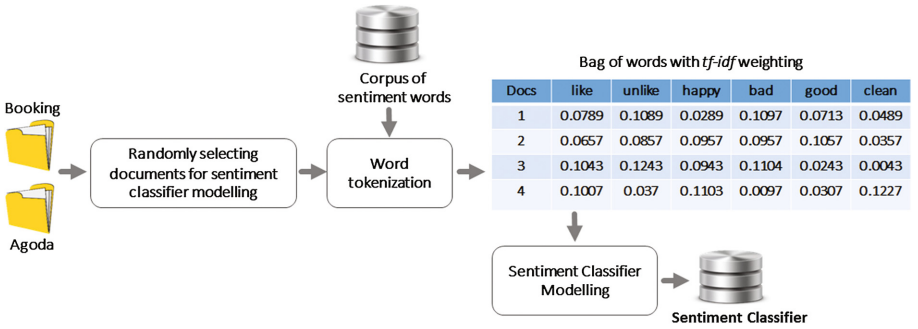


Fig. 5. The process for validating of the proposed concept

In the BOW, a hotel review document d_i is composed of a sequence of concepts, with $d_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,k}, \ldots, w_{i,v})$, where $w_{i,k}$ is the frequency of the k-th concept in the hotel reviews document d_i. Also, each concept is weighted by *tf-idf* [19]. It is used to provide a pre-defined set of features for exchanging information.

Afterwards, it is passed to the process of sentiment classifier utilizing the SVM algorithm with binary SVM classifier. The basic concept behind the training procedure is to find a hyperplane, represented by vector w, that not only separates the document

vectors in one class from those in the other, but for which the separation, or margin, is as large as possible. This search corresponds to a constrained optimization problem; letting $c_j \in \{1, -1\}$ (corresponding to *positive* and *negative*) be the correct class of document d_j, the solution can be written as

$$Minimize: \quad v(w, \xi, \rho) = \frac{||w||^2}{2} + \frac{1}{vl}\sum_{i=1}^{l} \xi_i - \upsilon \tag{1}$$

$$Subject\ to: \quad (w \cdot \Phi(x_i)) \geq \rho - \xi_i, \xi_i \geq 0 \tag{2}$$

where $\nu \in (0, 1)$ is a parameter which lets one control the number of support vectors and errors, ξ is a measure of the mis-categorization errors, and ρ is the margin. When we solve the problem, we can obtain w and ρ. Given a new data points x to classify, a label is assigned according to the decision function that can be expressed as follows:

$$f(x) = sign((w \cdot \Phi(x_i) - \rho) \tag{3}$$

where α_i are Lagrange multipliers and we apply the *Kuhn Tucker* condition [19]. We can set the derivatives with respect to the primal variables equal zero, and then we can get:

$$W = \Sigma\alpha_i \cdot \Phi(x_i) \tag{4}$$

There is only a subset of points x_i that lies closest to the hyperplane and has nonzero values α_i. These points are called support vectors. Instead of solving the primal optimization problem directly, the dual optimization problem is given by:

$$Minimize: \quad W(\alpha) = \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \tag{5}$$

$$Subject\ to: \quad 0 \leq \alpha_i \leq \frac{1}{vl}, \sum_i \alpha_i = 1 \tag{6}$$

where $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j))$ are the kernels functions performing the non-linear mapping into feature space based on dot products between mapped pairs of input points. In this work, we employ *LIBSVM* tools from National Taiwan University [20] to develop our sentiment classifier, and we select the RBF kernels for model building.

5 The Experiment

To validate the effectiveness of our concept, we have performed a preliminary exploration of the binary sentiment classification (positive and negative). We randomly select 3,000 positive hotel reviews and 3,000 negative hotel reviews for testing data. It is different from the training data that is used for generate the corpus of specific sentiment words.

Also, we experiment by comparing between two concepts: traditional BOW and the proposed BOW. In term of traditional BOW, it is the BOW that is obtained by extracting unique words, and then stop-words and a word that occurs only one are removed. For the proposed BOW, it is the BOW that is obtained by using the corpus of specific sentiment words that are generated by Word2Vec. Meanwhile, the proposed BOW does not need the process of stop-word removal. The results are evaluated by using the information retrieval standard [21]. Common performance measures for system evaluation are *precision* (*P*), *recall* (*R*), and *F-measure* (*F*). The experimental results can be presented as Table 1.

Table 1. The experimental results

Techniques	Size of words	Recall	Precision	F-measure
Traditional BOW	380	0.72	0.76	0.739
Proposed BOW	60	0.76	0.79	0.775

Through the experiment shown as Table 1, it can be seen that the results of the proposed BOW are better than the results of the traditional BOW. This would demonstrate that our proposed concept can achieve substantial improvement for sentiment classification.

6 Conclusion

In sentiment classification, most of the existing researches concentrated on the extraction of lexical features and syntactic features, but the sentiment of text is always disregarded. This work introduces a solution similar to word expansion task or word similarity task, which can discover sentiment specific words embedded in texts. We present a method based on Word2Vec to find words having similar meaning, and then these words will be grouped as a same concept (or class) with sentiment polarity. Then, polarity of each concept is estimated by the Cohen's Kappa Statistics. Finally, this concept is validated through sentiment classification based on the employing of the SVM algorithm. Through the experiment, we validate the effectiveness of our concept, by which we have performed a preliminary exploration of the sentiment classification of hotel reviews.

Acknowledgements. This work is supported by Faculty of Informatics, Mahasarakham University.

References

1. Polpinij, J., Ghose, A.: An ontology-based sentiment classification methodology for online consumer reviews. In: IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp. 518–524 (2008)

2. Jiang, L., Yu, M., Zhou, M., Liu, X., Zhao, T.: Target-dependent Twitter sentiment classification. In: Proceeding of Annual Meeting of the Association for Computational Linguistics, pp. 151–160 (2011)
3. Hu, X., Tang, J., Gao, H., Liu, H.: Unsupervised sentiment analysis with emotional signals. In: Proceedings of the International World Wide Web Conference, pp. 607–618 (2013)
4. Polpinij, J., Srikanjanapert, N., Wongsin, C.: Concept-based sentiment analysis for opinion texts with multiple-languages. In: Proceedings of the 12th International Conference on Computing and Information Technology, pp. 27–36 (2015)
5. Mohammad, T.M., Kiritchenko, S., Zhu, X.: NRC-Canada: building the state-of-the-art in sentiment analysis of tweets. In: Proceedings of International Workshop on Semantic Evaluation (2013)
6. Kiritchenko, S., Zhu, X., Mohammad, S.M.: Sentiment analysis of short informal texts. J. Artif. Intell. Res. **50**(1), 723–762 (2014)
7. Matsumoto, S., Takamura, H., Okumura, M.: Sentiment classification using word sub-sequences and dependency sub-trees. In: The Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 301–311 (2005)
8. Tang, D., Wei, F., Yang, N., Zhou, N., Liu, T., Qin, B.: Learning sentiment-specific word embedding for twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 1555–1565 (2014)
9. Vo, D.T., Zhang, Y.: Target-dependent twitter sentiment classification with rich automatic features. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI), pp 1347–1353 (2015)
10. Zhang, D., Xu, H., Su, Z., Xu, Y.: Chinese comments sentiment classification based on Word2Vec and SVM. Expert Syst. Appl. **42**(4), 1857–1863 (2015)
11. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 79–86 (2002)
12. Zhang, W., Xu, W., Chen, G., Guo, J.: A feature extraction method based on word embedding for word similarity computing. Nat. Lang. Process. Chin. Comput. **496**, 160–167 (2014)
13. Mikolov, T., Yih, W.T., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceeding of HLT-NAACL 2013, pp. 746–751 (2013)
14. 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 (2013)
15. Smeeton, N.C.: Early history of the kappa statistic. Biometrics **41**, 795 (1985)
16. Cohen, J.: A coefficient of agreement for nominal scales. Educ. Psychol. Meas. **20**(1), 37–46 (1960)
17. Gutmann, M.U., Hyävrinen, A.: Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. J. Mach. Learn. Res. **13**, 307–361 (2012)
18. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of the International Conference on Machine Learning (ICML) (1999)
19. Yang, Y., Pederson, J.O.: A comparative study on features selection in text categorization. In: Proceedings of the 14th International Conference on Machine Learning (ICML), pp 412–420 (1997)
20. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan (2004)
21. Baeza-Yates, R., Ribeiro-Neto, B.: Modern information retrieval. ACM Press, New York (1999)

Improving Aspect Extraction Using Aspect Frequency and Semantic Similarity-Based Approach for Aspect-Based Sentiment Analysis

Toqir A. Rana(✉) and Yu-N Cheah

School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia
toqirr@gmail.com, yncheah@usm.my

Abstract. Identifying the targets of users' opinions, referred as aspects, in aspect-based sentiment analysis, is the most important and crucial task. A large number of approaches have been proposed to accomplish this task. These approaches identify a huge amount of potential aspects from customer reviews. But not all the extracted aspects are interesting and include terms which are not related to the product and these irrelevant terms affect the performance of the aspect extraction approaches. Therefore, in this paper, we are proposing a two-level aspect pruning approach to eliminate irrelevant aspects. The proposed approach performs the task of aspect pruning in two steps: (a) by calculating the frequency of each word and selecting the most frequent aspects; and (b) by calculating the semantic similarity of non-frequent words and eliminate aspects which are not semantically related to the product. Our experimental evaluation has shown a significant improvement of the proposed approach over the compared approaches.

Keywords: Aspect-based sentiment analysis · Opinion mining · Aspect pruning · Explicit aspects

1 Introduction

In the modern era of information technology, people rely on online shopping services or retailers to buy daily used commodities and routine-life products. These online websites also provide different portals to the customers to leave their experiences or opinions, in the form of reviews, about different products which they have utilized. These online reviews played a very important role for both manufacturers and new customers of the product. But it is almost impossible to read these online reviews manually and conclude a decision due to the enormous amount of the reviews. Hence, there is a need of such system which can automatically analyze online reviews and generate an overall summary. Sentiment analysis is the area of research which determines the overall contextual polarity of the users' opinions towards the product.

Among different granularity levels of sentiment analysis, aspect-based sentiment analysis has attracted a large number of researchers during the last decade [1, 2]. Aspect-based sentiment analysis deals with the extraction of users' opinions and their targets. These opinion targets are usually referred as the features of the product or the

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_30

product itself and collectively called, for the simplicity, as aspect. Identification and extraction of these aspects is the most important and crucial task of the aspect-based sentiment analysis [3].

Usually, there are two types of aspects in online reviews, i.e. explicit and implicit [4]. Explicit aspects are such aspects where the users have used some explicit terms in the review to express their opinions. For example, in the sentence: "This phone is great", this sentence holds a users' opinion "great" which is associated with an explicit term "phone" and hence called as explicit aspect. On the other hand, implicit aspects are such aspects which are not expressed by any explicit term in the review. For example, consider the sentence: "The phone is small", this sentence holds an opinion "small" which represents the "size" of the "phone" but there is no explicit term used for this aspect. Explicit aspects are studied by most of the researchers while there are very few efforts for the implicit aspect extraction, due to the complexity of the implicit aspect identification [3].

Variety of approaches have been proposed for explicit aspect extraction including frequency-based, dependency parser-based, lexicon-based, machine learning approaches and topic modeling [3, 5]. These approaches extract a large number of potential aspects but not all the aspects are related to the product, which leads towards the low precision of the system. Therefore, researchers have used different pruning techniques to detect aspects which are not related to the product [6–10].

In this paper, we have proposed a two-level approach which includes aspect frequency and semantic similarity to detect irrelevant aspects. The aspect frequency-based model is quite similar to redundancy and compactness pruning [7]. Compactness pruning deals with the aspects with multi-words i.e. phrases and redundancy pruning deals with the aspects with the single-word. The compactness pruning eliminates aspects which are not compact i.e. the words in the phrase are not directly associated and this is because of the aspect generation using association rule miner CBA [11]. Our model does not require this method as we have adopted sequential pattern-based model [12] for the aspect extraction, which extracts noun phrases where noun words are associated to each other. Also, in our model, the threshold varies with to the number of sentences and semantic similarity matrix is integrated with the frequency-based model to improve the overall precision of the system. The experimental evaluation clearly indicates the significance of the proposed model.

The remaining of the paper is organized as follows: Sect. 2 outlines the related work. In Sect. 3, proposed methodology for the aspect pruning is being discussed. In Sect. 4, the experimental results are presented followed by the conclusions discussed in Sect. 5.

2 Related Work

Hu and Liu [4, 7] proposed frequency-based approach for both the aspect extraction and pruning. Their proposed approach mined all nouns from the sentence and the association rule-based CBA was applied to generate a set of all frequent itemsets with support higher than the given threshold. Once the frequent itemsets were generated, they used two pruning methods to eliminate irrelevant aspects i.e. compactness and

redundancy pruning. The compactness pruning was necessary because CBA generated all possible itemsets on the basis of extracted nouns in a single sentence without considering the position of the words. Redundancy pruning was applied to eliminate infrequent noun words. Popescu and Etzioni [10] further improved the precision of the above approach by calculating PMI for each aspect and eliminated such aspects which did not match with the input PMI score. Bafna and Toshniwal [6] applied probabilistic approach for the aspect pruning.

Eirinaki et al. [13] used an aspect ranking approach which assigned a score to each extracted noun on the basis of the noun occurrence and the associated opinion. Further, they used these scores to select potential aspects i.e. aspects with the score higher than the given threshold. Bagheri et al. [14] introduced two pruning methods using subset and superset-support. Du et al. [15] purposed support vector machine (SVM) and word alignment-based model to extract aspects. For the aspect pruning, they calculated the confidence of the aspects and eliminated all aspects where the confidence was lower than the given threshold. Hai et al. [16] calculated the Intrinsic (IDR) and Extrinsic-domain relevance (EDR) for each aspect and selected only those aspects which have IDR greater than and EDR lesser than the given threshold. Yu et al. [17] identified important aspects by ranking each aspect using the probabilistic regression algorithm. Ma et al. [18] combined LDA with the synonym lexicon for the aspect extraction.

Liu et al. [19] adopted the graph-based model which select only those patterns, within the graph, which had the higher confidence level. Xu et al. [20] refined the list of extracted aspect using semi-supervised model TSVM. Rana and Cheah [21] proposed a sequential pattern-based model to identify objective aspects. Kang and Zhou [8] proposed a two-step pruning approach i.e. non-frequent identification and semantic similarity. Similarly, Cruz et al. [22] used manually built aspect taxonomy and selected only those aspects which matched with the input list of aspects. Liu et al. [9] applied word vector model and association rules to improve the aspect extraction accuracy. They trained their model using a large corpus of available online reviews.

Dependency parser-based approaches have been applied widely from the last couple of years [23–28]. These approaches extracted aspects on the basis of the word dependencies produced by the dependency parsers. If no dependency was identified among aspect and opinion then the aspect will not be selected.

Our proposed model differs from all the above approaches as it deals with the single-word and multi-word aspects separately and defines the threshold on the basis of the size of the dataset. The threshold varies with the number of sentences i.e. the threshold will be higher for the dataset with the large number of sentences. Along with the frequency pruning, we have applied normalized Google distance (NGD) [29] as similarity measure which does not requires any corpus or trained model as other similarity measure tools suffer with.

3 Proposed Methodology

The proposed methodology for aspect pruning is being carried out in two steps: (1) frequency-based; and (2) semantic similarity-based. Figure 1 elaborates the overall hierarchy of the proposed approach.

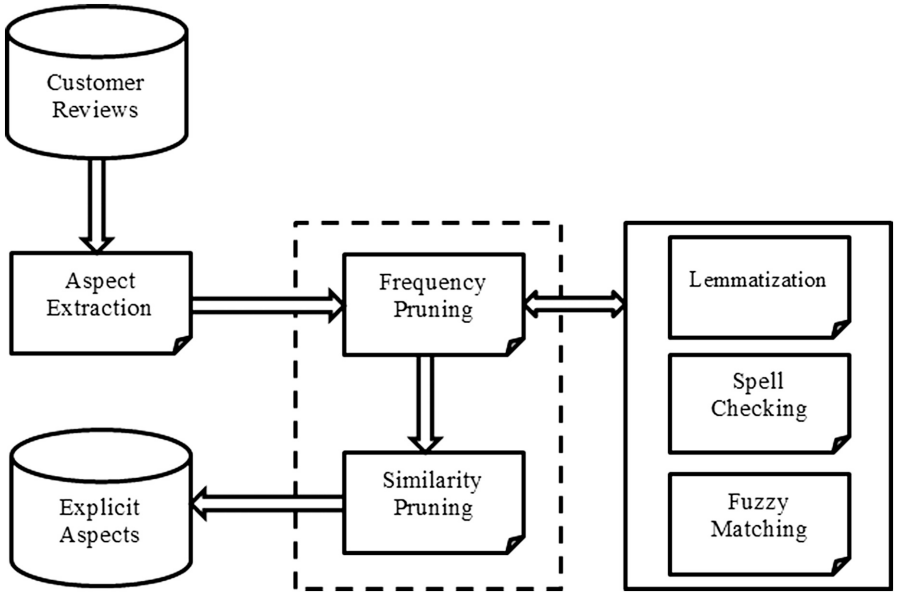


Fig. 1. A two-level aspect pruning model

We have adopted the sequential pattern-based approach for aspect extraction as proposed by Rana and Cheah [12, 30] and hence, we are not going into the details of this process. The core objective of the proposed approach is to improve the performance of the aspect extraction process. Therefore, we have proposed two-level pruning model for this task. This approach performs frequency and similarity pruning on aspects generated during the aspect extraction phase and produces a refined explicit aspects list. We have used natural language tool kit (NLTK[1]) to perform spell checking, fuzzy matching and lemmatization during the frequency pruning process.

3.1 Frequency Pruning

Frequency pruning step eliminates all such nouns/noun phrases, extracted during the aspect extraction phase, which have the frequency lower than the given threshold. First the frequency of all noun words, both in nouns/noun phrases, is being calculated on the

[1] http://www.nltk.org/.

basis of their occurrences in the review sentences. If any word is appeared in one sentence then the frequency of such word is one and it does not matter that how many times the same word appeared in the sentence. More than one occurrence of the same word is considered as one. Similarly, if any word appears in the ten sentences then the frequency of that word is considered as ten. In simple means, more the word appears in the review sentences higher will be the frequency. Spell checking is applied to deal with misspelled words, fuzzy matching is applied to deal with the words like "auto-focus" and "autofocus" and lemmatization is applied for words like "phone" and "phones".

After calculating the frequency of each word, next step is to eliminate all words with the frequency lower than the given threshold. To perform this elimination, we have defined discrete threshold for the nouns and noun phrases. For the noun phrases, the minimum support is set to 2 and the interval is set to 1000 sentences i.e. after each interval the value of the minimum support will be increment by one. It means that, if any noun phrase contains at least one word, with support higher than or equal to minimum support, then the noun phrase is considered as potential aspect. But, if no word, in the noun phrase, has the frequency higher than or equal to minimum threshold then the noun phrase will be eliminated.

For the nouns with single word, the interval and the rate of change in the minimum support are different. The minimum support is set to 2, as in the case of noun phrases, but the interval is set to 500 sentences. Also, after each interval the minimum support will be incremented by 2, i.e. for 1000 sentences the minimum threshold will be 4 and vice versa. Hence, all the nouns, which have the support less than the given threshold, will be eliminated.

3.2 Similarity Pruning

During the frequency pruning step, all such nouns/noun phrases were eliminated which did not meet the minimum threshold. In the case of noun phrases, this seems quite logical as if no word is frequent then the phrase will be eliminated. But in the case of nouns, which consist of only one word, all such potential aspects will be pruned out which were related to the product but not addressed by the large number of users. Therefore, we have applied the semantic similarity measure to detect those nouns which were eliminated in the frequency pruning phase but are relevant to the product.

We have applied NGD [29] to measure the similarity between the two terms. Although, WordNet [31] and Word2Vec[2] similarity measures were also applied for improving the aspect extraction accuracy, but these both tools required a huge corpus and a trained model to evaluate two terms. On the other hand, NGD does not suffer from these limitations and only requires the number of hits returned by the search engine. NGD calculates the similarity among two terms on the basis of co-occurrence of these terms on the web. If both terms are always appeared on the same web page

[2] https://code.google.com/p/word2vec/.

then the NGD score will be 0 and if both terms never appeared on the same web page then the score will be infinite. The following is the formula to calculate NGD.

$$\text{NGD}\ (x, y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log N - \min\{\log f(x), \log f(y)\}} \quad (1)$$

In the formula, *x* is the first term and *y* is the second term and *f()* represents the number of hits return by the search engine. *N* is the total number of web pages over the internet and estimated on the basis of the number of hits against "the" keyword.

For all the nouns, which were eliminated during the frequency pruning phase, we have calculated the NGD of the noun and the product. If the NGD is lower than the given threshold, in our case this is 0.1, then whether that noun is frequent or not, will be considered as potential aspect.

4 Experimental Evaluation

This section elaborates the experimental evaluation to assess the significance of the proposed approach. We have used customer review dataset[3] which contains five different electronic products. Table 1 shows the detailed information of all the products.

Table 1. Detailed description of the test datasets.

Data	Product	Total # of sentences	# of opinionated sentences	# of non-opinionated sentences
D1	Canon digital camera	597	40%	60%
D2	Nikon digital camera	346	46%	54%
D3	Nokia cell phone	546	49%	51%
D4	Creative MP3 player	1716	42%	58%
D5	Apex DVD player	740	47%	53%

These datasets were annotated by Hu and Liu [7] i.e. every opinionated sentence is tagged with all the aspects, which are the targets of users' opinions in that sentence. Sentences without any tagged aspect are the non-opinionated sentences i.e. the sentence does not hold any users' opinion and associated aspect.

As proposed approach focuses only on the aspect pruning, therefore, we have used precision as the evaluation matrix. The precision is being calculated using true positive (TP) and false positive (FP) ratio. Consider the set *A* as the set of extracted aspects, *T* as

[3] https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html.

the set of manually annotated aspects from the datasets. With this assumption, TP will be $|A \cap T|$ and FP will be $|A \backslash T|$. With the help of TP and FP, following is the formula to calculate precision.

$$P = \frac{TP}{TP + FP} \tag{2}$$

For the experimental evaluation, we have compared our results with the three different approaches i.e. rule-based extraction (RubE) [8], aspect extraction based on recommendation (AER) [9] and association rules-based system (ABS) [7]. RubE used WordNet and aspect frequency, AER applied Word2Vec along with association rules and ABS defined the compactness and redundancy pruning to improve opinion target extraction. As compared to these approaches, proposed model also uses the aspect frequency in the first phase. But unlike other approaches, the threshold is dynamic and changes with the number of sentences i.e. larger the dataset greater the threshold. This seems quite logical as the threshold required for small datasets is not applicable on large datasets. Also, NGD does not require any trained model or huge corpus as compared to the other similarity matrices.

Table 2 shows the results of proposed approach and compared approaches over the customer reviews datasets. For the selection of threshold values, different iterations have been performed on each product dataset with different threshold levels. We have selected those threshold levels which performed better on every product dataset. For the proposed approach, the D1 dataset produced the lowest accuracy score as compared to the other datasets. Because, the D1 dataset contains large number of non-opinionated sentences i.e. 60% sentences do not hold any users' opinion and product aspect, as shown in Table 1. Also, D1 holds a large number of sentences where users have discussed other products rather than expressing their views on Canon camera.

Table 2 clearly elaborates the significance of the proposed approach over the compared approaches. Our proposed approach shows 10% improvement over the state-of-the-art ABS which used the frequency-based pruning methods. As compared to the AER, which applied semantic similarity using Word2Vec and association rules, our approach improves by 5% and shows 1% improvement as compared to RubE which used WordNet for the similarity measure. Both Word2Vec and WordNet require a huge corpus and trained model to calculate the similarity between the two terms while our

Table 2. Accuracy comparison of proposed approach with ABS, AER and RubE

Data	ABS	AER	RubE	FB+NGD
D1	0.83	0.81	0.87	0.82
D2	0.78	0.83	0.90	0.88
D3	0.83	0.87	0.90	0.93
D4	0.75	0.82	0.87	0.88
D5	0.77	0.88	0.90	0.92
Avg	0.79	0.84	0.87	0.89

proposed approach uses the NGD which only requires the number of hits returned by the search engine and hence, does not suffer with the corpus or trained model constraints.

5 Conclusion

Aspect extraction is the key task of aspect-based sentiment analysis and requires the precise identification of the users' opinions and their targets. Many approaches have been proposed to accomplish the above task but all suffer from the irrelevant aspects. Therefore, aspect pruning is inevitable to enhance the performance of aspect extraction techniques. This paper proposed a two-level pruning model to eliminated irrelevant aspects. The first step detects aspects which are not frequent and eliminates all aspects with the frequency lower than the given threshold. The second step performs semantic similarity measure to re-select aspects which are semantically related to the product. We have used the NGD for semantic similarity because NGD does not depends on any corpus or trained model as compared to the other similarity measure tools. The experimental evaluation proves the significance of the proposed methodology over the related approaches for the aspect pruning.

Acknowledgements. Toqir A. Rana would like to gratefully acknowledge the Ministry of Higher Education (MOHE), Malaysia, for supporting his studies under the Malaysian International Scholarship (MIS) program.

References

1. Liu, B.: Sentiment analysis and opinion mining. Synth. Lect. Hum. Lang. Technol. **5**, 1–167 (2012)
2. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retr. **2**, 1–135 (2008)
3. Rana, T.A., Cheah, Y.-N.: Aspect extraction in sentiment analysis: comparative analysis and aurvey. Artif. Intell. Rev. **46**, 459–483 (2016)
4. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: 10th ACM SIGKDD International Conference on Knowledge discovery and Data Mining, pp. 168–177. ACM (2004)
5. Rana, T.A., Cheah, Y.-N., Letchmunan, S.: Topic modeling in sentiment analysis: a systematic review. J. ICT Res. Appl. **10**, 76–93 (2016)
6. Bafna, K., Toshniwal, D.: Feature based summarization of customers' reviews of online products. Proc. Comput. Sci. **22**, 142–151 (2013)
7. Hu, M., Liu, B.: Mining opinion features in customer reviews. In: 19th National Conference on Artificial intelligence, pp. 755–760. San Jose (2004)
8. Kang, Y., Zhou, L.: RubE: rule-based methods for extracting product features from online consumer reviews. Inf. Manag. **54**, 166–176 (2016)
9. Liu, Q., Liu, B., Zhang, Y., Kim, D.S., Gao, Z.: Improving opinion aspect extraction using semantic similarity and aspect associations. In: 13th AAAI Conference on Artificial Intelligence (AAAI), Phoenix (2016)

10. Popescu, A.-M., Etzioni, O.: Extracting Product Features and Opinions from Reviews. Natural Language Processing and Text Mining, pp. 9–28. Springer, New York (2007)
11. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: 4th International Conference on Knowledge Discovery and Data Mining (KDD) (1998)
12. Rana, T.A., Cheah, Y.-N.: Sequential patterns-based rules for aspect-based sentiment analysis. In: 3rd International Conference on Computational Science and Technology (ICCST) (2016)
13. Eirinaki, M., Pisal, S., Singh, J.: Feature-based opinion mining and ranking. J. Comput. Syst. Sci. **78**, 1175–1184 (2012)
14. Bagheri, A., Saraee, M., de Jong, F.: An Unsupervised Aspect Detection Model for Sentiment Analysis of Reviews. Natural Language Processing and Information Systems, pp. 140–151. Springer (2013)
15. Du, J., Chan, W., Zhou, X.: A Product aspects identification method by using translation-based language model. In: 22nd International Conference on Pattern Recognition (ICPR), pp. 2790–2795. IEEE (2014)
16. Hai, Z., Chang, K., Kim, J.-J., Yang, C.C.: Identifying features in opinion mining via intrinsic and extrinsic domain relevance. IEEE Trans. Knowl. Data Eng. **26**, 623–634 (2014)
17. Yu, J., Zha, Z.-J., Wang, M., Chua, T.-S.: Aspect ranking: identifying important product aspects from online consumer reviews. In: 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 1496–1505. ACL (2011)
18. Ma, B., Zhang, D., Yan, Z., Kim, T.: An LDA and synonym lexicon based approach to product feature extraction from online consumer product reviews. J. Electron. Commerce Res. **14**, 304–314 (2013)
19. Liu, K., Xu, L., Zhao, J.: Opinion target extraction using word-based translation model. In: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 1346–1356. ACL (2012)
20. Xu, L., Liu, K., Lai, S., Chen, Y., Zhao, J.: Mining opinion words and opinion targets in a two-stage framework. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pp. 1764–1773 (2013)
21. Rana, T.A., Cheah, Y.-N.: Exploiting sequential patterns to detect objective aspects from online reviews. In: 3rd International Conference on Advanced Informatics: Concepts, Theory and Application (ICAICTA), pp. 1–5. IEEE (2016)
22. Cruz, F.L., Troyano, J.A., Enríquez, F., Ortega, F.J., Vallejo, C.G.: Long autonomy or long delay? The importance of domain in opinion mining. Expert Syst. Appl. **40**, 3174–3184 (2013)
23. Liu, Q., Gao, Z., Liu, B., Zhang, Y.: Automated rule selection for opinion target extraction. Knowl. Based Syst. **104**, 74–88 (2016)
24. Poria, S., Cambria, E., Ku, L.-W., Gui, C., Gelbukh, A.: A rule-based approach to aspect extraction from product reviews. In: 2nd Workshop on Natural Language Processing for Social Media (SocialNLP), pp. 28–37. 28 (2014)
25. Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. Comput. Linguist. **37**, 9–27 (2011)
26. Wu, Y., Zhang, Q., Huang, X., Wu, L.: Phrase dependency parsing for opinion mining. In: Conference on Empirical Methods in Natural Language Processing, pp. 1533–1541. ACL (2009)
27. Yu, J., Zha, Z.-J., Wang, M., Wang, K., Chua, T.-S.: Domain-assisted product aspect hierarchy generation: towards hierarchical organization of unstructured consumer reviews. In: Conference on Empirical Methods in Natural Language Processing, pp. 140–150. ACL (2011)

28. Zhang, L., Liu, B., Lim, S.H., O'Brien-Strain, E.: Extracting and ranking product features in opinion documents. In: 23rd International Conference on Computational Linguistics, Posters, pp. 1462–1470. ACL (2010)
29. Cilibrasi, R.L., Vitanyi, P.M.: The Google similarity distance. IEEE Trans. Knowl. Data Eng. **19**, 370–383 (2007)
30. Rana, T.A., Cheah, Y.-N.: Hybrid rule-based approach for aspect extraction and categorization from customer reviews. In: 9th International Conference on IT in Asia (CITA), pp. 1–5. IEEE (2015)
31. Miller, G., Fellbaum, C.: Wordnet: An Electronic Lexical Database. MIT Press, Boston (1998)

Recognize the Same Users across Multiple Online Social Networks

Siqi Li, Wenxin Liang(✉), and Xianchao Zhang

School of Software, Dalian University of Technology, Dalian 116620, China
lisiqitony@mail.dlut.edu.cn, {wxliang,xczhang}@dlut.edu.cn

Abstract. Nowadays, online social networks (OSNs) play an important role in our daily lives. And it is very common for a person to have many profiles in different OSNs. However, different profiles in different OSNs of the same person are isolated from each other. User Identity Resolution (UIR) is the problem to recognize the same person in different OSNs. Most methods are mainly concerned with the profile attributes and they just use the information of profiles. In this paper, we propose a new algorithm, called Identity Matching based on Propagation of anchor links (IMP) which fully combines the profile attributes, the linkage information and the social actions, and solves the problem by expanding the anchor links (seed account pairs that belongs to the same user). In the IMP algorithm, we use the information of the nodes surrounding the anchor nodes and identify new links. As the spread of the anchor nodes, we can iteratively find more and more links. We conduct extensive experiments on Twitter and Facebook to evaluate our algorithm and the results show that our algorithm significantly improves the matching results and outperforms the baseline algorithms.

Keywords: Social networks · User identity matching · Anchor links

1 Introduction

Social Network is also called Social Network Service (SNS), and its original intention is to help people connect and interact through the Internet and form a social behaviour which is similar to the real world. With the development of the social action, many OSNs with abundant functions appear, users can publish contents and follow stars or web celebrities in Twitter and Micro Blog, can share remarks on different sites in Foursquare and can share multimedia contents and interact with friends in Facebook. Meanwhile, facing so many SNSs, people need to register as members of different OSNs to enjoy various services.

Unfortunately, the OSNs are isolated with each other and they do not share a profile. Also there is not a recognized indicator to identify users, so accounts of different OSNs that belong to a person do not have any connections with each other. In fact, these accounts have much information in common, but we have

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_31

to write the repeated profile information again and again. Because users take part in different OSNs for different purposes, they may have different behaviours in different OSNs. So the biggest obstacle of utilizing the data of OSNs is the fact that accounts and actions scatter in each OSN, which makes it difficult to capture the complete social graph.

The UIR problem is identifying the same users across different OSNs. Once the problem is solved, it has a great significance for providing more detailed and personalized services to the users. For instance, knowing a user's Twitter account, we can get more information such as his location and social circles, which is useful to support a more targeted and personalized recommendation in Foursquare.

However, it is not easy to tackle the UIR problem across different OSNs, because the features are sparse and always missing. Most existing methods mainly focus on the attribute matching. But now many OSNs protect the user's privacy and also someone is unwilling to open his profile to public. Additionally, different OSNs may have different data formats. Some attributes may exist in one OSN, but do not exist in another OSN. Besides, the format of the friend relationship may differ. For example, there is a directed connection between friends in Twitter while a mutual connection in Facebook. So the incomplete information makes the UIR problem very difficult.

In this paper, we use a propagation method of the anchor links. Anchor links are seed account pairs that belong to the same user, we also use it to represent the identified links in the matching process. We consider the anchor links have much information and a good friend relationship in one OSN may also exists in another OSN, so the nodes surrounding the anchor links are easy to be identified. At each iteration, we start from the anchor links, and detect the nodes next to them. By comparing the profile similarity and the social similarity (the closeness to the anchor links) between two nodes from different OSNs using the logistic regression model, we can identify some pairs and put them into anchor links set.

Our contributions are summarized as follows.

- In this paper, we propose Social Interaction Score (SIS) and Social Graph Score (SGS) to represent the closeness between two users from an intra network in the view of social action and social graph.
- Based on the profile similarity and social similarity, we put up with an Identity Matching based on Propagation of anchor links (IMP) algorithm, which fully uses anchor link information and iteratively find matching pairs next to anchor nodes.
- Experiments on two real OSNs show that our algorithm improves the matching results and outperforms the baseline algorithms.

The rest of this paper is organized as follows. Section 2 provides a brief overview of the related work. Section 3 formalizes the problem. Section 4 introduces the proposed methods and Sect. 5 describes the experimental study. Section 6 concludes this paper.

2 Related Work

Due to the importance of the UIR in many fields, deep and complete researches have been done on UIR in various aspects. We review some key technologies in this section.

Most UIR methods are focusing on profile attributes. Vosecky et al. [1] presented the VMN algorithm to measure the similarity of name. In their work, it used a vector representing each user profile, then it computed the similarity of each dimension. When computing the similarity of each dimension, it used the exact matching, partial matching and fuzzy matching for different features respectively. Malhotra et al. [3] used some automated classifiers and found that ID and Name of a user are the key features for matching. Raad et al. [11] came up with a FOAF method for attribute matching. It can solve the problem that attribute formats differ in different OSNs. This method transform all the attribute information into FOAF format. The most prominent contribution is using different similarities computing methods for different attributes. Zafarani et al. [2] proposed a methodology (MOBIUS) to match identities of individuals across OSNs. It extracted many features from the usernames containing the length, writing pattern and the prefix to exploit information redundancies.

However, only using attributes to match has some disadvantages, so some researches use the friend relationship and contents to do the further study. Bartunov et al. [4] proposed the JLA algorithm to solve the UIR problem from a local perspective in an ego-network. In JLA, Conditional Random Fields is used and both of profile attributes and friends linkages are considered. And the intuition of the algorithm is that sharing a friend with a few friends is more helpful for locating a user than sharing a friend who enjoys social and has many friends. Cui et al. [10] used graph matching for finding email correspondents in OSNs. Jain et al. [6] assumed that profiles with more mutual friends will have more possibility to be the matching users. And Cortis et al. [5] introduced a profile resolution technique using the profile information from the syntactic view and semantic view.

Also some methods used abundant information such as timestamp and location from tags and posts. MNA method derived by Kong et al. [7] used four features extracted from accounts. And Peled et al. [8] used a variety of features extracted from users' and their friends' profiles to match user profiles across multiple OSNs.

3 Problem Formulation

Given a source network G^s and a target network G^t, and we can use two undirected graphs to represent the two networks. Also we denote P to be the set of the real person. The source network can be molded as $G^s = (V^s, E^s, A^s, L^s, \phi^s)$, where

- $V^s = \{v_0^s, v_1^s, v_2^s, ..., v_n^s\}$ is the set of users;
- $E^s = \{e_0^s, e_1^s, e_2^s, ..., e_m^s\}$ is the set of links representing the relationship between the users;

·A^s is defined on V^s, and for each $v_i^s \in V^s$, $A^s(v_i^s)$ is a set of labels for v_i^s which can describe a user;

·L^s is defined on E^s, and for each $e_j^s \in V^s$, $L^s(e_j^s)$ is a set of labels for e_j^s which can represent the closeness of a relationship;

·ϕ^s is a mapping function $\phi^s : V^s \rightarrow P$ that connects an OSN user to a real person.

Also the target network G^t is similar with the source network G^s. And we also know the anchor links set AL already, which consists of a source anchor set AR^s in G^s and a target anchor set AR^t in G^t, and $AL = \{(ar_i^s, ar_i^n)|(ar_i^s, ar_i^n)$ is an anchor link provided, $ar_i^s \in AR^s, AR^s \in V(G^s), ar_i^t \in AR^t, AR^t \in V(G^t)\}$.

Problem Definition: Given two graphs $G^s = (V^s, E^s, A^s, L^s, \phi^s)$, $G^t = (V^t, E^t, A^t, L^t, \phi^t)$ and the anchor links set AL, the goal of the problem is to find the pairs $\{(v_i^s, v_j^t)|v_i^s \in V^s, v_j^t \in V^t$ and $\phi^s(v_i^s) = \phi^t(v_j^t)\}$ as many as possible.

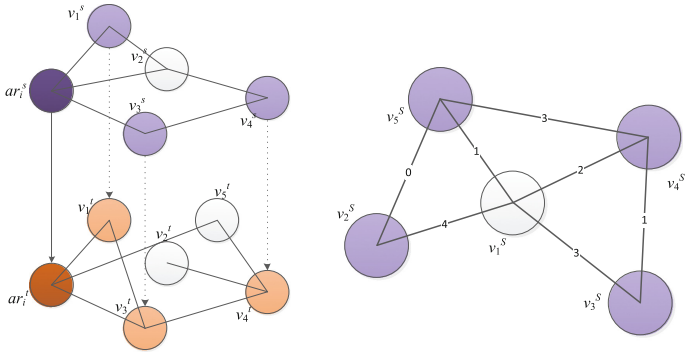
4 Our Algorithm

In this section, we produce a method using greedy strategy and local perspective which can effectively recognize users across OSNs.

4.1 IMP Algorithm

The intuition of the algorithm is that if two persons are friends in real life, they may also have friend relationship in every OSN, because such social links usually indicate the users social ties in real life. Thus the user v_i^s connecting with the anchor links ar_i^s is more recognizable than others, and the matching account v_j^t may also be the friend of ar_i^t. Even if they are not friends, the distance between them can't be too far. Based on the above theory, the algorithm starts from the anchor links, and detects the nodes next to them. By comparing the profile similarity and the social similarity between two nodes from different OSNs using the logistic regression model, we can identify some pairs and put them into generated anchor links set. As the propagation of the anchor nodes, we can match more nodes surrounding the anchor nodes iteratively.

Figure 1(a) shows our idea. (ar_i^s, ar_i^t) is an anchor link that is given. We first visit the neighbors of ar_i^s and select v_1^s as the next matching node. And we calculate the profile similarity and the social similarity between v_1^s and the nodes in G^t (because at the first time in this case, there is only one matching pair that is already known (ar_0^s, ar_0^t), so the social similarity of all the pairs is 1, and it relies on profile attribute to identify). With the logistic regression model, v_1^t is v_1^s's favorite matching node, then doing a reserve matching, and v_1^t's favorite matching node is v_1^s, so (v_1^s, v_1^t) is a matching pair and the pair joins the generated anchor links. The fact that v_1^s's favorite matching node is v_1^t means $Y_{(v_1^s, v_1^t)} = 1$ and $P(Y_{(v_1^s, v_1^t)} = 1|x)$ is the highest among all the pairs containing v_1^s. After identification of (v_1^s, v_1^t), we search new set of neighbors of the anchor links, and recognize the pair (v_3^s, v_3^t). At last the pair (v_4^s, v_4^t) is matched.



(a) An example of IMP algorithm

(b) An example of SIS

Fig. 1. Examples of IMP algorithm and SIS

Algorithm 1 shows the overall algorithm. Every time we search the neighbors of the anchor links and use a label to mark the anchor links set AL. Once the AL updates, we repeatedly seek the new neighbors of the anchor links. Inspired by the Stable Marriage Problem, a reverse matching step is used in our algorithm. And if the matching ends up but there still exist some unvisited nodes that are not neighbors of all the anchor nodes, we will randomly select a node from the unvisited nodes and continue to match.

Reverse Matching: When using the logistic regression to find the matching node of v_i^s, there exists a problem that more than one nodes in G^t matches v_i^s, which is impossible in the real life. We can choose the pair that has the highest score $P(Y_{(v_i^s, v_b^t)} = 1|x)$ and find the node v_b^t, after that, we still need to do a reverse matching. If v_b^t gets mapped back to v_i^s, the mapping is retained; otherwise, it is rejected. This reverse matching can help to improve the accuracy of matching.

4.2 Matching Feature Similarity

The features that we use for matching are divided into two parts:

Profile Similarity: Profile similarity has been extensively researched, and there are many methods to calculate the similarity of the profile information. The profile attributes that we collect only contain Name, Screen Name and URL, and we use the VMN algorithm (an effective method for full and partial matches of names consisting of one or more words) to measure the similarity degree.

Social Similarity: Matching based on profile similarity is an easy but sometimes not effective way, because some users may hide their profile information or write down different profiles on different purposes. However the social features cannot be easily hidden or modified, because online social links usually indicate

Algorithm 1. IMP Algorithm (G^s, G^t, AL)

Input: Source network G^s, Target network G^t, Anchor Links AL
Output: updated Anchor Links AL

```
while (Refresh) do
  for each v_i^s in AR^s.neighbor() do
    v_b^t = the favorite matching node of v_i^s in G^t using the logistic regression model
    v_j^s = the favorite matching node of v_b^t in G^s using the logistic regression model
    if v_i^s == v_j^s then
      AL.insert(v_i^s v_b^t)
      Refresh=true
      break
    else
      Refresh=false
    end if
  end for
end while
```

the users social ties in real life. And we can fully use the social similarity between different OSNs account to help match the same user.

For each edge in the graph, we calculate the score to represent the closeness between two users from an intra network, and the score is from social interaction view and the social graph view.

– *social graph score(SGS).* There are many indexes to compute the similarity of the two users in the view of the graph, and we simply use the Jaccard Coefficient to compute.

$$SGS(v_i^s, v_j^s) = \frac{F(v_i^s) \cap F(v_j^s)}{F(v_i^s) \cup F(v_j^s)} \tag{1}$$

Structural features are aiming at extracting connectivity properties for pairs of objects, it is widely used in link prediction. However, only using the structural feature is not enough, because social graph also has its unique features of social action.

– *social interaction score (SIS).* When using the social network, we can forward other's post in Facebook or retweet other's tweet in Twitter, also we can use the @ function in our own post or tweet to remind a friend. These social actions generate from two users and can reflect the closeness between them. We hold the view that the more social interactions that two users have, the closer the friends are. And the SIS can be calculated as follows:

$$SIS(v_i^s, v_k^s) = 0.5 * \frac{C_{ik}^s}{\sum_{v_j^s \epsilon F(v_i^s)} C_{ij}^s} + 0.5 * \frac{C_{ik}^s}{\sum_{v_j^s \epsilon F(v_i^s)} C_{jk}^s} \tag{2}$$

where C_{ij}^s means the times of the social interactions between v_i^s and v_j^s and $\mathrm{F}(v_i^s)$ represents v_i^s's neighbors;
And Fig. 1(b) shows an example of the SIS. In the figure, the number on the

edge means the times of the social interactions between two users, the final score between v_1^s and v_2^s : $0.5 * 4/(1+2+3+4) + 0.5 * 4/(4+0) = 0.7$. This score is very high and it indicates the relationship between them is very close.

For a node, we can get the vector of scores to the anchor nodes within two matching distance. Similarly, in another network, we can also get the vector of scores to the corresponding anchor nodes. At last, the social similarity can be measured by the cosine similarity of the two vectors.

4.3 Matching Classifier Based on Logistic Regression Model

Logistic regression model is suitable for probabilistic binary classification [9], and its result y can be on only 0 or 1. So a user-identity classifier is built to solve this problem. The input of the model is the features of a pair, and the output is 1 if the two accounts belong to the same person, or 0 otherwise. Logistic regression is based on a hypothesis function $h_\theta(x)$, which is

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \tag{3}$$

Equation 3 is called logistic function. $x = (x_0, x_1, ..., x_n)$ is a n-dimensional vector, which represents different attributes of a user. $\theta = (\theta_0, \theta_1, ..., \theta_n)$ is parameter vector corresponding to x. Given the logistic regression model and training data, we can find the fittest θ for the model. And the cost function $J(\theta)$ is

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} Cost(h_\theta(x^i), y^i) \tag{4}$$

$$Cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & y = 1 \\ -\log(1 - h_\theta(x)) & y = 0 \end{cases} \tag{5}$$

Then using labeled data to train the model, we can solve the problem by finding the optimum parameter vector $\theta = (\theta_0, \theta_1, ..., \theta_n)$ to minimize $J(\theta)$. We can use the Gradient descent method to find the optimum parameter to minimization.

5 Experiments

5.1 Evaluation Metrics

Generally, we use the precision, recall and F1-Measure to evaluate the method.

$$precision = \frac{tp}{tp + fp} \tag{6}$$

$$recall = \frac{tp}{tp + fn} \tag{7}$$

Here, tp is the number of correct account pairs, fp is the number of incorrect matching pairs, and fn represents the number of undiscovered matching pairs.

F1-Measure is defined as follow:

$$F1 - Measure = \frac{2 * precision * recall}{precision + recall} \tag{8}$$

5.2 Dataset

Because no available datasets is open for UIR, we have to create our own datasets. To evaluate our method for UIR, we do the experiment on two real social networks, and we choose Facebook and Twitter because they are popular OSNs in the world, and the information of a user's profile and linkage can be easily found in these two OSNs, which does not involve other's privacy.

The profile attributes we use in the experiment are Name, Screen Name and URL. We can get much information from Facebook, but in Twitter, the information of the profile is just a little, so we can only choose these three kinds. Also, there is a directed connection between friends in Twitter while a mutual connection in Facebook, so we just consider mutual relationship. And the dataset statistics we crawled are displayed in Table 1(a).

Table 1. Dataset and result

(a) Dataset

	Twitter	Facebook
users	1012	1433
links	8502	14021
matches	357	357
tweet/post	15364	37098

(b) Evaluation results

Algorithm	Recall	Precision	F1-score
MOBIUS	0.55	0.83	0.66
JLA	0.43	0.79	0.57
IMP_1	0.57	0.81	0.67
IMP_2	0.58	0.82	0.68
IMP_3	0.55	0.82	0.66
IMP_4	**0.64**	0.71	0.67
IMP	0.61	**0.85**	**0.71**

5.3 Baseline

We use six algorithms to compare with IMP in this work. MOBIUS uses many features from username, and it can be seen as a traditional matching method only using attribute, JLA combines profile and links information, and the others are reduced versions of IMP. Also the dataset does not contain much abundant information so some methods using geo-tag, location, timestamp cannot be used.

1. **MOBIUS:** Its full name is **Modeling Behavior for Identifying Users across Sites** and it extracts many features such as the length, the language pattern and the prefix from usernames to predict the social links. It can be seen as a matching method only using attribute. And because in Twitter, we can only get the name, screen name and URL, so some other complex methods can not have good effects.

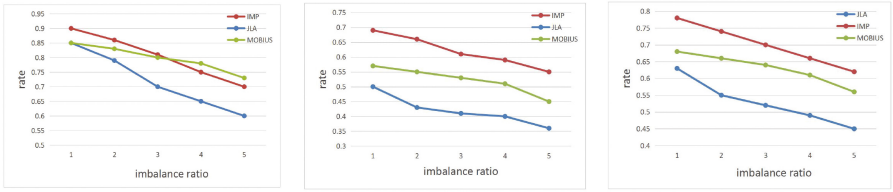
2. **JLA:** Its full name is **Joint Link-Attribute**, so it not only uses profile information but also the linkage information. It is applied on ego-network, which is very small and the distance between two nodes is no longer than two. And the intuition of the algorithm is that sharing a friend with a few friends is more helpful for locating a user than sharing a friend who enjoys social and has many friends.
3. **IMP_1**(IMP without SIS)**:** Only using the SGS and profile similarity of users as the matching features.
4. **IMP_2**(IMP without SGS)**:** Only using the SIS and profile similarity of users as the matching features.
5. **IMP_3**(IMP without social similarity)**:** Only using the profile similarity of users, and this is like the traditional method.
6. **IMP_4**(IMP without reverse matching)**:** Without reverse matching process, the generated results may have a large number of false matching pairs.

5.4 Experimental Evaluation

In our experiments, we use 5-fold cross validation and partition the users: one fold for training and the others for testing. In each round, we sample 10 users in each part of the testing data and treat them as anchor links.

We can see from Table 1(b), JLA does not perform well because JLA is applied on the ego-network, when the network is large, more situations need to be considered and the intuition may be not right. As for MOBIUS, it extracts many features from username and is good at dealing with pairs having similar names, but it does not use the information of the links, when the usernames of the same person across two OSNs differ a lot, it cannot find these pairs. And IMP without social similarity achieves a result similar to the MOBIUS because both of them only use the poor attribute information. Also, IMP without reverse matching achieves the highest recall at the cost of low precision.

In addition, we conduct another experiment to test our method with different imbalance datasets comparing with JLA and MOBIUS. In each round of the cross validation, the datasets is under different imbalance ratios (imbalance ratio = number of negative account pairs/number of positive account pairs). Figure 2 shows the performances of each of the models under different imbalance ratios. We can find our algorithm achieves the best matching results, which indicates our



(a) precision (b) recall (c) F1-measure

Fig. 2. Performance comparison under imbalance ratios

algorithm effectively take advantage of profile attributes and linkage information, and solves the UIR problem.

6 Conclusion

In this paper, we present a greedy and local algorithm to solve the UIR problem. In IMP, we pay attention to the nodes connecting with the anchor nodes, and as the spread of the anchor nodes, we can iteratively find more and more links. Besides, we think a good relationship between two users in one OSN may also exist in another OSN, so we define a social interaction score from online social action to mark a user's favorite friend. Experiments on two real OSNs show that our algorithm effectively uses the profile and linkage information, improves the matching results and outperforms the baseline algorithms.

Acknowledgement. This work was supported by NSFC (No. 61632019) and 863 project of China (No. 2015AA015403).

References

1. Vosecky, J., Hong, D., Shen, V.Y.: User identification across multiple social networks. In: First International Conference on Networked Digital Technologies, NDT 2009, pp. 360–365. IEEE (2009)
2. Zafarani, R., Liu, H.: connecting users across social media sites: a behavioral-modeling approach. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 41–49. ACM (2013)
3. Malhotra, A., Totti, L., Meira Jr., W., Kumaraguru, P., Almeida, V.: Studying user footprints in different online social networks. In: Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2012, pp. 1065–1070. IEEE Computer Society (2012)
4. Bartunov, S., Korshunov, A., Park, S.T., Ryu, W., Lee, H.: Joint link-attribute user identity resolution in online social networks. In: Proceedinghs of the Sixth SNA-KDD Workshop (2012)
5. Cortis, K., Scerri, S., Rivera, I., Handschuh, S.: An ontology-based technique for online profile resolution. Soc. Inf. **8238**, 284–298 (2013)
6. Jain, P., Kumaraguru, P.: Finding Nemo: searching and resolving identities of users across online social networks (2012). arXiv:1212.6147v1
7. Kong, X., Zhang, J., Yu, P.S.: Inferring anchor links across multiple heterogeneous social networks. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge management, pp. 179–188. ACM (2013)
8. Peled, O., Fire, M., Rokach, L., Elovici, Y.: Matching entities across online social networks. Neurocomputing **210**, 91–106 (2016)
9. Zhu, X., Nie, Y., Jin, S., Li, A., Jia, Y.: Spammer Detection on Online Social Networks Based on Logistic Regression, pp. 29–40. Springer, Cham (2015)
10. Cui, Y., Pei, J., Tang, G., Luk, W.S., Jiang, D., Hua, M.: Finding email correspondents in online social networks. World Wide Web **16**(2), 195–218 (2013)
11. Raad, E., Chbeir, R., Dipanda, A.: User profile matching in social networks. In: 13th International Conference on Network-Based Information Systems (NBiS), pp. 297–304. IEEE (2010)

Business Popularity Analysis from Twitter

Pajaree Yaisawas, Sukanlaya Lerdsri, Bundit Thanasopon(✉), and Ponrudee Netisopakul

Information Technology Faculty,
King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
{56070086, 56070140}@kmitl.ac.th,
{bundit, ponrudee}@it.kmitl.ac.th

Abstract. Social media is increasingly utilized for sharing information of online products, from business owners to customers, as well as among customers themselves. In order to utilize these sharing information, this paper proposes and demonstrates the methodology for analyzing business brand popularity based on Twitter posts. The analysis can be visualized by implementing a web application that keeps track and analyzes Twitter posts mentioning about cosmetic and beauty product. Specifically, the application focuses on Twitter posts in Thai; and its key features are, (1) identifying brands being in trend, (2) analyzing and virtualizing statistics provided by Twitter, and (3) classifying Twitter posts' sentiment into positive, negative and objective. The website provides useful insights to brand owners aiming at exploiting social media and to customers buying products from those brands.

Keywords: Twitter · Hashtag · Brand · Popularity · Social listening · Sentiment analysis · Social media analysis

1 Introduction

In recent years, social media has increasingly played an important role in today's economy. Social media allows individual consumer to easily interact and exchange information with each other [1]. One of the most popular social media in Thailand is Twitter. Twitter is a free social networking microblogging service. It allows users to broadcast short posts (140 characters) called tweets. Many use Twitter to keep up-to-date with events and news happening locally and globally [2]. Additionally, Twitter offers the so-called "hashtag" mechanism that allows users to categorize a tweet's topic and easily search for other tweets that are relevant to those topics [3].

However, the amount of information posted on the Internet is huge, therefore, it is virtually impossible for one to read all of those comments. We think that a bird-eye-view summarization of the product reviews and recommendations on social media may allow consumers to make well-informed buying decisions.

In this article, the authors introduce an approach for brand popularity analysis from Twitter data. To be more specific, we gather statistics and tweets from hashtags associated to cosmetic and beauty products. The objectives are (1) to identify performance measures of social media marketing, (2) to provide sentiment classification of

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_32

tweets mentioning on those brands and their products and (3) to unearth market trends in the cosmetic industry in Thailand (e.g., the most popular brands, the most talking about brands) based on those performance measures and sentiment analysis results.

This article is organized as followed. In the Sect. 2, we review the literature on social media marketing analysis and sentiment classification. Section 3, our data gathering and the analysis of Twitter data are described. The results are displayed in the Sect 4. Finally, we conclude and suggest future research directions.

2 Literature Review

2.1 Social Media Marketing: SMM

Social media marketing (SMM) is a new trend and effective channel for businesses to reach out and target customers. Companies have increasingly used social media channels to promote their brands and products. Social media marketing can be thought of as a subset of online marketing that involves five traditional web-based promotion strategies [4]. Firstly, since almost all social media sites are free to use and even provide free tools for businesses, businesses can run online marketing campaigns with relatively limited budget. Secondly, SMM encourages social interaction which is one of the most notable phenomena of social media. Generally, people spend more than a quarter of their free time on online communication activities. This phenomenon enables businesses to influence customers' behaviors using innovative methods. Thirdly, Social networking sites equip consumers with the ability to more interactive to brands and businesses. Interactivity enables users to access information at a deeper level as well as allows increased user control over and engagement with SMM content. The fourth SMM strategy is target marketing. Social media allow marketers to be able to target audiences and consumers more efficiently and effectively based on consumers' personal interests and what people in their network like. Moreover, social networking enables "word of mouth" to promote products beyond what advertising alone does. Lastly, customer service is another crucial area for SMM. Online customer service on social media platforms is important to success. In addition to basic services, such as toll-free numbers or contact forms, real-time online FAQs or representatives on social media sites are also necessary.

2.2 Social Media Marketing Analytic on Twitter

The return on investment of SMM campaigns should be measured in consumer behaviors or investments in interacting with a brand on social media [5]. From our review, there are several measures used to assess SMM performance on Twitter. Three measures that typically used are as follows:

- Engagement is an important KPI to demonstrate the success of one's campaigns concerning one's brand awareness, and overall market penetration [6].
- Follower rate captures the number of influencers of one's brand. An influencer is a user who can reach a large audience and drive its awareness and opinion about a

trend, brand, company, or product. If people want to voluntarily represent and promote your products or services is a good indicator of popularity and overall success [7].

- Content influential rate is a measure of how effective a brand's tweets is. A good tweet gives you the opportunity to reach and engage with people who may think your content is valuable. Those Twitter users who are interested to your Tweets have the potential to become a part of your primary network if they come back and follow you [7].

2.3 Sentiment Analysis

Sentiment analysis can be defined as "the computational study of people's opinions, appraisals, attitudes, and emotions toward entities, individuals, issues, events, topics and their attributes [8, p. 415]. Researchers have studied the topic at three levels [9]: Document level – sentiment analysis at this level involves classifying whether a whole opinion document expresses a positive or negative sentiment. Sentence level –at this level, the focus is on determining whether each sentence expressed a positive, negative, or neutral opinion. And finally, entity and aspect level– the task at the aspect level looks at the opinion itself. The assumptions are that an opinion consists of a sentiment and a target, and there is limited use for sentiment analysis of an opinion without its target.

Popular supervised learning methods, such as support vector machines and naive Bayesian classification, have been applied to sentiment classification. For example, Pang and colleagues [10] classified movie reviews using these supervised techniques. Similar approaches were also adopted at a sentence level, such as in [11, 12]. However, supervised learning methods have been criticized to be rather domain-specific.Another effective approach is lexicon-based, which is suggested to be more robust to the domain-specific problem [8]. The lexicon-based approach classifies sentiment orientation of a sentence based on sentiment scores of words or phrases in the sentence [13]. Nevertheless, one of its major weaknesses is dictionary generation processes are sometime inconsistent and unreliable [14].

Regarding Twitter sentiment analysis, recent research have focused on classification approaches that require no manual annotation. For example, Tang et al. [15] proposed a use of semantic-specific word embedding features with the state-of-the-art hand-crafted features. Neuron network was used with a training corpora annotated by positive and negative emoticons. Alternatively, a study by Khan and colleagues [16] applied a lexicon-based approach to tag the training dataset later used to train a supervised classifier. They found that the proposed method outperform the state-of-the-art baselines. To diminish the burden of human annotation, we therefore adopted the lexicon-based approach. We use a dictionary with sentiment polarity and score from an online service – i.e., S-Sense [17], provided by a public research lab.

3 Methodology

Figure 1 displays the overview of our "Business Popularity Analysis from Twitter" system. The web application is written in PHP. We also use a SQL database, i.e. MySQL, which is an open-source relational database management system. The key procedures of this system are: (1) obtaining data from Twitter using Tweepy [18], (2) tagging tweets with relevant brands, (3) computing SMM performance measures, and (4) performing sentiment analysis. These procedures are described in detail below.

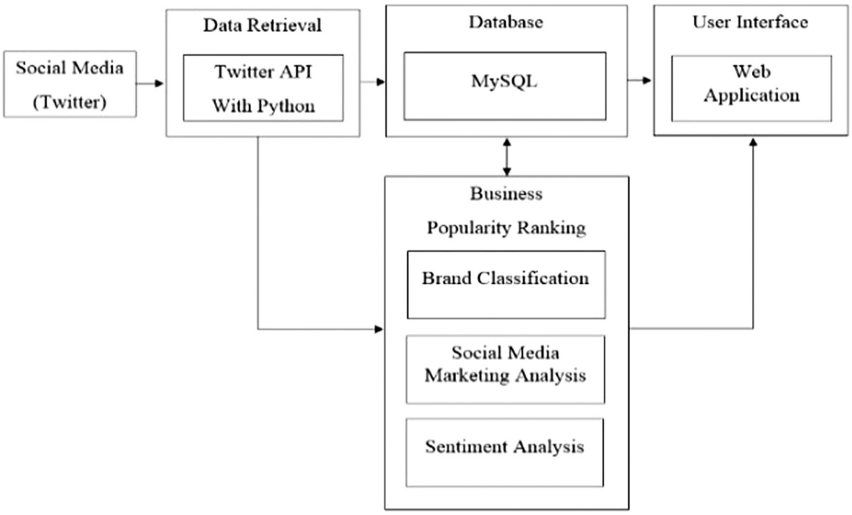


Fig. 1. The overview of business popularity analysis from Twitter system

3.1 Data Retrieval

The main task of this process is to gather data related to cosmetic and beauty products from Twitter. To be more specific, in this study, we collect those tweets with "#HowToPerfect" "#HowToBeauty", "#ฉันจะสวย" (meaning: I will be beautiful), and "#ใช้ดีบอกต่อ" (meaning: it is good, [so I] tell other people) hashtags using Tweepy, which is a Python library for accessing Twitter API [18]. These hashtags are regularly used by reviewers and bloggers of cosmetic and beauty products in Thailand.

The response in Fig. 2 suggests that the Tweet was created on November 6th, 2016 at 10:31:35 by the user named "axxtah". The content of the Tweet is "ตามหาตั้งนานเจอแล้วววว Revlon 365 ดีต่อใจ #HowToPerfect". In addition, the post was retweeted 1,920 times and had 1,001 likes and 412 followers.

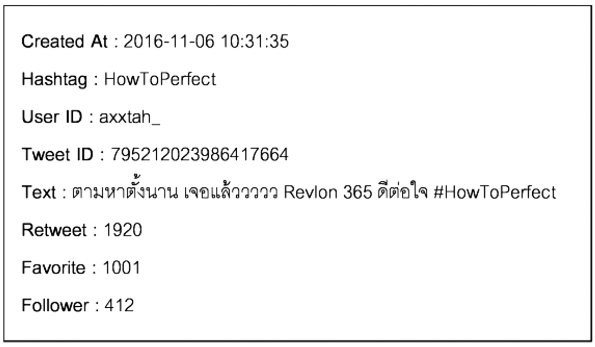


Fig. 2. Data retrieved from Twitter

3.2 Brand Classification

To classify which brand a tweet is related to, we created a database of cosmetic brands being popular in Thailand. From a seed database, when a new tweet mention about a brand that is not yet in the database, the system will automatically add the new brand into the database.

3.3 Social Media Marketing Performance Measures

There are three performance measures adopted in the present study.

Engagement. Engagement is an important KPI that measures how successful one's campaign penetrate the market and reach the audience. The calculation is [6]:

$$\text{Engagement} = \text{Retweet} + \text{Favorite} + \text{Reply} \tag{1}$$

Follower Rate. This measure aims to analyze the effectiveness of an influencer. Influencers often play a role in creating brand awareness and passing on information to others on Twitter. Equation (2) describes its calculation [19]:

$$\text{Follower Rate} = \frac{\text{Active Followers}}{\text{Followers}} \tag{2}$$

Due to the security constraints of Twitter API, it was not possible to obtain the number of active followers. Therefore, we modified the calculation by replacing the "active followers" in Eq. (2) by the number of Twitter users with unique user id that tweet messages containing a hashtag identifying a brand in our database. The "followers" represents the number of followers of the brand's official account.

Content Influential Rate. It assesses how good a brand is communicating with its customers. Whether the brand's messages and campaigns interest the audience and enhance brand awareness. This metric is calculated as in Eq. (3) from [19]:

$$\text{Content Influential Rate of User i} = \text{Engagement}_i/\text{Tweets}_i \quad (3)$$

$$\text{Content Influential Rate} = \sum \text{Content Influential Rate of User i}/\text{Total No. of Users} \quad (4)$$

Please note that we modified the calculation proposed by [17] to consider the influence of all relevant tweets posted by the users during a period of time. More specifically, we calculate average value of influence produced by relevant tweets posted by individuals as in Eq. (4). Table 1 displays some examples of how the proposed performance metrics were calculated.

Table 1. The example of the calculation of performance metrics for each brand

Brand	Engagement	Follower rate	Content influential rate
A'PIEU	= 128 + 34 + 20 = 182	= 179/89356 = 0.002	= 1/2 + 1/1 + 0/3 = 0.167
Etude	= 89 + 21 + 14 = 124	= 111/49032 = 0.0023	= 4/18 + 0/3 + 1/1 + 0/1 = 0.306

3.4 Sentiment Analysis

Since we adopted a lexicon-based approach, the procedure started with a dictionary development. We created a Thai dictionary and obtained the sentiment score and polarity of all opinion-baring words from S-Sense using S-Sense API. Next, we calculated the sentence-level sentiment score of all tweets that we had collected as follow. First, the system segments a sentence into a series of words. Second, sentiment scores of positive words obtained from the dictionary are summed up as $\sum$Positive and scores of positive words are summed up as $\sum$Negative. Third, the sentence-level sentiment score can be computed as in Eq. (5). Since, the result ranges between −1 and 1, it is convenient for evaluation and comparison [20].

$$\text{Sentiment Score} = \frac{(|\sum \text{Positive}| - |\sum \text{Negative}|)}{(|\sum \text{Positive}| + |\sum \text{Negative}|)} \quad (5)$$

3.5 Business Popularity Analysis Ranking

One of the key features of our system is "popularity ranking". We suggest that a business's popularity on Twitter is influenced by how well the company is doing with regard to the proposed four KPIs: engagement, follower rate, content influential rate, and sentiment analysis results. However, the degree of impact of these measures on business popularity may differ. Therefore, we conducted a multiple regression analysis with an aim to statistically weight the impact of those factors. The dependent variable is "business popularity" measured in terms of the number of tweets (tagged with the target four hashtags mentioned in Sect. 3-A) that mention the company's brand name or products. The independent variables are the four performance measures.

With regard to how the system ranks business popularity, since the ranges of all four measures are vary, we had to perform normalization. A z-score normalization method was adopted, which can be calculated as in Eq. (6).

$$z\text{ - score}_i = (x_i - \bar{x}_i)/s_i \tag{6}$$

Where: z-score$_i$ is the z-score of measure I of a particular brand; x_i is the score of measure i of a particular brand; $\bar{x}_i$ is the average value of measure i computing all brands; and s is the standard deviation of measure i computing all brands. The z-score together with weight from the regression analysis are used to rank "business popularity" on our web application. The formulas are as follows:

$$\text{Weighted Score} = Z * \text{Weight} \tag{7}$$

$$\text{Total Score} = \sum \text{Weighted Score} \tag{8}$$

4 Analysis Results and Implementation

For experimental purpose, data are collected daily in November 2016, from twitter with hashtag #HowToPerfect #HowToBeauty #ใช้ดีบอกต่อ #ฉันจะสวย, which are hashtags commonly used for tweets on cosmetic and beauty products in Thailand. The collection has 11,485 records with totally 75 brands. The collected data were analyzed to produce a business popularity ranking. The analysis is described below.

4.1 Brand Identification and Visual Analysis

The brand name is extracted from each tweet using brand name database, which was prepared from previously collected data. The top brand popularity analysis based on each factor is investigated. Image on the left of Fig. 3 shows a pie graph representing numbers and percentages of engagement, for top 10 brands in November 2016; while image on the right shows the daily trend of tweets of 3CE brand.

4.2 Correlation and Regressive Analysis

In order to assess the degree of impact of each factor on business popularity, correlation and regression analysis are performed. We defined the dependent variable as Total Tweet (TT) and four independent variables as: Engagement (Eng), Follower Rate (FR), Content Influential Rate (CIR) and Sentiment Score (SS). We then performed

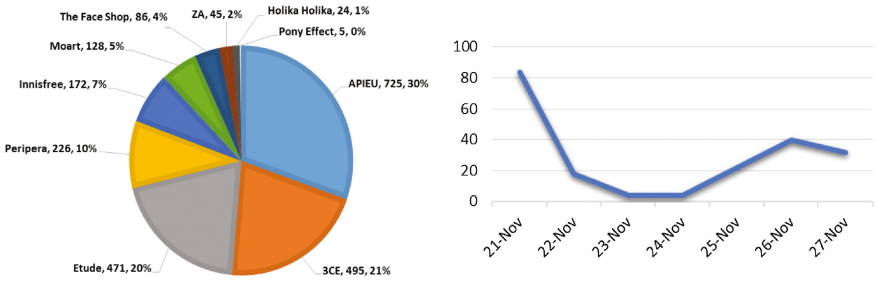


Fig. 3. Top 10 engagement brands and daily trend of 3CE brand in November 2016

correlation and regression analysis. For this purpose, 33 brands with substantial TT in November are selected for correlation and regression analysis.

Correlation Analysis Results. For the four independent variables, no pair has correlation value more than 0.8 or less than −0.8. In addition, the values of variance inflation factor (VIF) for Eng, FR, CIR and SS are 1.097, 1.017, 1.102 and1.025, respectively. None of them has value over 5. Hence, there is no multicollinearity among the four variables.The result also signifies that Eng has the highest positive correlation to TT; while the other three factors have minor correlations with TT.

Regression Analysis Results. The result is displayed in Table 2. The R-square value is 0.998, that is, the model explains 99.8% of the variation of TT variable.Further, the p-value of the F-test is less than 0.01, therefore we reject hull hypothesis at a 99% level of confidence. We conclude that the predictive capability of the model is significantly different from the base model with only the intercept.

Table 2. The result of regression analysis – ANOVA table

Model	Sum of squares	df	Mean square	F	Sig.
Regression	699761.590	4	174940.397	3733.697	0.000
Residual	1311.925	28	46.854		
Total	701073.515	32			

a. dependent variable: total tweet
b. predictors: (constant), Eng, FR, CIR, SS

Nevertheless, as displayed in Table 3, only Eng and CIR factors are significant, with p-value 0.000 and 0.009, respectively; while FR and SS factors have p-values of 0.089 and 0.855, respectively. The standardized coefficients describe the degree of impact of each popularity factor on Total Tweet. The results suggest that Eng is the best contributor 1.004, followed by CIR −0.024, FR 0.015, and SS 0.002, respectively.

Table 3. The result of regression analysis – coefficient table

Model	Unstandardized coefficients		Standardized coefficients	t	Sig.
	B	Std. error	Beta		
(Constant)	3.843	2.653		1.448	0.159
Eng	3.922	0.033	1.004	117.23	0.000
FR	296.634	168.577	0.015	1.760	0.089
CIR	−21.309	7.577	−0.024	−2.812	0.009
SS	0.610	3.301	0.002	0.185	0.855

4.3 Business Popularity Ranking

The four factors for business popularity analysis can be divided into two dimensions, the social media marketing dimension consisting of engagement, follower rate, content influential rate and the sentiment analysis dimension focusing on sentiment score. In order to compare and rank brands, raw data of the four factors are normalized into standard normal (z) as shown in Table 4 (the normalized values are in the parentheses). In addition, based on the previous correlation and regression analysis, we relied on the conclusion that Eng is the most significant factor, followed by CIR, while FR and SS are insignificant. We derived weights of the factors based on this conclusion. To simplify, we give the highest weight of 3 to Eng, the next highest weight of 2 to CIR, and FR and SS have equal weight at 1 as shown in Table 4. We then calculated the total score of each brand by multiplying normalized score with its respective weight. The total score was then used to rank business popularity.

Table 4. The result of z - score with weight and total

Brand	Eng	FR	CIR	SS	Total score
	W = 3	W = 1	W = 2	W = 1	
A'PIEU	4.3860 (1.462*3)	−0.3700 (−0.3700*1)	0.7750 (−0.3875*2)	0.8092 (0.8092*1)	5.60023
Etude	1.8573 (0.6191*3)	−0.4200 (−0.4200*1)	0.0250 (−0.0125*2)	−0.7292 (−0.7292*1)	0.73307
Daiso	−2.3718 (−0.7906*3)	1.6300 (1.6300*1)	2.0250 (−1.0125*2)	−0.7446 (−0.7446*1)	0.53859
Innisfree	−1.6305 (−0.5435*3)	−0.3700 (−0.3700*1)	0.4625 (−0.2313*2)	1.3477 (1.3477*1)	−0.19031
MAC	−2.2410 (−0.747*3)	−0.4700 (−0.4700*1)	−3.2875 (1.6438*2)	−0.6831 (−0.6831*1)	−6.68157

4.4 The Web Application

The analysis results as described on the previous section are displayed as numbers, text, graphs and charts on our business popularity web application. The target users are business people as well as end-customers. The design focuses on user-friendliness and easy-to-understand contents as followed. Image on the left of Fig. 4 is a part of our web application that shows the percentage of tweets among the ten most popular brands (based on our analysis), showing as a pie chart for better grasp of information.

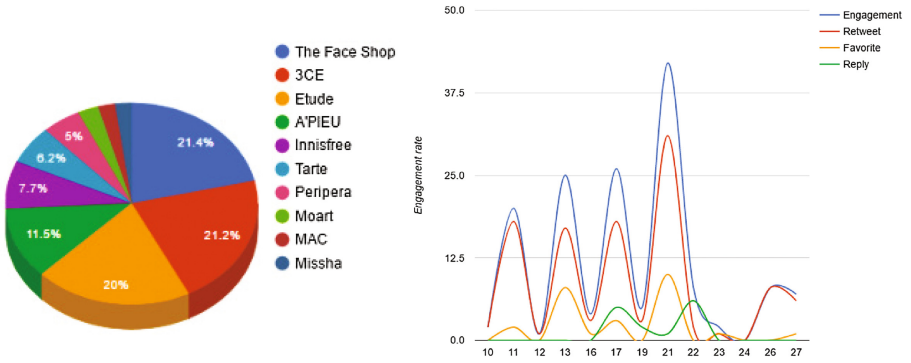


Fig. 4. Tweets percentage of the top 10 brand and trend analysis of A'PIEU brand

In addition, the proposed performance measures (i.e., Eng, FR, CIR, SS, and TT) are presenting as line charts to show daily trend for each brand. Image of the left of Fig. 4 is an example of the daily engagement for A'PIEU brand.For sentiment analysis, the web application does not only display sentiment scores but also text comments, which are classified into objective, positive and negative sentiment.

5 Discussion and Conclusion

The number of online buyers increases rapidly and this makes online commercial marketing become key factor to attract buyers. However, not only the business people can advertise their product on this channel, the buyers themselves can actively participate in this activity. Twitter is one of the popular channels for social commerce marketing. This paper proposed a methodology to access and rank brand popularity based on the activities on Twitter.

With regard to future directions, a study that explores other factors influencing business popularity on social media could be an interesting one. Moreover, since one of the weaknesses of the lexicon-based approach is the dictionary, a study that tries to develop a more complete dictionary that concerns types of words, negation, intensifier, etc. for Thai words commonly used on social media may be a fruitful one. Such a dictionary could help enhance our sentiment classification performance. Finally, an

integration of data from all major social media sites, such as Facebook, Twitter, Instagram, etc.may provide a more completed view of business's popularity and trends.

In summary, we have collected data on cosmetic products in November, 2016 and demonstrated the following tasks. First, analyzing and virtualizing statistics provided by Twitter. Second, classifying Twitter posts' sentiment into positive, negative and commercial. Third, identifying factors that influence overall brand popularity based on the correlation and regression analysis results. The regression results suggest that measures like engagement and content influence rate is an important predictor for a brand's popularity on Twitter. Finally, identifying and ranking brands being in trend. The results are presented using tables, graphs and charts on our develop web application. The information is useful for both business users and consumers.

References

1. Amedie, J.: The Impact of Social Media on Society. Advanced Writing: Pop Culture Intersections. Santa Clara University, California (2015)
2. Akshay, J., Song, X., Finin, T, Tseng, T.: Why we twitter: understanding microblogging usage and communities. In: The 9th Web KDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis, pp. 56–65. ACM, New York (2007)
3. HootSuite: Hashtags. http://socialbusiness.hootsuite.com/rs/hootsuitemediainc/images/whitepaper-hashtags.pdf
4. Nadaraja, R., Yazdanifard, R.: Social Media Marketing: Advantages and Disadvantages. Social Media Marketing, pp. 1–10. Centre of Southern New Hampshire University, Kuala Lumpur (2013)
5. Hoffman, D.L., Fodor, M.: Can you measure the ROI of your social media marketing? MIT Sloan Manag. Rev. **52**(1), 41–49 (2010)
6. Shively, K.: Twitter metrics defined: engagement. http://simplymeasured.com/blog/twitter-metrics-defined-engagement/
7. Fontein, D.: The top 26 social media KPIs marketers can't ignore. https://blog.hootsuite.com/social-media-kpis-key-performance-indicators/
8. Bing, L., Zhang, L.: A survey of opinion mining and sentiment analysis. In: Aggarwal, C.C., Zhai, C. (eds.) Mining Text Data, pp. 415–463. Springer, New York (2012)
9. Bing, L., Zhang, L.: Sentiment analysis and opinion mining. In: Sammut, C., Webb, G.I. (eds.) Encyclopedia of Machine Learning and Data Mining, pp. 1–10. Springer, New York (2016)
10. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: The 2002 Conference on Empirical Methods in Natural Language Processing, pp. 79–86. Association for Computational Linguistics, Stroudsburg (2002)
11. Yu, H., Hatzivassiloglou, V.: Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In: The 2003 Conference on Empirical Methods in Natural Language Processing, pp. 129–136. Association for Computational Linguistics, Stroudsburg (2003)
12. Gamon, M., Aue, A., Corston-Oliver, A., Ringger, E.: Pulse: mining customer opinions from free text. Adv. Intell. Data Anal. **1**, 121–132 (2005)
13. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Comput. Linguist. **37**(2), 267–307 (2011)

14. Andreevskaia, A., Bergler, S.: When specialists and generalists work together: domain dependence in sentiment tagging. In: The 46th Annual Meeting of the Association for Computational Linguistics, pp. 290–298. ACL, Columbus (2008)
15. Tang, D., Wei, F., Qin, B., Liu, T., Zhou, M.: Coooolll: a deep learning system for twitter sentiment classification. In: the 8th International Workshop on Semantic Evaluation (SemEval-2014), pp. 208–212 (2014)
16. Khan, A.Z.H., Atique, M., Thakare, V.M.: Combining lexicon-based and learning-based methods for Twitter sentiment analysis. Int. J. Electron. Commun. Soft Comput. Sci. Eng. 89–91 (2015). Special Issue of IJECSCSE (ATCON-2015)
17. Nectec.: S-sense social sensing. http://www.ssense.in.th/
18. Tweepy. http://www.tweepy.org/
19. Mosenson, P.: Are my social media efforts effective? https://www.nusparkmarketing.com/SocialMediaMetrics.pdf
20. Batra, A.: Sentiment indicator: social media KPI. https://webanalysis.blogspot.com/2011/09/sentiment-indicator-social-media-kpi.html#axzz4ad71FVya

Language and Text-Independent Speaker Recognition System Using Energy Spectrum and MFCCs

Pafan Doungpaisan[1(✉)] and Anirach Mingkhwan[2]

[1] Faculty of Information Technology,
King Mongkut's University of Technology North Bangkok, Bangkok, Thailand
pafan@kmutnb.ac.th
[2] Faculty of Industrial Technology and Management, King Mongkut's
University of Technology North Bangkok, Prachinburi, Thailand
Anirach.M@fitm.kmutnb.ac.th

Abstract. Speaker identification, especially in critical environments, has always been a subject of great interest. In this paper, we present a language and text independent speaker identification algorithm that able to automatically identify a speaker in an audio signal with noise or real environment sound in background. The method is inspired by using a pairing of Energy spectrum and MFCCs audio feature techniques generated from base on Discrete Fourier transform (DFT). After that the audio feature extracted in real time was compared with a Euclidean Distance to measures of different between speakers to obtain the most likely speakers. The Energy spectrum feature is adopted to supplement the MFCC features to yield higher recognition accuracy for speaker identification sound.

The proposed technique is test with 30 different speakers in three languages. The experimental result on speaker identification algorithm using an Energy spectrum and MFCCs features with Euclidean Distance can effectively identify speaker in noise or real environment sound in background with a language and text independent more than 83%. Notably, our approach is not language-specific; it can identify speaker in more than one language.

Keywords: Speaker identification · Energy spectrum · MFCCs

1 Introduction

Speech is the product of a complex behavior conveying different speaker specific nature that are potential sources of complementary information. Historically, speech signal processing and analysis has attracted wide consideration. Especially by using varied applications. For instance, automatic speaker recognition (ASR) have been research areas at least since earlier 70s [1]. Recently, voice has catches again researchers attention its usefulness in order to assess early vocal pathologies, neurodegenerative and mental disorders among others [2]. Progress achieved in these new applications have allowed for a better understanding of the resource of voice production, which have led to an improvement in speaker feature to solve the speaker recognition problem.

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_33

Speaker identification is one of the main tasks in speech processing. In addition to identification accuracy, large scale applications of speaker identification give rise to another challenge: fast search in the database of speakers. Research about Speaker recognition, there are two different types of Speaker Recognition [3, 4] consist of Speaker Verification and Speaker Identification.

Speaker verification is the process of verifying the claimed identity of a speaker based on the speech signal from the speaker call a voiceprint. In speaker verification, a voiceprint of an unknown speaker who claims an identity is compared with a model for the speaker whose identity is being claimed. If the match is good enough, the identity claim is accepted. A high threshold reduces the probability of impostors being accepted by the system, increasing the risk of falsely rejecting valid users. On the other hand, a low threshold enables valid users to be accepted consistently, but with the risk of accepting impostors. In order to set the threshold at the optimal level of impostor acceptance or false acceptance and customer rejection or false rejection. The data showing impostor scores and distributions of customer are needed.

There are two types of speaker verification systems: Text-Independent Speaker Verification and Text-Dependent Speaker Verification. Text-Dependent Speaker Verification requires the speaker saying exactly the enrolled or given password. Text independent Speaker Verification is a process of verifying the identity without constraint on the speech content. Compared to Text-Dependent Speaker Verification, it is more convenient because the user can speak freely to the system. However, it requires longer training and testing utterances to achieve good accuracy and performance.

In the speaker identification task, a voice of an unknown speaker is analyzed and then compared with speech samples of known speakers. The unknown speaker is identified as the speaker whose model best matches the input model. There are two different types of speaker identification consist of open-set and closed-set.

Open-set identification similar as a combination of closed-set identification and speaker verification. For example, a closed-set identification may be proceed and the resulting ID may be used to run a speaker verification session. If the test speaker matches the target speaker, based on the ID returned from the closed-set identification, then the ID is accepted and it is passed back as the true ID of the test speaker. On the other hand, if the verification fails, the speaker may be rejected all together with no valid identification result. Closed-set identification is the simpler of the two problems. In closed-set identification, the audio of the test speaker is compared against all the available speaker models and the speaker ID of the model with the closest match is returned. In closed-set identification, the ID of one of the speakers in the database will always be closest to the audio of the test speaker; there is no rejection scheme.

This research, we have worked on language and text-independent speaker verification. Research interesting of speaker recognition such as. Research of Poignant, J. [5] used unsupervised way to Identifying speakers in TV broadcast without biometric models. Existing methods usually use pronounced names, as a source of names, for identifying speech clusters provided by a speaker divarication step but this source is too imprecise for having sufficient confidence. There propose two approaches for finding speaker identity based only on names written in the image track such as with the “late naming” and “Early naming”. These methods were tested on the REPERE corpus phase 1, containing 3 h of annotated videos. With the “late naming” system reaches an

F-measure of 73.1%. With the "early naming" improves over this result both in terms of identification error rate and of stability of the clustering stopping criterion. By comparison, a mono-modal, supervised speaker identification system with 535 speaker models trained on matching development data and additional TV and radio data only provided a 57.2% F-measure.

Research of M.K. Nandwana [6] focused on an unsupervised approach for detection of human scream vocalizations from continuous recordings in noisy acoustic environments. The proposed detection solution is based on compound segmentation, which employs weighted mean distance, T2-statistics and Bayesian Information Criteria for detection of screams. This solution also employs an unsupervised threshold optimized Combo-SAD for removal of non-vocal noisy segments in the preliminary stage. A total of five noisy environments were simulated for noise levels ranging from −20 dB to +20 dB for five different noisy environments. Performance of proposed system was compared using two alternative acoustic front-end features (i) Mel-frequency cepstral coefficients (MFCC) and (ii) perceptual minimum variance distortion less response (PMVDR). Evaluation results show that the new scream detection solution works well for clean, +20, +10 dB SNR levels, with performance declining as SNR decreases to −20 dB across a number of the noise sources considered.

Research of Almaadeed, N. [7] is to investigate the problem of identifying a speaker from its voice regardless of the content. In this study, the authors designed and implemented a novel text-independent multimodal speaker identification system based on wavelet analysis and neural networks. The related system, found to be competitive and it improved the identification rate by 15% as compared with the classical MFCC. In addition, it reduced the identification time by 40% as compared with the back propagation neural network, Gaussian mixture model and principal component analysis. Performance tests conducted using the GRID database corpora have shown that this approach has faster identification time and greater accuracy compared with traditional approaches, and it is applicable to real-time, text-independent speaker identification systems.

Research of Xiaojia Zhao [8] investigates the problem of speaker identification and verification in noisy conditions, assuming that speech signals are corrupted by environmental noise. This paper is focused on several issues relating to the implementation of the new model for real-world applications. These include the generation of multi-condition training data to model noisy speech, the combination of different training data to optimize the recognition performance, and the reduction of the model's complexity. The new algorithm was tested using two databases with simulated and realistic noisy speech data. The first database is a redevelopment of the TIMIT database by rerecording the data in the presence of various noise types, used to test the model for speaker identification with a focus on the varieties of noise. The second database is a handheld device database collected in realistic noisy conditions, used to further validate the model for real-world speaker verification. The new model is compared to baseline systems and is found to achieve lower error rates.

Pathak, M.A. and Raj, B., [9] present frameworks for privacy preserving speaker verification and speaker identification systems, where the system is able to perform the necessary operations without being able to observe the speech input provided by the user. In this paper we formalize the privacy criteria for the speaker verification and

speaker identification problems and construct Gaussian mixture model-based protocols. We also report experiments with a prototype implementation of the protocols on a standardized dataset for execution time and accuracy.

Bhardwaj, S. [10] presents three novel methods for speaker identification of which two methods utilize both the continuous density hidden Markov model (HMM) and the generalized fuzzy model (GFM), which has the advantages of both Mamdani and Takagi Sugeno models. In the first method, the HMM is utilized for the extraction of shape based batch feature vector that is fitted with the GFM to identify the speaker. On the other hand, the second method makes use of the Gaussian mixture model (GMM) and the GFM for the identification of speakers. Finally, the third method has been inspired by the way humans cash in on the mutual acquaintances while identifying a speaker. To see the validity of the proposed models [HMM-GFM, GMM-GFM, and HMM-GFM (fusion)] in a real life scenario, they are tested on VoxForge speech corpus and on the subset of the 2003 National Institute of Standards and Technology evaluation data set. These models are also evaluated on the corrupted VoxForge speech corpus by mixing with different types of noisy signals at different values of signal-to-noise ratios, and their performance is found superior to that of the wellknown models.

This paper proposes a speaker verification algorithm that able to automatically identify a speaker in an audio signal with noise or real environment sound in background. The method is inspired by using the Energy spectrum audio feature techniques generated from base on Discrete Fourier transform (DFT). The method is made of two phases: First, the characteristic of the user's voice is generated from components of sound. Second, the characteristic extracted in real time are compared with the Speaker sound using a Euclidean Distance to measures of different between speakers to obtain the most likely speakers.

The rest of the paper is organized as follows. The detail of our proposed algorithm described in Sect. 2. Experimental results showed in Sect. 3 and Sect.4 concludes paper.

2 Methodology

Figure 1 shows a Content-based Speaker Identification Framework. The method is in-spired by using a concatenation of the Energy spectrum and MFCCs features. First, of speaker was extracted without needing a filtering phase. All audio windows were extracted comprehensive characteristic of speaker sound are belonging to two components of sound consist of Energy spectrum and MFCCs feature to yield higher recognition accuracy for speaker identification sound.

In Fig. 2, The Mel-frequency cepstral coefficients (MFCCs) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. These features are typically obtained by first applying a Fourier transform to short-time window segments of audio signals followed by further processing to derive the features of interest. Some commonly used ones include the MFCC [11]: After taking the FFT of each short-time window, the first step in MFCC calculation is to obtain the mel filter bank outputs

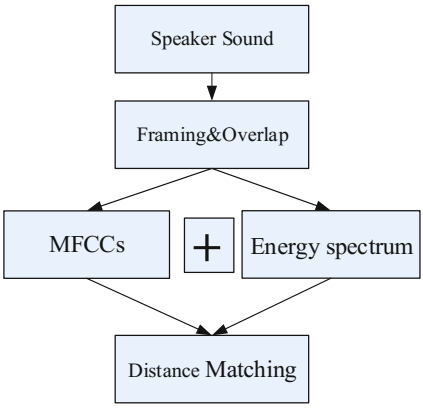


Fig. 1. Content-based Speaker Identification Framework

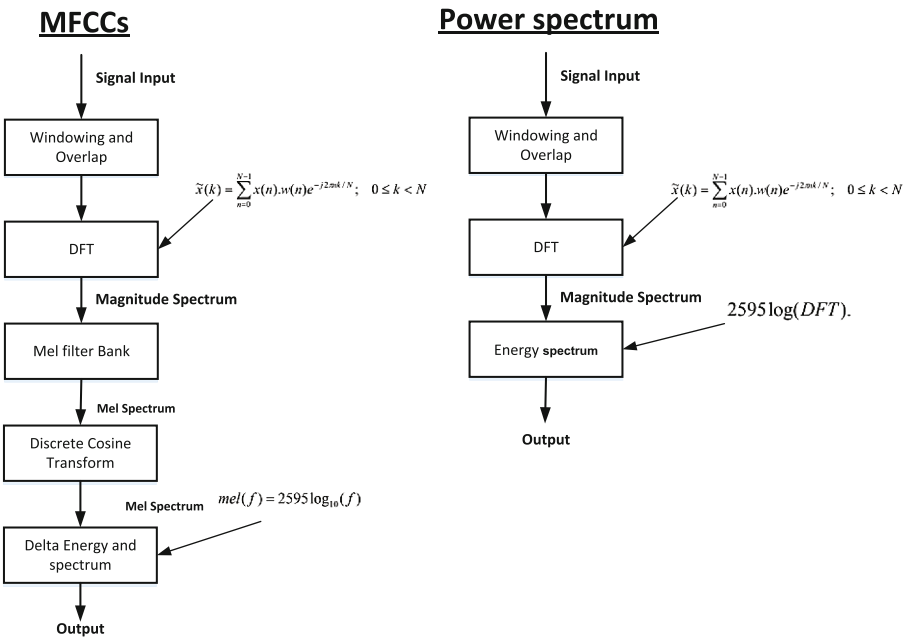


Fig. 2. To calculate the energy spectrum (power spectrum) and calculated MFCCs

by mapping the powers of the spectrum onto the mel scale, using 23 triangular mel filter bank, and transformed into a logarithmic scale, which emphasizes the low varying frequency characteristics of the signal. Typically, 13 Mel-frequency cepstral coefficients are then obtained by taking the discrete cosine transform (DCT).

Figure 2, the process of creating Energy spectrum features. The first step is to segmenting the audio signal into frames with the length with in the range is equal to a

power of two, usually by applying Hamming window function. The next step is to take the Discrete Fourier Transform (DFT) of each frame. The next step is to take the power of each frames, denoted by *P(k)*, is computed by the following equation.

$$P(k) = 2595 \log(DFT) \tag{1}$$

The result of *P(k)* is called Energy spectrum.

3 Experimental Evaluation

3.1 Data Collection

Audio data used for this experiment included 303 files, total length of 130 h or 7855 min. Sound clips was take from two different sources, the teachings of the MIT OpenCourseWare (http://ocw.mit.edu/courses/audio-video-courses/) and YouTube website (https://www.youtube.com/). All audio files consist of 30 people in three languages with varied environments sound in background including the meeting rooms of various sizes, office, construction site, television studio, streets, parks, the International Space Station. All downloaded video files was used Pazera Audio Extractor to extract audio tracks from video. All audio files after extracted are code in the Wave Files (for uncompressed data, or data loss) Mono Channel and sample rate at 11,025 Hz. We chose this sample rate because the human range is commonly given as 20 to 20,000 Hz, though there is considerable variation between individuals, especially at high frequencies, and a gradual loss of sensitivity to higher frequencies with age is considered normal.

3.2 Measure of Similarity

The purpose of a measure of similarity is to compare two vectors and compute a single number that evaluates their similarity. Euclidean distance often used to compare profiles of respondents across variables. For example, suppose our data consist of demographic information on a sample of individuals, arranged as a respondent-by-variable matrix. Each row of the matrix is a vector of m numbers, where m is the number of variables. We can evaluate the similarity or the distance between any pair of rows. Euclidean Distance is the basis of many measures of similarity and dissimilarity is Euclidean distance. The distance between vectors X and Y defined as follows:

$$\left|d_j - d_k\right| = \sqrt{\sum_{i=1}^{n} \left(d_{i,j} - d_{i,k}\right)^2} \tag{2}$$

In other words, Euclidean distance is the square root of the sum of squared differences between corresponding elements of the two vectors. Note that the formula treats the values of X and Y seriously: no adjustment is made for differences in scale. Euclidean distance is only appropriate for data measured on the same scale. As you will

see in the section on correlation, the correlation coefficient is related to the Euclidean distance between standardized versions of the data.

Here is an algorithm step by step on how to use Euclidean Distance to measures a different between speaker:

1. Calculate the distance between the query instance and all the training samples each category Y.
2. Sort the distance and determine nearest samples based on the minimum distance.
3. Gather the category Y of the minimum distance nearest samples.
4. Use simple majority of the category of nearest samples as the prediction value of the query instance.

4 Result

In each experiment, we performed 50 runs of the 5-fold cross-validation to obtain statistically reliable results. The mean recognition rate was calculated based on the error average for one run on test set. We examined the performance of the Energy spectrum and Mel Frequency Cepstral Coefficents (MFCCs) as described in Sect. 2, a concatenation of the Energy spectrum and MFCCs to form a feature vector as showing in Fig. 2. For statistically reliable results, we compare the overall recognition accuracy using an Energy spectrum and MFCCs with a variety of Distance Measure algorithm as shown in Table 1.

Table 1. Summarized the average accuracy of all Distance Measure.

Feature	Distance measure	Accuracy (%)
Energy spectrum + MFCCs	Euclidean	81.62
	Cityblock	80.40
	Cosine	46.22
	Correlation	44.47
MFCCs	Euclidean	74.40
	Cityblock	72.62
	Cosine	59.36
	Correlation	54.73
Energy spectrum	Euclidean	78.27
	Cityblock	77.98
	Cosine	43.98
	Correlation	44.64

From results in Table 1, by using concatenation of the Energy spectrum and MFCCs was performed better recognition accuracy than using an Energy spectrum or MFCC only for all Distance Measure algorithm. The highest performance of Energy spectrum and MFCC was obtain by using Euclidean Distance 81.62%.

Next, we examined the performance of concatenation of the Energy spectrum and MFCCs with another feature vector such as Spectral centroid, Discrete Fourier Transform, Haar Discrete Wavelet Transform, Mel Frequency Cepstral Coefficents

Table 2. Summarized the average accuracy of Euclidean Distance Measure and all Feature.

Feature	Accuracy (%)
Energy spectrum + MFCCs	81.62
Energy spectrum	78.27
Spectral centroid	12.03
Discrete Fourier Transform	62.36
Haar Discrete Wavelet Transform	16.75
Mel Frequency Cepstral Coefficents (MFCCs)	74.40
RollOff	11.40
Root Mean Square (RMS)	26.22
Linear prediction (LP)	48.26
perceptual linear prediction (PLP) coecients	24.02
partial correlation coefficients (PARCORs)	68.18

(MFCCs), RollOff, Root Mean Square (RMS), Linear prediction (LP), perceptual linear prediction (PLP) coecients and partial correlation coefficients (PARCORs). We comparable performance to another feature extraction method on a similar task. The result was show in Table 2.

From results in Table 2, by compare accuracy all feature with Energy spectrum and MFCC. The Energy spectrum and MFCC was show the best accuracy.

5 Summary

The paper reports a concatenation of the Energy spectrum and MFCCs a small set of time–frequency features, which is flexible, intuitive and physically interpretable. A combination of Energy spectrum and MFCC features can identification a speaker sounds in real environment and improve the overall performance.

The experimental results show promising performance in identifying a different audio speaker and shows comparable performance to another feature extraction method on a similar task. By using Energy spectrum and MFCC was show the best accuracy when compare accuracy all feature. Notably, our approach is not language-specific; it can identify speaker in more than one language.

References

1. Rosenberg, A. E.: Automatic speaker verification: a review. In: Proceedings of the IEEE, pp. 475–487 (1976)
2. Gómez Vilda, P., Rodellar Biarge, V., Nieto Lluis, V., Muñoz Mulas, C., Mazaira-Fernández, L.M., Martínez Olalla, R.: Characterizing neurological disease from voice quality analysis. Cognit. Comput. **5**(4), 399–425 (2013)
3. Furui, S.: Digital Speech Processing: Synthesis, and Recognition. CRC Press, New York (1989)
4. Hansen, J.H.L.: Analysis and compensation of speech under stress and noise for environmental robustness in speech recognition. Speech Commun. **20**(1–2), 151–173 (1996)
5. Poignant, J., Besacier, L., Quénot, G.: Unsupervised speaker identification in TV broadcast based on written names. IEEE Trans. Audio Speech Lang. Process. **23**(1), 57–68 (2015)
6. Nandwana, M.K., Ziaei, A., Hansen, J.H.L.: Robust unsupervised detection of human screams in noisy acoustic environments. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 161–165, South Brisbane (2015)
7. Almaadeed, N., Aggoun, A., Amira, A.: Speaker identification using multimodal neural networks and wavelet analysis. IET Biom. **4**(1), 18–28 (2015)
8. Zhao, X., Wang, Y., Wang, D.: Robust speaker identification in noisy and reverberant conditions. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3997–4001, Florence (2014)
9. Pathak, M.A., Raj, B.: Privacy preserving speaker verification and identification using gaussian mixture models. IEEE Trans. Audio Speech Lang. Process. **21**(2), 397–406 (2013)
10. Bhardwaj, S., Srivastava, S., Hanmandlu, M., Gupta, J.R.P.: GFM-based methods for speaker identification. IEEE Trans. Cybernet. **43**(3), 1047–1058 (2013)
11. Rabiner, L., Juang, B.H.: Fundamentals of Speech Recognition. PTR Prentice Hall, Englewood Cliffs (1993)

Author Index

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7