Advances in Intelligent Systems and Computing 650

Fei Chao
Steven Schockaert
Qingfu Zhang *Editors*

Advances in Computational Intelligence Systems

Contributions Presented at the 17th UK Workshop on Computational Intelligence, September 6–8, 2017, Cardiff, UK

Springer

Advances in Intelligent Systems and Computing

Volume 650

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.

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

Fei Chao · Steven Schockaert
Qingfu Zhang
Editors

Advances in Computational Intelligence Systems

Contributions Presented at the 17th UK Workshop on Computational Intelligence, September 6–8, 2017, Cardiff, UK

Springer

Editors

Fei Chao
Xiamen University
Xiamen Shi, Fujian
China

Steven Schockaert
Cardiff University
Cardiff
UK

Qingfu Zhang
Department of Computer Science
City University of Hong Kong
Kowloon
Hong Kong

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-66938-0 ISBN 978-3-319-66939-7 (eBook)
DOI 10.1007/978-3-319-66939-7

Library of Congress Control Number: 2017951425

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

This volume contains the papers to be presented at the 17th UK Workshop on Computational Intelligence (UKCI 2017), which will be held in Cardiff, UK, on 6–8 September 2017. Since 2001, UKCI has been the UK premier forum for presenting leading research on all aspects of Computational Intelligence. The overall objective of UKCI is to encourage the academic community and industry to share and exchange ideas on theoretical and practical aspects of Computational Intelligence techniques.

UKCI 2017 has attracted 40 submissions, on areas such as fuzzy systems, neural networks, evolutionary computation, machine learning, data mining, robotics, and big data. A growing number of researchers focus on solving problems related to traffic congestion, which is an important global challenge. To emphasise this important trend, UKCI 2017 has featured a special track on Intelligent Transportation.

Each paper was reviewed by at least three members of the programme committee. Based on their recommendations, 34 papers have been accepted for publication (25 long papers and nine short papers), of which 32 appear in this volume (after two papers have been withdrawn). These papers have been organised into five sections: (1) Modelling and Representation, (2) Optimisation, (3) Learning, (4) Control and Human-Machine Systems, and (5) Intelligent Transportation.

Although UKCI has been advertised mainly as a national event for the UK, it has always attracted significant attention from further afield. UKCI 2017 continued this trend by featuring papers and participants from a number of countries on several continents, including Saudi Arabia, Japan, China, Hong Kong, Turkey, and Singapore. In this respect, UKCI 2017 is a clear manifestation of the fact that academic research is international and collaborative by nature.

The UKCI 2017 programme also featured keynote talks by established researchers in the field of Computational Intelligence.

Finally, we would like to thank everyone who contributed to the success of UKCI 2017, the members of the programme and organising committees, the keynote speakers, the authors, and the presenters of papers. We are grateful for support from the Welsh Government for the organisation of the special track on Intelligent Transportation.

July 2017

Fei Chao
Steven Schockaert
Qingfu Zhang

Organisation

Programme Committee

Giovanni Acampora	University of Naples Federico II, Italy
Donglin Cao	Xiamen University, China
Yidong Chen	Xiamen University, China
George Coghill	University of Aberdeen, UK
Chris Cornelis	Ghent University, Belgium
Damien Coyle	University of Ulster, UK
Keeley Crockett	Manchester Metropolitan University, UK
Sven F. Crone	Lancaster University, UK
Xin Fu	Xiamen University, China
Jonathan M. Garibaldi	University of Nottingham, UK
Alexander Gegov	University of Portsmouth, UK
Christopher Hinde	Loughborough University, UK
Jose Antonio Iglesias	Carlos III University of Madrid, Spain
Shoaib Jameel	Cardiff University, UK
Thomas Jansen	Aberystwyth University, UK
Richard Jensen	Aberystwyth University, UK
Bob John	University of Nottingham, UK
Ondřej Kuželka	Cardiff University, UK
Ke Li	University of Exeter, UK
Han Liu	University of Portsmouth, UK
Honghai Liu	University of Portsmouth, UK
Ahmad Lotfi	Nottingham Trent University, UK
George Magoulas	Birkbeck College, UK
Trevor Martin	University of Bristol, UK
Qinggang Meng	Loughborough University, UK
Daniel C. Neagu	University of Bradford, UK
Samia Nefti	University of Salford, UK
Ann Nowe	Vrije Universiteit Brussel, Belgium

Vasile Palade	Coventry University, UK
Wei Pang	University of Aberdeen, UK
Girijesh Prasad	University of Ulster, UK
Yvan Saeys	Ghent University, Belgium
Araceli Sanchis	Universidad Carlos III de Madrid, Spain
Qiang Shen	Aberystwyth University, UK
Jialong Shi	City University of Hong Kong, Hong Kong
Irena Spasic	Cardiff University, UK
Jianyong Sun	Essex University, UK
Longzhi Yang	Northumbria University, UK
Shengxiang Yang	De Montfort University, UK
Yingjie Yang	De Montfort University, UK
Xiao-Jun Zeng	University of Manchester, UK

Additional Reviewers

Gao, Xingen
Guo, Feng
Jiang, Min
Ju, Zhaojie
Korik, Attila
Li, Xiang
Li, Zhenhua
Lo, Hong
Pavlidis, Nicos
Pedrycz, Witold
Shang, Changjing
Shi, Minghui
Xin, Zhang

Contents

Modelling and Representation

Optimisation

Learning

Control and Human-Machine Systems

Intelligent Transportation

Modelling and Representation

Integrating Association Rules Mined from Health-Care Data with Ontological Information for Automated Knowledge Generation

John Heritage[1], Sharon McDonald[2], and Ken McGarry[1](✉)

[1] School of Pharmacy and Pharmaceutical Sciences,
Facuty of Health Sciences and Wellbeing, University of Sunderland,
City Campus, Sunderland, UK
ken.mcgarry@sunderland.ac.uk

[2] Faculty of Computing, University of Sunderland,
St Peters Campus, Sunderland, UK

Abstract. Association rule mining can be combined with complex network theory to automatically create a knowledge base that reveals how certain drugs cause side-effects on patients when they interact with other drugs taken by the patient when they have two or more diseases. The drugs will interact with on-target and off-target proteins often in an unpredictable way. A computational approach is necessary to be able to unravel the complex relationships between disease comorbidities. We built statistical models from the publicly available FAERS dataset to reveal interesting and potentially harmful drug combinations based on side-effects and relationships between co-morbid diseases. This information is very useful to medical practitioners to tailor patient prescriptions for optimal therapy.

Keywords: Comorbidity · Side-effect · Association rules · Support · Confidence · Pharmaco-epidemiology

1 Introduction

As people age and suffer from several illnesses, they will require more medications. When individuals start taking several medications the chances that the drugs they take will interact in harmful ways will increase. Drug-to-drug interactions are difficult to predict as there are so many confounding factors at work - people vary in their genetic predisposition and thus response to treatment, age, gender, and environmental factors all play a role. Although every drug undergoes rigorous safety trials during its development, these are conducted on participants using only the drug being investigated, it is impossible to conduct the trial any other way. Our knowledge of drug-to-drug interactions, side-effects and disease comorbidity is derived from healthcare record systems and these are now starting to receive increased attention as a way of improving public health and drug safety.

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_1

Collecting data on drug side-effects and carefully analyzing it can reveal much about how drugs are acting in the body and should assist doctors tailor drug prescriptions for their patients [11,12]. This is made possible by identifying shared biological pathways through similar side-effects. The USA and UK have online systems such as the FAERS [16] and *Yellow Card* [5] databases in place for medical professionals to report incidents when patients experience an adverse drug reaction (ADR). Unfortunately, there is a great deal of noise present in these databases and in fact potentially the majority of cases may be anecdotal and unreliable. For example a patient, in the early stages of drug treatment may present themselves at the doctors complaining of headaches, dizziness and feelings of nausea. Some symptoms may not be listed on the drug information sheet and there is a chance it is not a result of taking the drug, perhaps the patient has an additional undiagnosed condition or had taken medicine for flu. However, the value of *big data* patient records comes from the luxury of being able to discard the poor quality, noisy cases and to keep only a fraction [18]. Powerful statistical models need only a few hundred high quality records to perform reliable comparisons that can unravel the complex interactions between drug regimens, patient variability and random chance.

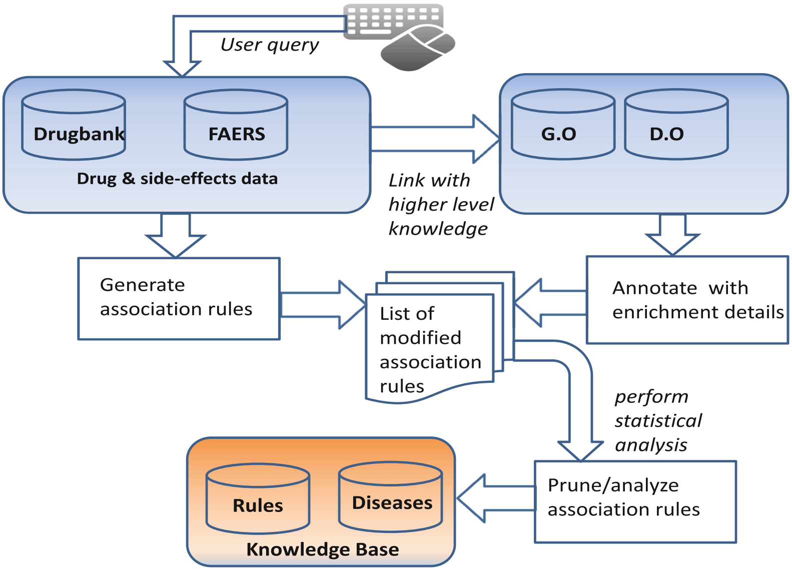


Fig. 1. Overview of system operation, showing database sources, data flow and statistical analysis. The user query initiates a search of the various databases resulting in a knowledge base and ruleset related to the disease of interest.

Referring to Fig. 1 the system is intended to be used by healthcare specialists wishing to test a hypothesis relating to the drugs their patients are currently receiving for a particular disease, their patient may already have a second medical condition and be taking medicine for this. The practitioner would like to query the system to see if anything is known about potential drug conflicts. Clearly, some drug-to-drug interactions are already well known and there would be no need to use this system, but for suspected combinations of drugs the system would be useful.

The process is initiated by a user query containing the disease(s) of interest which accesses the various databases, the results of which are used to build statistical models to assess drugs and to build the knowledge base. We briefly describe the key methods and data.

Association rule mining and frequent item set analysis originated in market basket analysis 20 years ago whereby trends and patterns in consumer purchasing activities could be identified and used to increase profits [1]. Several companies use association rules to influence customers to purchase items, most notably the Amazon recommendation facility which is based on an individual's purchasing history and also the purchasing history of others with similar tastes and preferences [3].

Association rule mining is now starting to receive attention in the bioinformatics field where the majority of the data are binary or categorical in nature. So far, the bulk of association rules research in bioinformatics appears to be oriented towards mining Electronic Health Records (EHR) and Health Information Technology (HIT) on patient clinical information [23] and little attention focused on drug-to-side-effects [22]. In this paper association rules are used to uncover relationships between drugs, their side-effects and co-interacting protein targets. However, it should be noted that the item sets can only show the commonality of items as they appear in the database, however by using association rules and a statistical measure we can imply strong correlation between the associated items.

The databases used include Drugbank for the chemical properties and protein targets of the candidate drugs and SIDER4 for a comprehensive list of drug side-effects. Furthermore, we used the Gene Ontology (GO) for providing details and characteristics of specific proteins and products [2], the Disease Ontology (DO) which relates the various diseases into a taxonomy [17]. KEGG is a store of biological pathways and provides more information on the normal biological process of the cell and how they can be affected by drugs [14].

The remainder of this paper is structured as follows; Sect. 2 describes the system architecture in terms of flow of information, the sources of data and the computational techniques we use; Sect. 3 describes the results; Sect. 4 is the discussion; finally Sect. 5 presents the conclusions and future work.

2 Methods

2.1 Programming Environment

We implemented the system using the R language with the RStudio programming environment. R is primarily a statistical data analysis package but is gaining popularity for various scientific programming applications and is very extendable using packages written by other researchers [15]. We used the following R packages: aRules [7], GOSim [24]. Our R code and data files are freely available to all researchers on GitHub for download: https://github.com/kenmcgarry/UKCI2017-AR.

2.2 Databases, Ontologies and Pre-processing

The Gene Ontology (GO), KEGG and Disease Ontology (DO) are used to annotate the proteins and drug targets with additional information useful for a deeper interpretation of the biological processes and structures [8,17]. The DO database contains knowledge on 8,043 inherited, developmental and acquired human diseases. Through enrichment analysis, the R package DOSim is able to explore the biological meaning of related genes in terms of structure, function and hierarchy. The concepts in DOSim are organized into a directed acyclic graph (DAG) similar to a tree structure, the concepts are linked through various relationships. The lower the term or concept is positioned in the hierarchy then the more specific the term is, higher-up terms describe higher level or more abstract concepts. The ontologies are used to tag the association rules with biological meaning, in the sense they provide the medical users some indication of the pathways that are affected by the drugs, and how such pathways may lead to specific side-effects being presented by the patient. Incidentally, this kind of information is useful to drug companies as they are now desperate to reuse existing drugs (repurpose) for potentially very different diseases to the ones they were originally designed to treat [10,12,20].

The FDA (Food and Drug Administration) provide a freely available database called the adverse event reporting system (AERS). Online reporting of FDA AERS occurs on a quarterly basis, and began in January of 2004. Legacy records are available through the National Technical Information Service on compact disc (on a fee basis) or downloaded electronically. A quarterly AERS report contains several subsets of information:

- Demographics (DEMO); basic patient information.
- Drug types (DRUG); a list of drugs administered to patients.
- Indications (INDI); why they were given the drug initially.
- Outcomes (OUTC); the end result, e.g. hospitalisation, death.
- Reactions (REAC); side effect(s) experienced.
- Reporting sources (RPSR); where the information originated from.
- Therapy types (THER); how and when the drug was administered.

These seven subsets are linked using the primary key ID we can identify any patient with the drugs, side-effects and diseases they suffer from. During the course of a year a given patient may have more than one ADR and can appear several times. The data is all text based and stored in flat files which we saved in (CSV) format to enable access by our R programming environment.

As it pertains to the prediction of associations, all such reporting systems are inherently acute in nature; the information they contain has already been filtered based upon a close temporal association observed by those reporting to the databases. A prime example of how analysis of such databases may be lacking in predictive powers would be the development of cancers; a drug may induce the formation of a cancer, but the significant degree of latency between exposure and the formation of an observable symptom will likely be too great for some possible causal agents to be considered worth entering into the database.

2.3 Association Rule Mining

In a database, when considering the occurrence of one item with another in the same record (or transaction) represents an association. The frequency with which these items appear together overall in the database may represent some important relationship or trend. Several techniques are available such as the *Apriori* algorithm that can extract rules which highlight these occurrences and their frequencies. Formally we can define the following: where $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$ are a set of items. Let $\mathcal{D}$ be a collection of transactions in a database, where each transaction t has a unique identifier and contains a set of items such that $t \subseteq \mathcal{I}$. A set of items is called an *itemset*, and an itemset with k items is called a *k-itemset*.

A number of statistics are available to rank and order the association rules such as the *support*, $confidence$ and $lift$ of a rule. The *support* of an itemset x in $\mathcal{D}$, denoted as $\sigma(x/\mathcal{D})$, is the ratio of the number of transactions (in $\mathcal{D}$) containing x to the total number of transactions in $\mathcal{D}$.

An *association rule* is an expression $x \Rightarrow y$, where $x, y \subseteq \mathcal{I}$ and $x \cap y = \emptyset$. The *confidence* of $x \Rightarrow y$ is the ratio of $\sigma(x \cup y/\mathcal{D})$ to $\sigma(x/\mathcal{D})$. There are several association rule interestingness measures available should the number of extracted rules be large in number. The measures will select only those rules that have a certain statistical strength and confidence, and thus prune down the number rules to a manageable size.

Perhaps more helpfully we can say from a shopping database:

$$\{bread, cheese\}x \Rightarrow y\{butter\}$$

If a customer buys bread and cheese, then they are also likely to buy butter.

Lift is analogous to the relative response of patients and is of primary interest in identifying novel adverse events as this metric accounts for the high frequency of some consequences; e.g. "nausea" is a frequent consequence of many drugs combinations and if confidence alone were used to order associations this consequence, and others like, it would reduce the signal to noise ratio of the survey by appearing in every other record.

2.4 Related Work

The MOAL (Multi Ontology At All Levels) system of Manda *et al.* [9] is dedicated to extracting meaningful patterns and relationships from the Gene Ontology. When presented a gene product/disease, MOAL will generate association rules across all three sub-databases and use these to annotate the new gene products. Manda *et al.* have also developed their own interestingness metrics to evaluate and assess the discovered rules: *MOConfidence* and *MOSupport* derive the necessary information from a cross-ontology platform and select the most informative rules, thus pruning any superfluous information. [13] Uncovering disease-disease relationships through the incomplete human interactome. The DIseAse MOdule Detection (DIAMOnD) algorithm by Ghiassian used a systematic analysis of connectivity patterns of disease proteins in the human interactome [6]. Tatonetti analyzed drug interactions from adverse-event reports where they discovered interaction between paroxetine and pravastatin that increased blood glucose levels, thus warning doctors about this combination [19]. The work of Cai et al. is similar to ours [4], they also use association rules but frame these in the context of Bayesian networks in an attempt to explain causality. We use complex network theory to frame our rules.

3 Results

The drugs were first filtered to find the most frequent one hundred in recognition of fact that these have a propensity to occupy the majority of drug entries (approx 47% of the dataset) and that there is an apparent window of filtering that occurs around this region in which the ratio of side effect per drug entry reaches a maximum before again decreasing; the quantity of frequently occurring drugs filtered out can be easily adjusted within the code to examine the extents of the filters impact. These highly frequent drugs were then mapped back to the original drug list to mark patient entries containing them. The primaryid of the patients consuming the more frequent drugs was extracted and used as a master filter to remove all entries corresponding to those particular patient cases from both the drug and side effect lists. This ID always us to uniquely identity the patient and to link all known ADR's this person has suffered along with drugs they take, the diseases they suffer from. Since we are using one years worth of data (2016), we also miss the patient's history of previous ADR's and their drugs. Although this is not our objective in this work, we realize the importance of these databases to track trends in disease development over time, the drugs used and perhaps discarded through ADR's.

Rules are then generated in the form of an antecedent (left hand side, LHS) =>consequent (right hand side, RHS) relationship. Table 1 shows the initial setup for the association rule algorithm and the early set of results. Finally, the rules generated were filtered to remove those with less than ten observations, those with none unique side effects (as many similar drug combinations produce similar side effects) and sorted by lift criterion. The results were validated by comparing to Stockley's interaction checker available via British National Formulary.

In Fig. 2 the frequency of observations (drugs and side effects) per patient case after filtering off cases containing most frequent hundred drugs and subsequently cases containing the hundred most frequent side effects. The majority of patients

Table 1. Data mining parameters for Apriori algorithm on the FAERS dataset

Parameter	Value
Number of patients remaining in dataset	364,368
Number of drug and side effect observations in filtered dataset	19,790
Number of unique drug and side effect observations in filtered dataset	15,743
Apriori minimum support threshold	0.000005
Apriori minimum confidence	0.000001
Apriori minimum rule length	3
Apriori maximum rule length	3
Number of association rules initially generated	718,662
Number of association rules when constrained for drug antecedent and side effect consequent	778,820
Rules with at least ten observations present	368

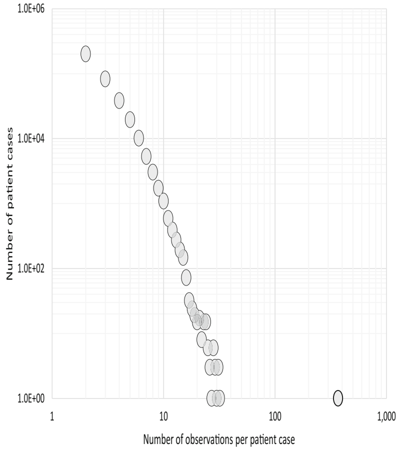


Fig. 2. Number of side-effects per patient case

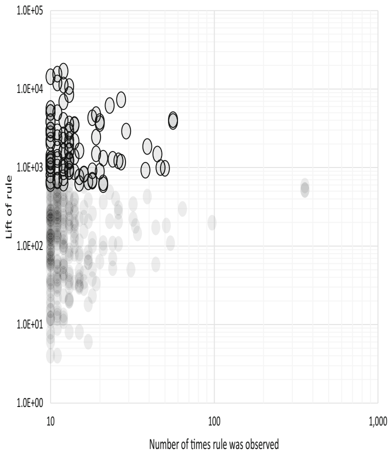


Fig. 3. Number of rules per patient case

have around ten or less observations in their records. 364,368 unique patients under examination, displaying 15,743 observations (drugs or side effects) at an array density of 0.02%.

Referring to Fig. 3 illustrates the distribution of rules found on the basis of their lift over the number of times they were observed. When data mining, it can be useful to have the axes include all possible values as it is one of the few occasions available to visually survey for anomalies in transformations; the datasets themselves are far too large to survey by eye. Rules that will be investigated further are encircled with a perimeter.

In Fig. 5 the highest lift associations made ten or more times. The rule will be stated along with its number of observations and lift. It will be proceeded by an explanation of the medical terminology, then any and all interactions listed by Stockley's and the rules predictability.

Rather than sift through the extracted rule-base, a health-care practitioner who is interested in a specific disease or indication and wishes to control how the rules are extracted they need to get the correct text name or UMLS code for

Table 2. Top scoring rules for all indications based on lift criteria

	Rules	Support	Confidence	lift
1	{Low density lipoprotein increased} => {Cardiovascular event prophylaxis}	0.20	82.64	397.15
2	{Cardiovascular event prophylaxis} => {Low density lipoprotein increased}	0.20	55.02	397.15
3	{Low density lipoprotein increased} => {Blood cholesterol increased}	0.18	76.80	90.20
4	{Blood cholesterol increased} => {Low density lipoprotein increased}	0.18	12.50	90.20
5	{Myelofibrosis} => {Product used for unknown indication}	0.20	46.14	1.36
6	{Cardiac failure} => {Product used for unknown indication}	0.18	40.84	1.21
7	{Epilepsy} => {Product used for unknown indication}	0.22	30.18	0.89
8	{Chronic myeloid leukaemia} => {Product used for unknown indication}	0.22	24.03	0.71
9	{Schizophrenia} => {Product used for unknown indication}	0.23	28.40	0.84
10	{Cardiovascular event prophylaxis} => {Blood cholesterol increased}	0.19	53.49	62.83
11	{Blood cholesterol increased} => {Cardiovascular event prophylaxis}	0.19	13.07	62.83
12	{Gait disturbance} => {Multiple sclerosis}	0.29	42.76	9.39
13	{Multiple sclerosis} => {Gait disturbance}	0.29	3.69	9.39
14	{Gait disturbance} => {Product used for unknown indication}	0.51	75.65	2.24
15	{Prostate cancer} => {Product used for unknown indication}	0.26	22.21	0.66
16	{Pulmonary hypertension} => {Product used for unknown indication}	0.36	34.84	1.03
17	{Bipolar disorder} => {Product used for unknown indication}	0.19	28.08	0.83
18	{Seizure} => {Product used for unknown indication}	0.22	29.41	0.87
19	{Deep vein thrombosis} => {Product used for unknown indication}	0.24	33.42	0.99
20	{Ankylosing spondylitis} => {Product used for unknown indication}	0.19	13.57	0.40

their query. The system at the moment is highly sensitive to case/spelling and words used. Future work will address these shortcomings and make the system more robust to user errors and differences in nomenclature.

This produced six rules with low support but high confidence, the lift criteria was substantive. These are shown in Table 3, the PU entry refers to *Product used for unknown indication* which is a frequently occurring item for the majority of drugs and implies that they are used for off-the-label diseases. Conversely, taking the problem the other way around what can association rules tell us about the comorbidities a patient can suffer from if they do develop Atrial fibrillation? Table 4 displays these rules.

The next stage was to calculate and integrate semantic similarity between our comorbid diseases identified by the association rules and to pull out any valid connections between them. Whilst the scope of the work described in this paper is beyond formal knowledge representation of any gene-disease associations using Gene Ontology (GO), we have Disease Ontology (DO) which provides a consistent description of gene products with disease perspectives, and is essential for supporting functional genomics in disease context (Table 2).

The comorbidities are identified using their local identifiers in DO. A number of measures can be used to rank semantic similarity, here we used the *Wang* criterion as it reflects the biological plausibility better than other measures because of the way semantic similarity of the DO terms are calculated, using both the locations of the terms in the DO graph and their relations with their ancestor terms [21].

$$Wang(A,B) = \frac{\sum\limits_{t \in T_A \cap T_B} S_A(t) + S_B(t)}{SV(A) + SV(B)} \tag{1}$$

Table 3. Comorbidities associated with Atrial Fibrillation generated six rules. It supports the question: What problems are patients likely to suffer from before developing Atrial Fibrillation? The left hand side (LHS) contains the antecedent and the right hand side (RHS) contains the consequent. Where: ∗PU (Product used for unknown indication)

	LHS	RHS	Support	Confidence	Lift
6	{Cerebrovascular accident prophylaxis,PU,Thrombosis prophylaxis}	{Atrial fibrillation}	0.00	0.99	47.30
3	{Cerebrovascular accident prophylaxis,Thrombosis prophylaxis}	{Atrial fibrillation}	0.01	0.98	47.03
5	{Cerebrovascular accident prophylaxis,PU}	{Atrial fibrillation}	0.01	0.80	38.31
4	{PU,Thrombosis prophylaxis}	{Atrial fibrillation}	0.00	0.77	36.83
2	{Cerebrovascular accident prophylaxis}	{Atrial fibrillation}	0.01	0.76	36.35
1	{Thrombosis prophylaxis}	{Atrial fibrillation}	0.01	0.63	30.04

Table 4. Comorbidities associated with Atrial Fibrillation generated four rules. It supports the question what problems will patients suffer from if they develop Atrial Fibrillation? This time the left hand side (LHS) contains the antecedent and the right hand side (RHS) contains the consequent. Where: ∗PU (Product used for unknown indication)

	LHS	RHS	Support	Confidence	Lift
2	{Atrial fibrillation}	{Cerebrovascular accident prophylaxis}	0.01	0.60	36.35
4	{Atrial fibrillation}	{Product used for unknown indication}	0.01	0.39	1.14
1	{Atrial fibrillation}	{Thrombosis prophylaxis}	0.01	0.37	30.04
3	{Atrial fibrillation}	{Hypertension}	0.00	0.11	3.18

For the Wang equation, where $SA(t)$ represents the S-value of DO term t related to term A and $SB(t)$ is the S-value of DO term t related to term B.

3.1 Ontology Integration

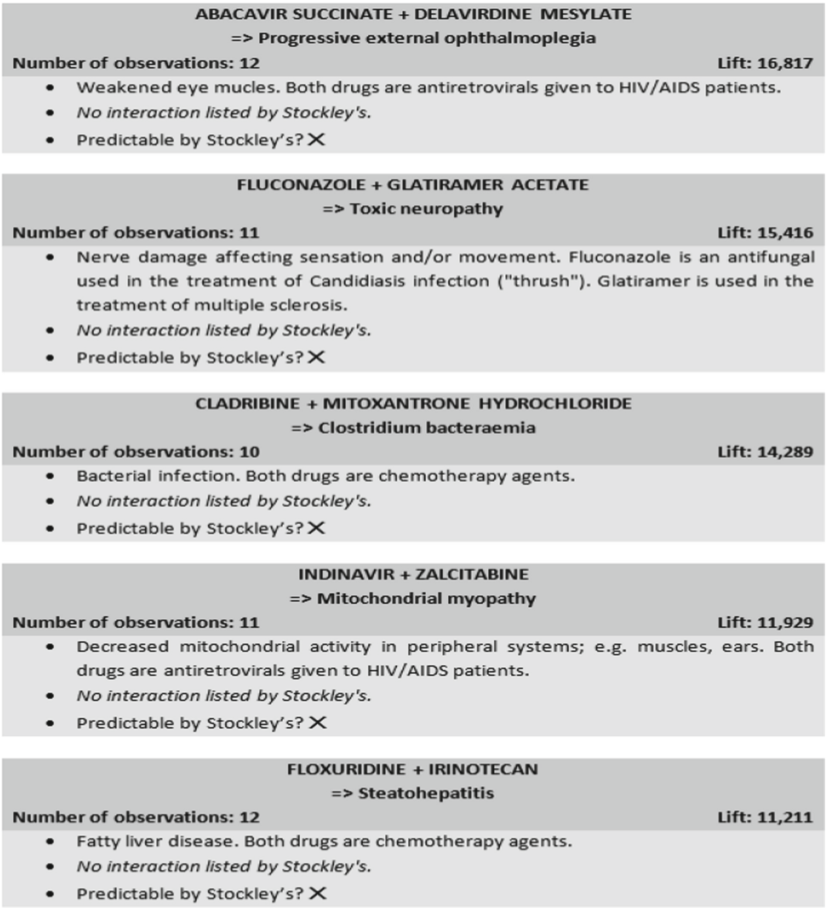
Individually, each method of data analysis provides important information in a specific area. However, further value of comes from the integration of these disparate sources of knowledge in a principled way. We use a variation of the Jaccard similarity coefficient to integrate the many sources of heterogenous information into a coherent entity for decision making.

$$Score = \text{disease}_{ij} + \text{drugs}_{ij} + \text{sideeffects}_{ij} + \text{DO-similarity}_{ij} + \text{GO-similarity}_{ij}$$

$$\text{Score index}_{ij} = \frac{|F(D_i) \cap F(D_j)|}{|F(D_i) \cup F(D_j)|}$$

where $F(D_j)$ are the features of interest, such as disease, side-effects, the drugs $F(D_i)$. The rules are then reevaluated using these scores and re-ranked. Implementing the equation produces a matrix with the diagonal containing the Jaccard score for the combination of association rule and the attached ontological terms.

The strongest correlation is between hypertension and cerebrovascular disease (.66), hardly a novel discovery but does reveal that this method is useful for integrating association rules with the semantic similarity of any disease.



ABACAVIR SUCCINATE + DELAVIRDINE MESYLATE
=> Progressive external ophthalmoplegia
Number of observations: 12 **Lift: 16,817**

- Weakened eye mucles. Both drugs are antiretrovirals given to HIV/AIDS patients.
- *No interaction listed by Stockley's.*
- Predictable by Stockley's? ✗

FLUCONAZOLE + GLATIRAMER ACETATE
=> Toxic neuropathy
Number of observations: 11 **Lift: 15,416**

- Nerve damage affecting sensation and/or movement. Fluconazole is an antifungal used in the treatment of Candidiasis infection ("thrush"). Glatiramer is used in the treatment of multiple sclerosis.
- *No interaction listed by Stockley's.*
- Predictable by Stockley's? ✗

CLADRIBINE + MITOXANTRONE HYDROCHLORIDE
=> Clostridium bacteraemia
Number of observations: 10 **Lift: 14,289**

- Bacterial infection. Both drugs are chemotherapy agents.
- *No interaction listed by Stockley's.*
- Predictable by Stockley's? ✗

INDINAVIR + ZALCITABINE
=> Mitochondrial myopathy
Number of observations: 11 **Lift: 11,929**

- Decreased mitochondrial activity in peripheral systems; e.g. muscles, ears. Both drugs are antiretrovirals given to HIV/AIDS patients.
- *No interaction listed by Stockley's.*
- Predictable by Stockley's? ✗

FLOXURIDINE + IRINOTECAN
=> Steatohepatitis
Number of observations: 12 **Lift: 11,211**

- Fatty liver disease. Both drugs are chemotherapy agents.
- *No interaction listed by Stockley's.*
- Predictable by Stockley's? ✗

Fig. 4. Top five ADR's ranked by lift criteria

4 Discussion

We processed and analyzed 162,744 patients entries, 24,641 unique drugs and 8,025 unique side effects were surveyed over four hours in a referenced study, using an approximately similar methodology, yielding 2,603 association rules at a minimum observation occurrence of 50. The results presented here began with 3,045,688 patients entries, 8,106 unique drugs, 16,248 unique side effects, was completed in approximately ten minutes and produced 78,820 association rules that were further refined to 368 at a minimum observation occurrence of just 2. An 18 fold increase in the number of patients that can be examined and 24 fold reduction in the time required has been achieved.

Some drug interactions were not found in Stockley's because the active ingredients used were not recognised; e.g. "HYPROMELLOSE 2910 (4000 MPA.S)"

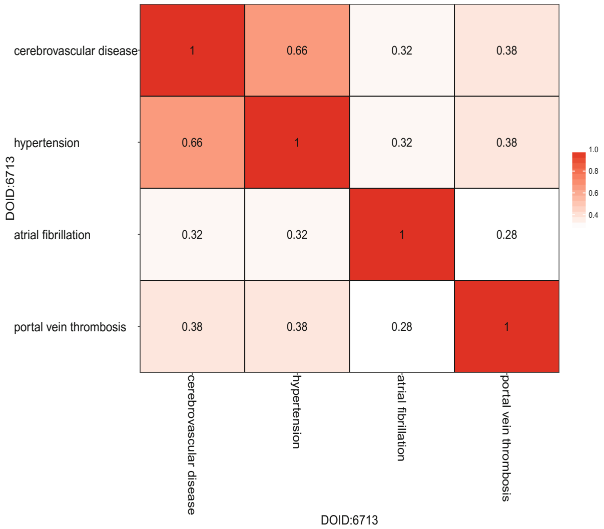


Fig. 5. Correlation matrix for the disease similarity

is not listed, but hypromellose is. Whilst no interactions were found for some drugs, the side effects for a drug alone could suggest the possible outcome. Alternatively, someone skilled or otherwise in the field of pharmacology could likely predict some of the outcomes based on medical experience. The approach on which this work was based employed the skills of a pharmacovigilance expert to assess the results for predictable interactions. Drug names were used verbatim here, and no attempt was made to further guess at the likelihood of a possible interaction outcome as, in practice, an algorithm lacking such worldly knowledge would be incapable of doing this without first having a reference for such predictable interactions, and it is the application of algorithms to medical records that is of interest.

There are interesting observations present; for instance, that the combination of anti-psychotics results in bleeding and bruising, even though these drugs are thought to be highly specific to neuronal functionality. Assurance that the methodology followed is able to detect rare events is exemplified by "ANENOCOUMAROL + LEVOFLOXACIN → International normalized ratio increased" and "EFAVIRENZ + ETONOGESTREL → Pregnancy with implant contraceptive"; both rare occurrences per Stockley's and high lift results where obtained.

In terms of the association rule generator (Apriori) we found that the candidate generation could be extremely slow based on the number of elements in LHS (pairs, triplets, etc.). Furthermore, the candidate generation process could generate duplicates depending on the implementation. The counting method iterates through all of the transactions each time.

During our analysis we found if a pathology is frequently encountered, such as heart attacks, a connection may not be drawn between the outcome and its possible causal agent being a particular drug or combination of them. As a result, it is likely that such chronic conditions are under-represented in the current works. Conversely, some entries may be over-reported as a result of media attention, accidental data entry, legal issues, a product being newer to the market.

The issues involved with the FAERS database have proven difficult to resolve, the manner in which the FDA list sequences of events for a particular patient's entry has changed over time. In its totality, the database is neither entirely structured or unstructured in form; information not only moves location within the database, with some fields being deleted, created or translocated, the associated field headers also change. Demographics such as gender was originally listed as *gndr-cod* and subsequently as *sex*. The same field, like others, has also moved location within the database.

5 Conclusions

There may be better approaches for finding less frequent (more novel) patterns, as apriori itself is fundamentally intended for finding frequent patterns. Huge quantities of the results apriori generates from medical records are just very common, very well known side effects; warfarin + aspirin → bleeding is a common discovery in the literature. However, when searching for more novel observations, it is more appropriate to reverse that, starting at the lower frequencies and working up towards the frequent rules. We achieved something roughly similar with the software we developed by first filtering off all the patients consuming frequent drugs and experiencing frequent side effects before running the algorithm; around half of the all the drug entries correspond to just 10 drugs in the FDA records. Future work will involve a more effective highlighting of unique drug combinations which may be achieved by initially filtering on a combinatorial basis instead; only excluding drugs that form frequent combinations. This would be most rapidly achieved using the first pass (FP-growth) hierarchical tree algorithm.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: SIGMOD 1993, pp. 207–216 (1993)
2. Ashburner, M.: Gene ontology: tool for the unification of biology. Nature Genet. **25**, 25–29 (2000)
3. Brin, S., Motwani, R., Silverstein, C.: Beyond market baskets: generalizing association rules to correlations. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 265–276 (1997)
4. Cai, R., Liu, M., Hu, Y., Melton, B.L., Matheny, M.E., Xu, H., Duan, L., Waitman, L.R.: Identification of adverse drug-drug interactions through causal association rule discovery from spontaneous adverse event reports. Artif. Intell. Med. **76**, 7–15 (2017). http://www.sciencedirect.com/science/article/pii/S0933365716305437

5. Dunn, N., Mann, R.: Prescription-event and other forms of epidemiological monitoring of side-effects in the UK. Clin. Exp. Allergy **29**(3), 217–239 (1999)
6. Ghiassian, S., Menche, J., Barabasi, A.: A DIseAse MOdule Detection (DIAMOnD) algorithm derived from a systematic analysis of connectivity patterns of disease proteins in the human interactome. PLoS Comput. Biol. **11**(4), e1004120 (2015)
7. Hahsler, M., Chelluboina, S., Hornik, K., Buchta, C.: The arules R-package ecosystem: analyzing interesting patterns from large transaction datasets. J. Mach. Learn. Res. **12**, 1977–1981 (2011)
8. Li, J., Gong, B., Chen, X., Liu, T., Wu, C., Zhang, F., Li, C., Li, X., Rao, S., Li, X.: Dosim: an R package for similarity between diseases based on disease ontology. BMC Bioinform. **12**(1), 266 (2011). http://www.biomedcentral.com/1471-2105/12/266
9. Manda, P., McCarthy, F., Bridges, S.: Interestingness measures and strategies for mining multi-ontology multi-level association rules from gene ontology annoations for the discovery of new GO relationships. J. Biomed. Inform. **46**(5), 849–856 (2013)
10. McGarry, K.: Discovery of functional protein groups by clustering community links and integration of ontological knowledge. Expert Syst. Appl. **40**(13), 5101–5112 (2013)
11. McGarry, K., Emery, K., Varnakulasingam, V., McDonald, S., Ashton, M.: Complex network based computational techniques for edgetic modelling of mutations implicated with human diseases. In: The 16th UK Workshop on Computational Intelligence, UKCI-2016, pp. 89–105. Springer-Verlag, University of Lancaster, UK (7th–9th September 2016)
12. McGarry, K., Slater, N., Amaning, A.: Identifying candidate drugs for repositioning by graph based modeling techniques based on drug side-effects. In: The 15th UK Workshop on Computational Intelligence, UKCI-2015. University of Exeter, UK (7th-9th September 2015)
13. Menche, J., Sharma, A., Kitsak, M., Ghiassian, S., Vidal, M., Loscalzo, J., Barabasi, A.: Uncovering disease-disease relationships through the incomplete human interactome. Science **347**(6224), 1257601 (2015)
14. Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., Kanehisa, M.: Kegg: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res. **27**, 29–34 (1998)
15. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2015). https://www.R-project.org/
16. Rodriguez, E., Staffa, J., Graham, D.: The role of databases in drug postmarketing surveillance. Pharmacoepidemiol. Drug Safety **10**(5), 407–410 (2001)
17. Schriml, L., Arze, C., Nadendla, S., Chang, Y.W., Mazaitis, M., Felix, V., Feng, G., Kibbe, W.: Disease ontology: a backbone for disease semantic integration. Nucleic Acids Res. **40**, D940–D946 (2012)
18. Tatonetti, N., Fernald, G., Altman, R.: A novel signal detection algorithm for identifying hidden drug-drug interactions in adverse event reports. J. Am. Med. Inform. Assoc. **19**(1), 79–85 (2012)
19. Tatonetti, N.P., Denny, J.C., Murphy, S.N., Fernald, G.H., Krishnan, G., Castro, V., Yue, P., Tsao, P.S., Kohane, I., Roden, D.M., Altman, R.B.: Detecting drug interactions from adverse-event reports: interaction between paroxetine and pravastatin increases blood glucose levels. Clin. Pharmacol. Ther. **90**, 133–42 (2011)
20. Wang, F., Zhang, P., Cao, N., Hu, J., Sorrentino, R.: Exploring the associations between drug side-effects and therapeutic indications. J. Biomed. Inform. **51**, 15–23 (2014)

21. Wang, J., Du, Z., Payattakool, R., Yu, P., Chen, C.: A new method to measure the semantic similarity of GO terms. Bioinformatics **23**(10), 1274–1281 (2007)
22. Wright, A., Chen, E., Maloney, F.: An automated technique for identifying associations between medications, laboratory results and problems. J. Biomed. Inform. **43**(6), 891–901 (2010)
23. Yang, J., Li, Z., Fan, X., Cheng, Y.: Drug disease association and drug-repositioning predictions in complex diseases using causal inference probabilistic matrix factorization. J. Chem. Inf. Model. **54**(9), 2562–2569 (2014)
24. Yu, G., Yan, G., He, Q.: DOSE: an R/Bioconductor package for disease ontology semantic and enrichement analysis. Bioinformatics **31**(4), 608–609 (2015)

Sentiment Analysis Model Based on Structure Attention Mechanism

Kai Lin[1,2,3], Dazhen Lin[1,2,3](✉), and Donglin Cao[1,2,3]

[1] Cognitive Science Department, Xiamen University, Xiamen, China
Linkai@stu.xmu.edu.cn, {dzlin,another}@xmu.edu.cn
[2] Fujian Key Laboratory of Brain-Inspired Computing Technique
and Applications, Xiamen University, Xiamen, China
[3] Fujian Provincial Key Laboratory of Information Processing
and Intelligent Control, Minjiang University, Fuzhou, China

Abstract. Since the long short-term memory (LSTM) network is a sequential structure, it is difficult to effectively represent the structural level information of the context. Sentiment analysis based on the original LSTM causes a problem of structural level information loss, and its capacity to capture the context information is finite. To address this problem, we proposed a novel structure-attention-based LSTM as a hierarchical structure model. It may capture relevant information in the context as much as possible. We propose HM (h_t matrix) to storage the structural information of the context. Furthermore, we introduce the attention mechanism to realize vector selection. Compared with the original LSTM and normal attention-based sentiment classification methods, our model obtains a higher classification precision. It is proved that the structure-attention-based method proposed in this study has an advantage in capturing the potential semantic structure.

Keywords: Structure-attention-based LSTM · Structural information · Vector selection

1 Introduction

In the era of Web 2.0, we can comment on current hot events, movies or goods. We can express our own preferences for these things via the social media. The sentiment analysis methods can obtain the sentiment tendencies of users in the commentary data. These sentiment tendencies have important research value and practical significance in the fields of public opinion analysis, financial analysis, advertising and so on.

The current technology for sentiment analysis is mainly based on traditional machine learning methods. Traditional SVM and Bayes method depend on sparse lexical features including bag-of-word (BoW) models and exquisitely designed patterns. Due to this, it ignores the relationship between words and barely captures the structure information in context. Even though current LSTM methods take context relationship into account, there remains sequential structure, which can hardly express hierarchical structural semantics. Our approach is to dynamically select the appropriate context semantic information in the recurrent network structure through the selection

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_2

mechanism based on LSTM. By our approach, we can get closer to the hierarchical syntax and semantic structure of tree structures. For example, in the sentence “For a person who was terrified of flying and could never get on a plane before, this was certainly a daunting task”, traditional methods might only notice the words “terrified”, “daunting”, but our model could analyze that there is a causal link in the long distance context.

We investigate the use of attention mechanism to automatically capture the most relevant words in the context to the recommendation task. The structure-attention-based LSTM proposed in this paper adopts serialization model in data modeling. By modeling the interactions between the words, our model can understand the context efficiently. In this model, with HM (h_t matrix) added, the most relevant feature vector is selected, which will be given a higher weight factor. This process strengthens the protection of important hierarchical information and it is favorable to sentiment classification.

2 Related Work

Since 2002, sentiment analysis has received a great improvement. In recent years, text sentiment analysis has been a popular research topic. During this period of time, machine learning and deep learning methods take the leading position.

2.1 Sentiment Classification with Traditional Machine Learning Methods

Most supervised sentiment classification methods are in close contact with the machine learning technology. They extract the appropriate features from the text to build classifiers. Naive Bayesian and support vector machine (SVM) are widely used.

Pang et al. [1] used SVM and Naive Bayesian to classify critic data into positive, negative. After the classification experiments, the effect of SVM classifier was found better with the precision of 83% achieved. In 2008, Pang [2] introduced bag-of-word (BoW) model in the original Bayesian model. Then he improved the classification performance to some extent. This model cannot understand semantic information of the text itself, so it is difficult to capture the deep semantic information of the context.

Hatzivassiloglou et al. [3] used the word sentiment as sentence-level features to analyze sentiment tendencies. Na et al. [4] analyzed specific words and negative phrases.

These machine learning methods do not take relationships between different words into account. They are significantly lacking in the characterization capabilities to represent the context.

2.2 Sentiment Classification with RNN and Attention Mechanism

Bengio et al. [5] proposed the neural network language model. The advantage of the neural network model in the study of emotion analysis is obvious. It can learn the character representation of phrases with different length and syntactic structure.

These feature representations can be used as classifier for the classification of phrases and sentences.

Traditional RNN has a good effect in obtaining text features. Unfortunately, it is hard for RNN to remember too much input information, so that long term dependency is a fatal problem in traditional RNN. Scholars add some gate functions to RNN [6]. It solves the problem of long distance dependency, but its capacity of characterization is still inadequate. It still tends to remember the closer information instead of the more relevant. There has been increasing research interest on LSTM [7].

LSTM has achieved a great success in various NLP tasks. Tang et al. added some structure to LSTM [8], which took target information into consideration, achieved state-of-the-art performance in target-dependent sentiment classification. Tang et al. obtained a target vector by averaging the vectors of words that the target phrase contains. However, simply averaging the word embedding of a target phrase is not sufficient to represent the semantics of the target phrase, resulting a suboptimal performance. Attention mechanism was exactly right introduced.

Attention mechanism was first used in the field of image. Google Mind team applied the [9] model in the field of image classification. Bahdanau et al. [10] made the first attempt to use an attention-based neural machine translation (NMT) approach to jointly translate and align words. The model is based on the basic encoder-decoder model. Differently, it encodes the input sentence into a sequence of vectors and chooses a subset of these vectors adaptively through the attention mechanism while generating the translation. Since then, the attention model based on RNN model expansions had been applied to a variety of NLP tasks.

Luong et al. [11] proposed two simple and effective attention mechanisms and achieved better results in handling long sentences. The model infers a variable-length alignment weight vector based on the current target state as well as all source states.

Kokkinos et al. [12] put forward structural attention model as a sequential model, which extracts informative nodes out of a syntactic tree and aggregates the representation of those nodes in order to form the sentence vector.

Despite the effectiveness of those methods, as the past models are nearly serialization models, it is still challenging to discriminate different sentiment polarities at a context level. Therefore, we are motivated to design a powerful neural network which can fully employ context information for sentiment classification.

3 Structure-Attention-Based LSTM

Original LSTM. A good language model should capture correct syntax at least. In order to do prediction that enjoys this property, we often need to consider a few preceding words. LSTM does consider word order, but it may lose the structural information due to long distance dependency. Original LSTM neural network is shown in Fig. 1. This simple RNN is a serialization model. It doesn't focus on the most contributing part of the text affective orientation. It has tendency to capture the nearer information. Thereby, it is easy to miss the decision proneness in the sentiment analysis process.

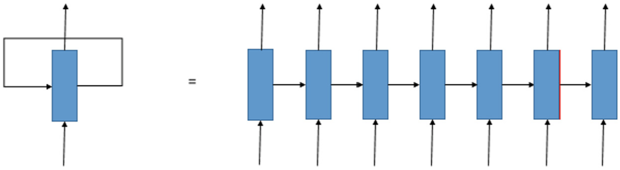


Fig. 1. Original LSTM structure

Attention-Based Model. Traditional network is able to capture the background information, but not able to distinguish which part is more important in the context. To solve this problem, attention is introduced to capture the emphasis of context information. For example, in sentence "great food but the service was dreadful", it is positive in "food", but negative in "service". In NLP field, attention is regarded as an automatic weighting mechanism that links different modules by weighting. As shown in Fig. 2, attention layer computes weight according to each LSTM unit. It includes a weight matrix. In this model, it has an attention layer to distribute weight, however, it is still a serialization model. As a serialization model, it can hardly learn the complete context information.

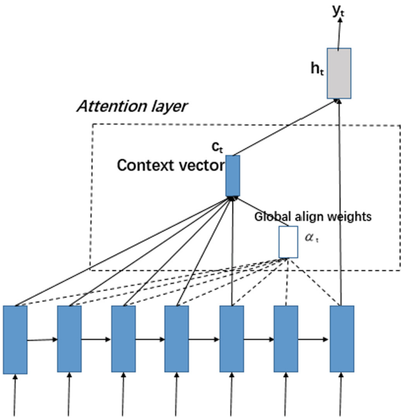


Fig. 2. Attention-based models

Proposed Method. The proposed structure-attention-based LSTM structure in this paper is shown in Fig. 3. On the basis of original LSTM and attention-based model, our model introduces a memory storage matrix HM (h_t matrix). At each LSTM unit, the output h_t of the unit is added to the HM. When a new vector is added, the HM is updated by specific operation. Then the HM vector is screened out by the attention mechanism.

Compared with LSTM and attention-based model, the advantages of our approach can be summarized in the following two aspects. (1) We construct a matrix, which is used as temporary context information storage. In virtue of the storage mechanism, our

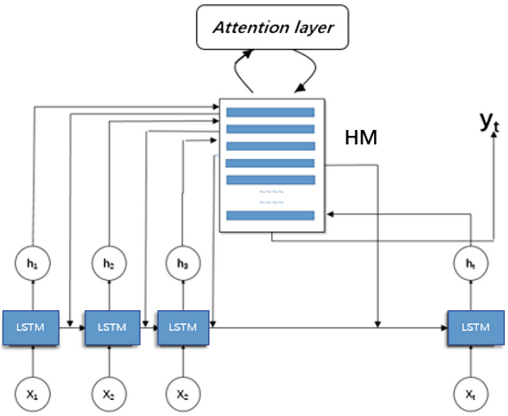


Fig. 3. Structure-attention-based LSTM neural network

model can obtain more complete context information than attention model. In view of the HM, this model is a hierarchical structure model, it has advantages in representing the context hierarchical information as well as learning context semantics. (2) The attention mechanism in our model works when each LSTM outputs. Therefore, each LSTM unit can refer to ample context information. While the previous attention mechanism works only when the last LSTM unit outputs, which cannot provide each LSTM unit with adequate context structural information.

Original LSTM is composed of the input gate i, output gate o and forget gate f and memory cell c. The input gate reads, output gate writes and forget gate learns to forget, shown in Eq. (1):

$$\begin{aligned}
\mathrm{i}_{\mathrm{t}} &= \sigma(W_i * [h_{t-1}, x_i] + b_i) \\
f_t &= \sigma(W_f * [h_{t-1}] + b_f) \\
\overline{C}_t &= \tanh(W_c * [h_{t-1}, x_t] + b_c) \\
C_t &= f_t \otimes C_{t-1} + i_t \otimes \overline{C}_t \\
o_t &= \sigma(W_o * [h_{t-1}, x_i] + b_o) \\
h_t &= o_t \otimes \tanh(C_t)
\end{aligned} \tag{1}$$

With structure attention mechanism added, the HM unit is updated as:

$$hm = attention([h_1; h_2; \ldots; h_{\mathrm{n}}], h_{\mathrm{t}}) = \sum_{i=1}^{t} (\alpha_{\mathrm{i}} h_i) \tag{2}$$

Taking all hidden states $[h_1, h_2, \ldots, h_n]$, the HM matrix outputs a continuous context vector $hm \in R^{\mathrm{dx1}}$ for each text. The output vector hm is computed as a weighted sum of each hidden state h_i.

$$
\begin{aligned}
\mathrm{i_t} &= \sigma(W_i * [hm, x_i] + b_i) \\
f_t &= \sigma(W_f * [hm] + b_f) \\
\overline{C}_t &= \tanh(W_c * [hm, x_t] + b_c) \\
C_t &= f_t \otimes C_{t-1} + i_t \otimes \overline{C}_t \\
o_t &= \sigma(W_o * [hm, x_i] + b_o) \\
h_t &= o_t \otimes \tanh(C_t)
\end{aligned}
\tag{3}
$$

where $\otimes$ stands for element-wise multiplication, σ is the sigmoid function, all $W \in R^{\mathrm{d} \times 1}$ and $U \in R^{\mathrm{d} \times \mathrm{d}}$ *are* weight matrices, all $b \in R^{\mathrm{d}}$ are bias vectors.

The output of LSTM layer is a sequence of hidden vectors [h_1, h_2,…, h_n]. Each annotation h_t contains information about the whole input text with a strong focus on the parts surrounding the t-th word of the input document.

The HM matrix vector structure is updated by Eq. (2), and the weight of the h_i vector is computed with the attention mechanism. The Original LSTM uses o_t as the input for the LSTM unit, but the proposed model uses the updated HM as the next input Weight factor α is defined as follows:

$$
\alpha_i = \frac{\exp(\beta_i)}{\sum_j^n \exp(\beta_j)} \tag{4}
$$

Where the weight scores β are calculated by using the target representation and the context word representation,

$$
\beta_i = U^T \tanh(W_1 * [h_i; h_t] + b_1) \tag{5}
$$

The parameter α is computed through Eqs. (2) and (4). New HM is computed as Eq. (2).

4 Experiment

4.1 Dataset

We evaluate the performance of the proposed structure-attention-based LSTM on Stanford critic IMDB dataset. This dataset consists of 9163 reviews on movie, classified as positive and negative polarity types. We divide it into training set, test set and validation set, as shown in Table 1. The data set is available for download at the following URL: http://ai.stanford.edu/~amaas/data/sentiment/.

Table 1. IMDB two-class experimental data set

Data set	Amount	Percent
Train set	6920	72%
Validation set	872	9%
Test set	1821	19%

4.2 Vector Dimension Selection

In the experiment, because, as the dimensions of the internal node influence the entire complexity of the model, it is necessary to select a suitable dimension to characterize vector internal nodes. In the task, we test different vector dimensions and record the precision of the structure-attention-based LSTM model, the experimental results are shown in Fig. 4.

Figure 4 shows different vector dimensions will affect the final classification results. When the vector dimension is 100, there will be an ideal result. When the dimension is less than 100, the precision is lower because the loss of information is likely to lead to poor classification results. If the dimension is 150 or greater, there will be many redundant dimensions.

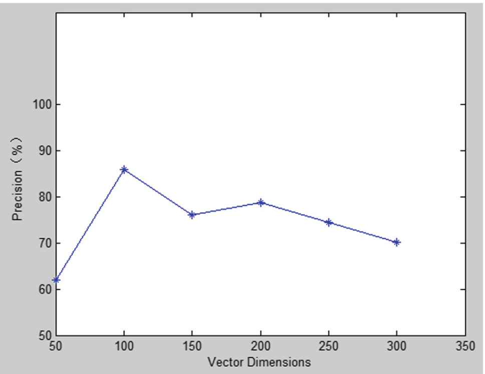


Fig. 4. The influence of vector dimension on classification accuracy

4.3 Results

Proposed structure-attention-based LSTM inherits most of the features from original LSTM. In order to validate our approach, we evaluate the performance on IMDB datasets. In this experiment, we use SVM, Naive Bayes, Original LSTM and Attention-based LSTM as a baseline. We respectively observe the detection results in positive sample data, negative sample data and on the whole. The results are presented in Table 2.

Table 2. Sentiment detection performance

Model	Pos	Neg	Precision
SVM	0.741	0.712	0.727
Naive bayes	0.840	0.811	0.821
Original LSTM	0.821	0.833	0.825
Attention-based LSTM	0.862	0.848	0.854
Structure-ATT LSTM	0.855	0.871	**0.863**

As can be seen from Table 2, we compare the results of our method with the existing discriminative and generative methods on the dataset. We have the following observations. (1) First of all, LSTM methods are superior to machine learning methods in this classification task. (2) Attention-based model performs significantly better than original LSTM. It demonstrates that attention model does better in capturing the correlation between words in the long distance dependency. (3) Structure-attention-based LSTM outperforms attention-based method on the whole. It shows that our model might have a better capacity in context characterization. (4) Proposed model presents superiority in negative sample data detection. In negative sample data, there are many obvious negative structural sentences, which means our model is extremely perceptive about negative structure like that in the sentence "main roles play cool that I don't enjoy". To prove this point, we have done the following work.

We analyze what feature can be much more easily learned with our model. Before this, we make a statistic analysis for negative sample data, which is presented in Table 3.

Table 3. Right results statistics for negative sample data

Model	Specific emotion words	Long distance dependency
Attention-based LSTM	498	188
Structure-ATT LSTM	502	204

As shown in Table 3, our approach performs as well as attention-based LSTM in detecting specific emotion words, but do much better when long distance dependency exists in the text. Further analysis is shown in Table 4.

Table 4. Negative sample data examples

Sample data	Classification result	Analysis
Alas, another Costner movie that was an hour too long. Credible performances, but the script had nowhere to go and was in no hurry to get there. Firstly, we are offered an unrelated string of events few of which further the story. Will the script center on Randall and his wife? Randall and Fischer? How about Fischer and Thomas? In the end, no real front story ever develops and the characters themselves are artificially propped up by monologues from third parties	Ground Truth: Negative Structure-ATT LSTM: Negative Attention-based LSTM: Positive	No obvious emotion word is in this example Our model detects the structural information, paying attention to phrase "too long", "but", "in no hurry", "no real front story", nearly every part of the text While attention-based model only notices "in the end", leading to a fuzzy evaluation
Once again Mr. Costner has dragged out a movie for far longer than necessary. Aside from the terrific sea rescue sequences, of which there are very few I just did not care about any of the characters. Most of us have ghosts in the closet, and Costner's character are realized early on, and then forgotten until much later, by which time I did not care	Ground Truth: Negative Structure-ATT LSTM: Negative Attention-based LSTM: Positive	Our model pays attention to "far longer", "did not care about", "did not care" While attention-based model detects "care about" and ignore "did not"

In these examples given in Table 4, there exists even no specific emotion word, but long distance dependency. Attention-based LSTM can hardly handle this situation exactly but our model could. We can conclude that our model has superiority in learning context information of long distance dependency.

As for the positive sample data, the accuracy of our approach is lower than that of attention-based model. We find out some examples for analysis. Representative examples are given in Table 5.

Table 5. Positive sample data examples

Sample data	Classification result	Analysis
My boyfriend and I went to watch The Guardian. At first I didn't want to watch it, but I loved the movie- It was definitely the best movie I have seen in sometime. They portrayed the USCG very well, it really showed me what they do and I think they should really be appreciated more. Not only did it teach but it was a really good movie. The movie shows what the really do and how hard the job is	Ground Truth: Positive Structure-ATT LSTM: Negative Attention-based LSTM: Positive	Our model finds the phrase "did not want", finally judges it as negative. It means our model is sensitive to negative structure
This film has its detractors, and Courtney's fey dresser may offend some folks (who, frankly, need a good smack upside the head) – but the film is top notch in every way: engaging, poignant, relevant. Finney, naturally, is larger than life	Ground Truth: Positive Structure-ATT LSTM: Negative Attention-based LSTM: Positive	"Detractor" is marked in our model, but the sentence itself contains an adversative relation. Our model doesn't perform well in this situation

The examples given in Table 5 are positive but detected as negative with our model. It indicates in our model, further information may also get a higher weight, which causes wrong classification while confronting adversative relation sentences. As mentioned, our model performs better in negative sample data.

4.4 Discussion

In the structure-attention-based model, we introduce attention mechanism to update the HM memory matrix structure. HM, which can be regarded as a memory bank, stores full text vectors. In the process of sentiment analysis, there are often close contextual connections. For example, the modification of words or the transition relations in longer text. When dealing with such text analysis, simple LSTM and attention mechanisms perform linearly.

In Figs. 5, 6 and 7, we use different colors to indicate which block the model focus more. Blue means the model focuses less. Green refers to what the model pays more attention to, but not much enough. While red represents the most contributing part in the model.

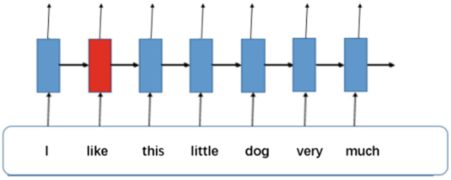


Fig. 5. Original LSTM example

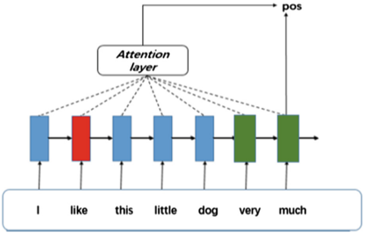


Fig. 6. Attention-based model

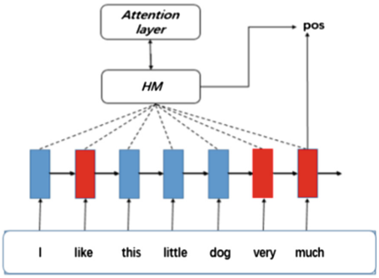


Fig. 7. Example for sentiment analysis with structure-attention-based model

As shown in Fig. 5, in the sentence "I like this little dog very much", original LSTM can only seize the sentiment word "like". But it is even not sensitive to the sentiment modifiers due to the long distance dependency. Visible LSTM cannot understand the context hierarchy. It can hardly grasp syntactic structural information.

In Fig. 6, simple attention-based model can analyze the text sentiment, but it lacks close contact between the sentiment word "like" and the adverb phrase "very much". Attention-based model has a problem that its acquisition ability to capture simple syntactic structural information is not robust.

But in the proposed structure-attention-based model, the hierarchical structural information is retained, so the modifiers can be analyzed accurately, as shown in Fig. 7.

5 Conclusion and Outlook

In this work, we proposed a novel hierarchical structure LSTM to capture the potential semantic structure. The added storage matrix filters information through the attention mechanism. It helps optimize memory in previous models. Experiments show that, compared to ordinary LSTM models, sentiment classification precision increased by 3.5%.

But as far as the method is concerned, there still remains much room for improvement. The future research can proceed from the following two aspects:

1. To generalize a complete encoder-decoder model, we can continue to expand the model by adding a decoder. By this, we can learn contextual information more completely.
2. The amount of experimental data can be further expanded. With the IMDB 100 k sentiment classification data sets, larger training set may be able to get better experimental results.

Acknowledgement. This work is supported by the Nature Science Foundation of China (No. 61402386, No. 61305061, No. 61502105, No. 61572409, No. 81230087 and No. 61571188), Open Fund Project of Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University) (No. MJUKF201743), and Education and scientific research projects of young and middle-aged teachers in Fujian Province under Grand No. JA15075. Fujian Province 2011 Collaborative Innovation Center of TCM Health Management and Collaborative Innovation Center of Chinese Oolong Tea Industry — Collaborative Innovation Center (2011) of Fujian Province.

References

1. Pang, B., Lee, L., Vaithyanathan, S.: Sentiment classification using machine learning techniques. In: Conference on Empirical Methods in Natural Language Processing, vol. 10, pp. 79–86. ACM, Stroudsburg, PA, USA (2002)
2. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retr. **2**(1–2), 1–135 (2008)
3. Yu, H., Hatzivassiloglou, V.: Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In: EMNLP, pp. 129–136 (2003)
4. Khoo, H., Zhou, Na, J.C., et al.: Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews. In: ISKO, pp. 49–54. Ergon Verlag, Wurzburg, Germany (2004)
5. Bengio, Y., Ducharme, R., Vincent, P., et al.: A neural probabilistic language model. J. Mach. Learn. Res. **3**, 1137–1155 (2003)
6. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with LSTM. In: 9th International Conference on IET, pp. 2002–2451 (1999)
7. Stollenga, M.F., Byeon, W., Liwicki, M., et al.: Parallel multi-dimensional LSTM, with application to fast biomedical volumetric image segmentation. Comput. Sci. (2015)
8. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: EMNLP, pp. 1422–1432 (2005)
9. Mnih, V., Heess, N., Graves, A.: Recurrent models of visual attention. In: Advances in Neural Information Processing Systems, 2014, pp. 2204–2212 (2014)
10. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate. In: ICLR 2015, pp. 1–15 (2015)
11. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. Comput. Sci. (2015)
12. Kokkinos, F., Potamianos, A.: Structural attention neural networks for improved sentiment analysis (2017)

Fuzzy Representation for Flexible Requirement Satisfaction

Ratih N.E. Anggraini[1,2(✉)] and T.P. Martin[1]

[1] Intelligent Systems Lab, University of Bristol, Bristol BS8 1UB, UK
ra16032@bristol.ac.uk

[2] Informatics Department, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Abstract. The need for adaptive systems is growing with the increasing number of autonomous entities such as software systems and robots. A key characteristic of adaptive systems is that their environment changes, possibly in ways that were not envisaged at design-time. These changes in requirements, model and context mean the functional behaviour of a system cannot be fully defined in many cases, and consequently formal verification of the system is not possible. In this research, we propose a fuzzy representation to describe the result of requirement verification. We use an adaptive assisted living system as the case study. The RELAX language is used to create a flexible system specification. We model and simulate the system using UPPAAL 4 and use a fuzzy approach to translate the simulation result into fuzzy requirement satisfaction. The result shows the benefit of a more flexible representation by describing the degree of requirement satisfaction rather than a strict yes/no Boolean judgment.

Keywords: Fuzzy · Requirement satisfaction · Adaptive system

1 Introduction

The trend towards intelligent and autonomous systems (such as assistive robots, autonomous vehicles, network controllers and other intelligent software agents) has led to a need to create software that is increasingly adaptive. An adaptive system is defined as one that can alter its behaviour to suit changes in its environment [1] such as sensor failures, human factors, or network condition. The adaptation subsequently modifies requirements, models, and context making it harder to verify, particularly when changes may have not have been fully anticipated by designers.

RELAX is a requirements engineering language for adaptive systems, able to capture the uncertainty in adaptive system requirements [2] so that the verification process can become easier. RELAX was combined with SysML/KAOS which is a goal oriented requirement engineering to model self-adaptive system in UML modelling language [3]. Unfortunately, RELAX is only a requirement language to describe system specification and is unable to assess requirement satisfaction.

Verification is a way to prove or to disprove requirement satisfaction in a system [4]. The classic way of describing requirement satisfaction is using Boolean values, i.e. a yes or no answer. Using this approach, we only know whether a requirement is

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_3

satisfied or not - we are unable to know that a requirement was almost always satisfied, or that it was almost satisfied in all cases.

Thus, in this research we utilize a fuzzy approach to describe the degree to which a requirement is satisfied, when the requirement is specified in RELAX. We used UPPAAL [5] to model and simulate the system. The simulation result is translated into a fuzzy representation showing the degree to which the requirement is satisfied. The application is adapted from the RELAX description [2] and is intended to demonstrate the principles of our approach rather than to be a fully realistic system.

2 Background

2.1 Fuzzy Sets

Classical set theory uses Boolean truth values (0 for false and 1 for true) to represent set membership. On the other hand, as noted by Zadeh [6], many concepts used in natural language are loosely defined and admit elements according to a scale of membership rather than according to an absolute yes/no test. Since requirements are initially expressed in natural language, and concepts can be regarded as labels for sets of entities, it makes sense to use a fuzzy representation for flexible requirements. The key idea of a fuzzy set is that its elements are members of a set to some degree, and that a specific element can belong to a greater (or lesser) degree than another element. This is obvious when the membership is based on a numerical measurement (e.g. *tall* people or *expensive* restaurants) and is equally valid in more complex concepts such as *socially-responsible company,* or *reliable software.* Fuzzy set theory, in its simplest form, expresses set membership as a real number between 0 and 1. By using a fuzzy approach, we can represent requirements in a more flexible way [6].

For example, consider a set of coffee drinks. Instead of assessing the sweetness in a crisp format such as sweet or bitter, a fuzzy approach describes it in a more flexible way such as sweet, rather sweet, rather bitter (or slightly sweet) and bitter (not sweet). This flexible representation is more commonly used in human language.

A fuzzy set can be represented by a membership function which maps elements in the universe U to a membership value between 0 and 1 as shown in Eq. 1.

$$f{:}U \rightarrow [0,1] \tag{1}$$

For the coffee drinks example, we may measure sweetness based on how many spoons of sugar are added. Let us say that 3 spoons are sweet and no sugar added is bitter. Then we can represent coffee sweetness by a fuzzy membership as shown in Table 1.

Table 1. Representation of coffee sweetness membership function f(x)

Spoon of sugar x	Fuzzy value $f(x)$
3	1
2	0.67
1	0.33
0	0

$$U\text{: } \{0, 1, 2, 3\}$$
$$f{:}U \rightarrow [0,1]$$
$$f(x) = x/3$$

The *X-μ* representation of fuzzy sets [7] focuses on the degree of membership and views a fuzzy set as a collection of objects with a loosely defined boundary. At different membership grades, we have different sets. Thus, if asked to define *sweet coffee*, we would accept coffee containing 3 spoons of sugar as a definite member. By relaxing the definition slightly (i.e. lowering the membership threshold) we would accept coffee with 2 or 3 spoons; by relaxing it even more, we might accept 1, 2, or 3 spoons of sugar. The strength of *X-μ* lies in its focus on crisp sets, which can be processed by standard methods; these sets vary according to the membership threshold.

2.2 RELAX Requirement Language

A dynamically adaptive system (DAS) can have large uncertainty due to continuous change of the system by the modification of its environment. Subsequently, satisfying the system requirement becomes more challenging and hence, a tolerance of requirement satisfaction is necessary. To facilitate this toleration, the requirement language called RELAX was proposed [2].

RELAX allows requirement specifications to be written in a structured natural language. Requirements are written using the SHALL operator. For non-invariant requirements, the SHALL operator will be followed by a temporal operator to handle its flexibility. The RELAX grammar is shown below (see formulae (2)).

$$\begin{aligned}
&\phi ::= true|false|P|SHALL\ \phi|\mathrm{MAY}\ \phi_1\ \mathrm{OR\ MAY}\ \phi_2\\
&|EVENTUALLY\ \phi|\phi_1\mathrm{UNTIL}\ \phi_2|BEFORE\ e\ \phi|\\
&\mathrm{AFTER\ e}\ \phi|\mathrm{IN\ t}\ \phi|\mathrm{AS\ CLOSE\ AS\ \ POSSIBLE\ f}\ \phi|\\
&\mathrm{AS}\ \{\mathrm{EARLY, LATE, MANY, FEW}\}\ \mathrm{AS\ POSSIBLE}\ \phi
\end{aligned} \tag{2}$$

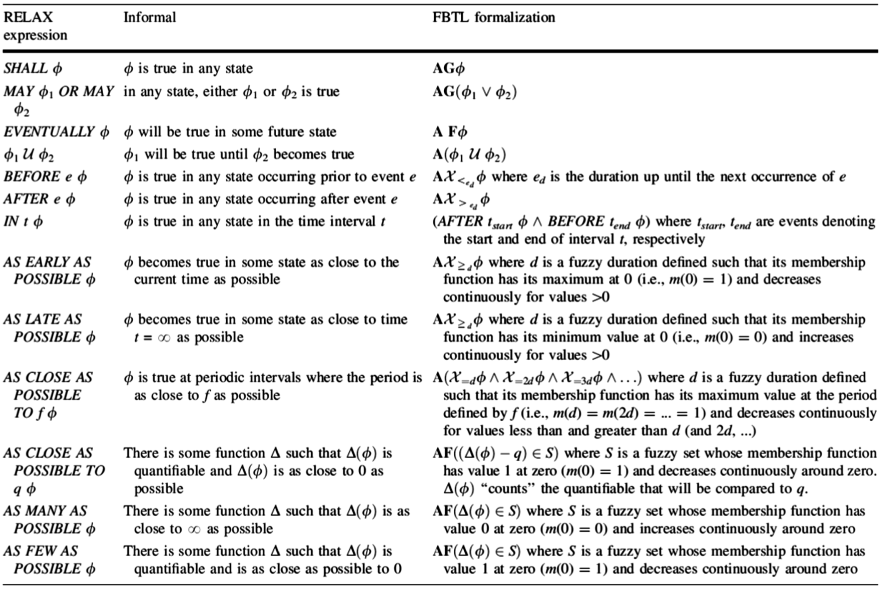
RELAX semantics are expressed using fuzzy branching temporal logic (FBTL) [8]. Moreover, FBTL describes requirement satisfaction using a standard fuzzy membership scale of real numbers between [0, 1] instead of saying the requirements are satisfied or not. The semantics of RELAX are shown in Fig. 1.

3 Fuzzy for Requirement Satisfaction Representation

To show how we can represent fuzzy requirement satisfaction, we use the example given in [2], of an adaptive assisted living system, focusing on R1.1 and R1.2. The RELAX requirements we use are shown in Table 2. UPPAAL 4 was used to model and simulate the system [9].

Table 2. AAL requirements used as example

R1.1	The fridge SHALL detect and communicate information with AS MANY food packages AS POSSIBLE
R1.2	The fridge SHALL suggest a dietplan with total calories AS CLOSE AS POSSIBLE TO the daily ideal calories



RELAX expression	Informal	FBTL formalization
SHALL ϕ	ϕ is true in any state	$\mathbf{AG}\phi$
MAY ϕ_1 *OR MAY* ϕ_2	in any state, either ϕ_1 or ϕ_2 is true	$\mathbf{AG}(\phi_1 \vee \phi_2)$
EVENTUALLY ϕ	ϕ will be true in some future state	$\mathbf{A\ F}\phi$
$\phi_1\, \mathcal{U}\, \phi_2$	ϕ_1 will be true until ϕ_2 becomes true	$\mathbf{A}(\phi_1\, \mathcal{U}\, \phi_2)$
BEFORE $e\ \phi$	ϕ is true in any state occurring prior to event e	$\mathbf{A}\mathcal{X}_{<_{e_d}}\phi$ where e_d is the duration up until the next occurrence of e
AFTER $e\ \phi$	ϕ is true in any state occurring after event e	$\mathbf{A}\mathcal{X}_{>_{e_d}}\phi$
IN $t\ \phi$	ϕ is true in any state in the time interval t	(*AFTER* $t_{start}\ \phi \wedge$ *BEFORE* $t_{end}\ \phi$) where t_{start}, t_{end} are events denoting the start and end of interval t, respectively
AS EARLY AS POSSIBLE ϕ	ϕ becomes true in some state as close to the current time as possible	$\mathbf{A}\mathcal{X}_{\geq_d}\phi$ where d is a fuzzy duration defined such that its membership function has its maximum at 0 (i.e., $m(0) = 1$) and decreases continuously for values >0
AS LATE AS POSSIBLE ϕ	ϕ becomes true in some state as close to time $t = \infty$ as possible	$\mathbf{A}\mathcal{X}_{\geq_d}\phi$ where d is a fuzzy duration defined such that its membership function has its minimum value at 0 (i.e., $m(0) = 0$) and increases continuously for values >0
AS CLOSE AS POSSIBLE TO $f\ \phi$	ϕ is true at periodic intervals where the period is as close to f as possible	$\mathbf{A}(\mathcal{X}_{=d}\phi \wedge \mathcal{X}_{=2d}\phi \wedge \mathcal{X}_{=3d}\phi \wedge \ldots)$ where d is a fuzzy duration defined such that its membership function has its maximum value at the period defined by f (i.e., $m(d) = m(2d) = ... = 1$) and decreases continuously for values less than and greater than d (and $2d$, ...)
AS CLOSE AS POSSIBLE TO $q\ \phi$	There is some function Δ such that $\Delta(\phi)$ is quantifiable and $\Delta(\phi)$ is as close to 0 as possible	$\mathbf{AF}((\Delta(\phi) - q) \in S)$ where S is a fuzzy set whose membership function has value 1 at zero ($m(0) = 1$) and decreases continuously around zero. $\Delta(\phi)$ "counts" the quantifiable that will be compared to q.
AS MANY AS POSSIBLE ϕ	There is some function Δ such that $\Delta(\phi)$ is as close to ∞ as possible	$\mathbf{AF}(\Delta(\phi) \in S)$ where S is a fuzzy set whose membership function has value 0 at zero ($m(0) = 0$) and increases continuously around zero
AS FEW AS POSSIBLE ϕ	There is some function Δ such that $\Delta(\phi)$ is quantifiable and is as close as possible to 0	$\mathbf{AF}(\Delta(\phi) \in S)$ where S is a fuzzy set whose membership function has value 1 at zero ($m(0) = 1$) and decreases continuously around zero

Fig. 1. Semantics of RELAX expressions [2]

The model of the food information detection sub system is shown in Fig. 1. A sensor in the fridge will detect food information once a day. We used probability 1:9 to model unforeseen situations that make the food information sensor unable to gather the information on all packages. This is represented in the subsystem diagram by the dotted lines labelled 9 and 1. The success of the system depends on how many food packages are not detected. We use a variable *r* in the model to represent the number of undetected food packages, to be relaxed in accordance with the requirement "as many as possible". The system does not meet the requirement if it fails to detect food packages *r* or more times, otherwise it meets this requirement. RELAX-ing R1.1 allows us to incorporate flexibility, which means that this system requirement is considered satisfied to a degree even though a certain number of food packages are not detected. The fuzzy membership function *f(x)* is used to represent the fuzzy requirement (see Eq. (3)), where *r* is the (relaxed) threshold value and Δx is the number of undetected food packages. The X-μ interpretation is that at membership 1, the system must detect all 10 food packages to satisfy "as many as possible", at membership 0.75, either 9 or 10 packages must be detected, etc.

$$f(\Delta x) = \begin{cases} if\ \Delta x \leq r, & \frac{(r\ +\ 1) - \Delta x}{r\ +\ 1} \\ else, & 0 \end{cases} \tag{3}$$

In this simple case, the process can also be treated analytically. Figure 3 shows the relationship between the degree of relaxation and the probability that the system detects *all* food packages for a range of detection probabilities. For the fully relaxed definition of *as many as possible* (i.e. at least 7 out of 10 packages), a sensor success rate of 0.9 or 0.98 will almost always detect *"all"* food packages (Fig. 2).

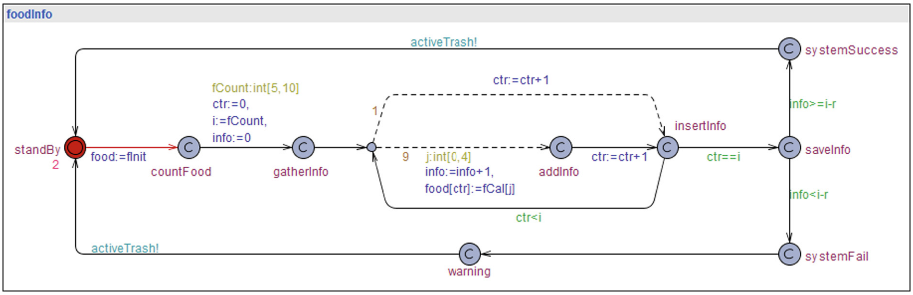


Fig. 2. UPPAAL model to detect food information

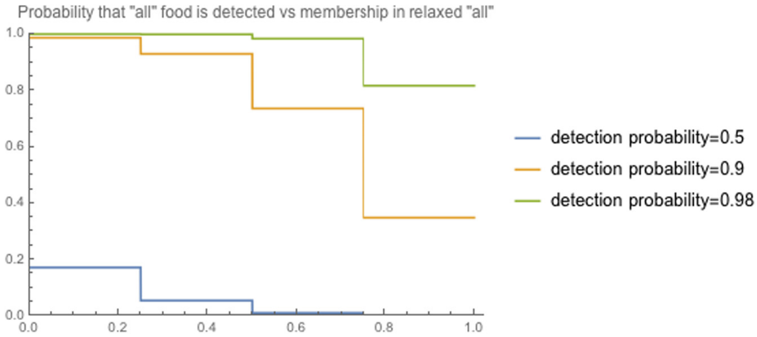


Fig. 3. Probability of detecting *all* food packages vs relaxation of "*all*" definition. The *y*-axis shows the probability, the *x*-axis shows the degree of relaxation in the definition of *all*

In general, we use extended simulation of the system to determine the degree to which the system satisfies the relaxed requirements. Figure 4 shows the simulation result of the system model to gather food information. The system performs this task daily. The blue line represents the real food in the fridge and the red line shows the food info that was successfully detected. Moreover, the green one is the absolute minimum food information that should be detected by the system in order for the system to be considered as meeting requirements. In this case, the number of undetected packages is RELAX-ed by 3, so it is considered as a success (to some degree) if the number of undetected food packages is 3 or less. From the chart, we can see that the simulated system works well except at day 32. It means that out of 40 days, the system fails only once in gathering minimum food information. Figure 5 shows the (fuzzy) degree to which the requirement is met during 40 days' simulation.

The model of daily calorie intake in Fig. 6 assumes that Mary[1] has three meals and two snacks time a day where each meal is suggested to be 400 calories and snack 200 calories. The next meal calorie suggestion will be calculated based on how many calories have been taken up to the current meal. If the system detects that Mary has not taken her meal, it will activate the alarm (3 times). At the end of the day, the system computes total calorie intake and calorie deviation to 1600 calories. The value of 1600 calories is based on calorie calculator[2] which is suggested the total calorie for 65-year-old female, overweight and sedentary activity. The system will be relaxed by ±r calories and will send a warning to the care-giver or other parties if the diet fails.

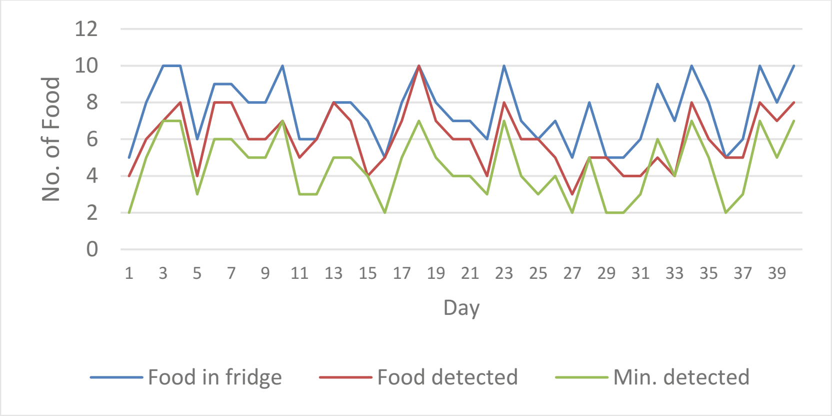


Fig. 4. Simulation result on food information detection

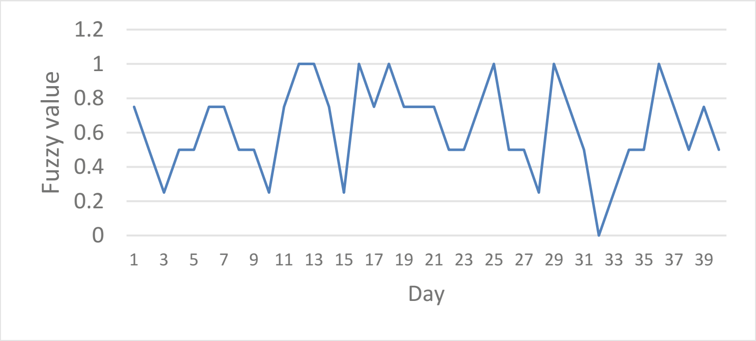


Fig. 5. Fuzzy requirement satisfaction of food detection for 40 days' simulation

[1] Mary is the subject of the assisted living system described in the RELAX paper.

[2] http://www.healthycalculators.com/calories-intake-requirement.php.

The simulation result of calorie intake is shown in Fig. 7. The blue line is the daily calorie intake, whilst the green and red lines are the maximum and minimum relaxed calorie intake, respectively. The graph shows rough rises and falls because the decision on taking meals depends on Mary herself. The system can only remind her (activate alarm) and gives warnings if Mary fails to follow the ideal daily intake as shown in the model (see Fig. 6).

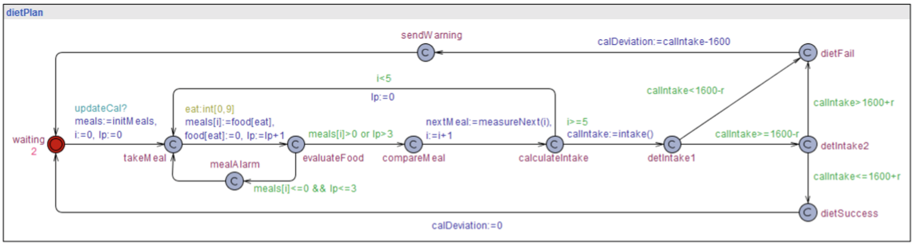


Fig. 6. UPPAAL model to monitor calorie intake

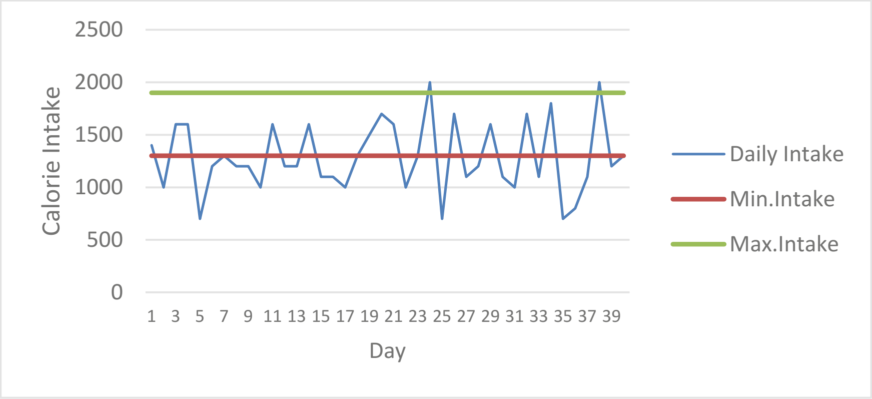


Fig. 7. The simulation result on daily calorie intake

Equation (4) is the membership function *f(x)* used to convert the value of daily calorie intake into fuzzy, where *r* is relaxing value of calorie deviation and Δ*x* is the actual calorie deviation indicating the difference of ideal daily calorie intake to real consumption. As for the fuzzy requirement satisfaction of calorie intake is described in Fig. 8.

$$f(\Delta x) = \begin{cases} -r \le \Delta x \le r, \frac{(r+1)-|\Delta x|}{(r+1)} \\ else, 0 \end{cases} \tag{4}$$

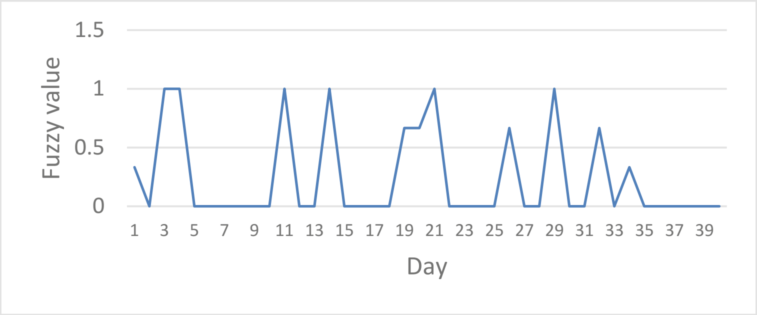


Fig. 8. Fuzzy satisfaction of calorie intake.

Figures 5 and 8 describe the satisfaction of RELAX requirement. Fuzzy value 1 indicates that the requirement is fully satisfied and 0 means it is unsatisfied, and values in between indicate that the requirement is satisfied to some degree.

4 Conclusion

This paper introduces a new way of representing requirement satisfaction using a fuzzy model of flexible requirements. We have modelled a simple system to illustrate the underlying ideas, with the requirement specification written in the RELAX language and UPPAAL 4 used to model and simulate the system. The fuzzy approach has more flexibility than the classic crisp representation so we can describe the degree to which the requirement is satisfied. Simulations are included to illustrate the principles of the approach, and additional analysis will be undertaken to investigate the sensitivity and requirements for reliable simulation.

Future work will examine the theoretical aspects of this approach in greater detail, and develop a fully integrated approach to modelling and refining flexible requirements so that we can verify that a system satisfies the requirements to some degree.

References

1. Tamura, G., Villegas, N., et al.: Towards practical runtime verification and validation of self-adaptive software systems. In: de Lemos, R., et al., (eds.) Software Engineering for Self-adaptive Systems II, Revised Selected and Invited Papers, Dagstuhl Castle, Germany, 24–29 October 2010, pp. 108–132. Springer, Heidelberg (2013)
2. Whittle, J., Sawyer, P., et al.: RELAX: a language to address uncertainty in self-adaptive systems requirement. Requir. Eng. **15**, 177–196 (2010). ISSN 1432-010X
3. Ahmad, M., Belloir, N., Bruel, J.-M.: Modeling and verification of functional and non-functional requirements of ambient self-adaptive systems. J. Syst. Softw. **107**, 50–70 (2015)
4. Systems and software engineering – Vocabulary. ISO/IEC/IEEE 24765:2010(E), pp. 1–418. (2010). http://www.uppaal.org/

5. UPPAAL website (2017). http://www.uppaal.org/
6. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965)
7. Martin, T.P.: The X-μ representation of fuzzy sets. Soft. Comput. **19**(6), 1497–1509 (2015)
8. Moon, S., Lee, K.H., Lee, D.: Fuzzy branching temporal logic. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **34**(2), 1045–1055 (2004)
9. David, A., et al.: Uppaal SMC tutorial. Int. J. Softw. Tools Technol. Transf. **17**(4), 397–415 (2015)

A Multidisciplinary Method for Constructing and Validating Word Similarity Datasets

Yu Wan[1], Yidong Chen[1(✉)], Xiaodong Shi[1], Guorong Cai[2], and Libai Cai[3]

[1] Department of Cognitive Science, School of Information and Engineering, Xiamen University, Xiamen 361005, Fujian, People's Republic of China
ydchen@xmu.edu.cn

[2] State Grid Fujian Liancheng Electric Power Company Limited, Longyan 366200, Fujian, People's Republic of China

[3] Computer Engineering College, Jimei University, Xiamen 361005, Fujian, People's Republic of China

Abstract. Measuring semantic similarity is essential to many natural language processing (NLP) tasks. One widely used method to evaluate the similarity calculating models is to test their consistency with humans using human-scored gold-standard datasets, which consist of word pairs with corresponding similarity scores judged by human subjects. However, the descriptions on how such datasets are constructed are often not sufficient previously. Many problems, e.g. how the word pairs are selected, whether or not the scores are reasonable, etc., are not clearly addressed. In this paper, we proposed a multidisciplinary method for building and validating semantic similarity standard datasets, which is composed of 3 steps. Firstly, word pairs are selected based on computational linguistic resources. Secondly, similarities for the selected word pairs are scored by human subjects. Finally, Event-Related Potentials (ERPs) experiments are conducted to test the soundness of the constructed dataset. Using the proposed method, we finally constructed a Chinese gold-standard word similarity dataset with 260 word pairs and validated its soundness via ERP experiments. Although the paper only focused on constructing Chinese standard dataset, the proposed method is applicable to other languages.

Keywords: Word similarity · Dataset · Multidisciplinary method · ERP

1 Introduction

Measuring semantic similarity between two words is a key problem for many Natural Language Processing (NLP) applications and has attracted much attention of researchers from the NLP community. One widely used method for evaluating similarity calculating models, e.g. models based on Word2Vec [1], is to test their consistency with humans by using human-scored gold-standard datasets, which consist of word pairs with corresponding similarity scores judged by human subjects. There exist several well-known gold-standard datasets, e.g. RG65 [2], MA30 [3] and WordSimilarity353 [4] for English whereas WordSimilarity240 [5] and WordSimilarity296 [6] for Chinese. However, the descriptions on how these datasets are built are often not sufficient. For example, it is not

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_4

clearly stated whether or not length and frequency of words, which has been proven to have impacts on human psycholinguistic processing [7], are controlled when selecting word pairs. Moreover, all these datasets do not distinguish semantic similarity and semantic relatedness, which in fact have wide differences in terms of not only cognitive language processing but NLP applications [8]. Additionally, all previous work only validated their datasets by showing high correlations of the human scores and do not introduce a special validating process for testing the soundness of the scores gathered by psychological scaling. In fact, psychological scaling is easily influenced by metalinguistic and response-related processes. Therefore, it is necessary to integrate validation processes that depend on more objective measures, e.g. brain waves.

To tackle these problems, this paper proposes a multidisciplinary method for constructing semantic similarity validating dataset. Figure 1 shows the pipeline of the proposed method, which is composed of three steps. Firstly, word pairs are selected based on computational linguistic resources, i.e. data and tools from Artificial Intelligence (AI) or Computer Science (CS). Secondly, similarities for the selected word pairs are scored by human subjects, which is one of the traditional methods in Psychology. Finally, the soundness of the constructed dataset is test via Event-Related Potentials (ERPs) experiments, methods borrowed from Brain Science.

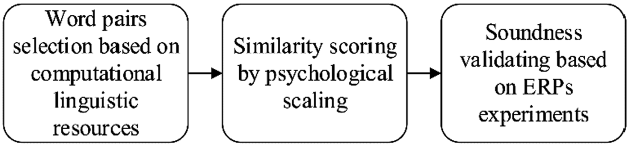

Word pairs selection based on computational linguistic resources
Similarity scoring by psychological scaling
Soundness validating based on ERPs experiments

Fig. 1. The pipeline of the proposed method

This paper uses the proposed method to construct a Chinese gold-standard word similarity dataset. And the rest of this paper describes the process in detail, which is organized as follows. Section 2 describes how we use resources from NLP to select the candidate word pairs. After that, Sect. 3 addresses the process of the similarity scoring by psychological scaling. Then the ERPs experiments for validating the soundness of the constructed dataset are presented and the results are discussed in detail in Sect. 4. Finally, Sect. 5 gives conclusion.

2 Word Pairs Selection Based on Computational Linguistic Resources

This section describes the process of the word pairs selection, during which all the words come from a frequently used Chinese corpus called Sogou News Corpus[1]. Since the similarity scoring will be performed by psychological scaling in the following step,

[1] http://www.sogou.com/labs/resource/ca.php.

several factors are taken into account in order to make the psychological experiment more reliable. Concretely, the proposed method considers four factors, i.e. word length, word frequency, word category, and data balance. The rest of this section will discuss these factors in detail.

2.1 Word Length and Frequency

It has been shown that length and frequency of words have a significant effect on the semantic processing of the subjects in (neuro)psychological experiments, when words are used as stimuli. Actually, [7] investigated the influence of the length and frequency of printed words on the amplitude and peak latencies of ERPs and found that long words produced the strongest brain response early, while lower ERPs amplitudes were elicited by words with high frequency compared with low frequency words in the latency. Therefore, good word pairs should have words with similar length and close frequency.

In terms of Chinese, which this paper focuses on, the words in dataset are determined to have the same length. Concretely, the selected length is 2-character, since most Chinese words are 2-character ones.

As far as the word frequency is concerned, we decided that in a selected word pair the frequency of the high frequency one should be at most 1.5 times that of the low frequency one. Moreover, since most high frequency words are easily to establish semantic relatedness with other words while low frequency words may often be unrecognized by subjects, we set a limitation on frequency between 250,000 and 7,500.

2.2 Entity Words vs Non-entity Words

The word category is also an important factor that should be considered. Actually, functional words, e.g. 经过 (jīngguò, via), tend to establish semantic relatedness with other words, and thus are not suitable for being chosen as stimuli. To avoid using these words in the following psychological scaling, only entity words are considered for selection in the proposed method. Concretely, HowNet, which is a famous Chinese-English common-sense knowledge base [9, 10], is used to guide the selection. Precisely, only words whose primary sememe belongs to the Entity taxonomy tree are kept for further selection.

2.3 Data Balance and Word Pairs Classification

In order to produce a good gold-standard dataset, the data balance need also being controlled. Ideally, word pairs should be evenly distributed among different scales in the final dataset. To do so, our idea is to estimate the similarity scores for candidate word pairs using traditional similarity measuring models and use these estimations to guide the selection.

In particular, two frequently used similarity measuring models, i.e. HowNet-based method and Word2Vec-based method, are applied in this paper. Specifically, we use the method proposed by [11] to calculate the HowNet-based scores and use the original implementation[2] of Word2Vec to calculate the Word2Vec-based scores. During the training of Word2Vec, the vector dimension is set as 300 and the window size is set as 5.

Carefully comparisons show that, based on HowNet and Word2Vec, we can roughly distinguish four types of word pairs, as shown in Fig. 2. First, word pairs are mostly similar if their HowNet-based scores and Word2Vec-based ones are both high. Second, word pairs that have high Word2Vec-based scores but low HowNet-based scores are also strongly related, although the relationships contain no hyponymy information. Third, most word pairs are unrelated if their HowNet-based scores and Word2Vec-based scores are both low. Finally, the word pairs whose HowNet-based score is high but Word2Vec-based one is low suggest under-fitting training on given corpus and thus are problematic.

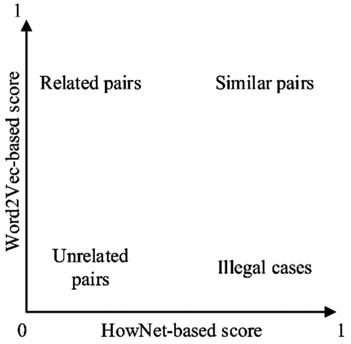


Fig. 2. Four types of word pairs

Therefore, in the proposed method, the candidate word pairs are organized into two subsets. One is called similarity subset, which contains word pairs with the corresponding Word2Vec and HowNet scores being roughly the same (i.e. similar pairs and unrelated pairs in Fig. 2), and the other is called relatedness subset, in which word pairs have high Word2Vec scores but low HowNet scores (i.e. related pairs in Fig. 2).

Precisely, when distinguishing the word pairs in the similarity subset, we require that the absolute value of the score difference between the HowNet-based scores and Word2Vec-based scores is no more than 0.2. On the other hand, word pairs in relatedness subset are distinguished if their Word2Vec-based scores are higher than 0.5 while their HowNet-based ones are lower than 0.5 and the differences are larger than 0.25.

2.4 Word Pairs Selection Result

After taking all the above-mentioned factors into consideration, we finally select 320 candidate word pairs, of which 240 ones are in the similarity subset and the other 80 pairs are in the relatedness subset.

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

3 Scoring by Psychological Scaling

Given word pair candidates derived from methods described in Sect. 2, we further score them by psychological scaling, similar to all the other dataset construction methods [2–6].

Particularly, two experiments are organized. First, we organize a preliminary experiment to ensure a higher quality dataset as possible. In the pre-experiment, we pay 15 volunteers (all aged from 20 to 22) to participate in accomplishing the psychological scale based on same randomized order, which contains 320 pairs of words that generated in Sect. 2. We set the number of scales an even integer [12] to make the results more accurate. For each pair, every subject is required to mark an integer value from 1 to 8, to describe how related two words are by his or her own cognition. The higher the score is; the more related two words are considered. After the pre-experiment, the means and variances of each word pairs are calculated. Then, 45 word pairs from the similarity subset and 15 word pairs from the relatedness subset, which with high most variances are dropped.

After the preliminary experiment, we recruit another 13 volunteers (8 males and 5 females, aged from 18 to 25 and 21.9 in average) with payments to score the final word pairs again and use their judgements as the final scores. Please note that, in this stage, every volunteer is also required to finish the same psychological scale values from 1 to 8 for every pair of words. And, the pairs of words are again randomized but every scale contains the same order.

The statistical result of psychological scale is shown in Fig. 3, from which we could learned that most word pairs that have high Word2Vec-based scores or HowNet-based scores also have high scaling scores (i.e. being dark in Fig. 3) and most word pairs with low Word2Vec-based and HowNet-based scores also have low scaling scores (i.e. being light in Fig. 3). Moreover, the statistical analysis also shows high reliability, with Cronbach's coefficient values at 0.981.

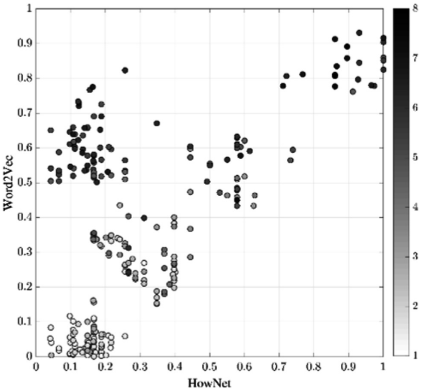


Fig. 3. Comparison on HowNet, Word2Vec and scale scores.

4 Validation Based on ERPs Experiments

This section addresses the validating process based on ERPs experiments. First, Subsect. 4.1 gives a brief introduce to ERPs. Then, the detailed setup and procedure of the ERPs experiments are described in Subsects. 4.2–4.5. After that, Subsects. 4.6–4.8 present three comparisons accordant with the discussions.

4.1 A Brief Introduce to ERPs

ERPs analysis, which first record Electroencephalography (EEG) data from the scalp and then analyze the brain waves time-locked to the relevant event, has been applied widely to psychological research or brain research for a half century. In order to obtain ERP data, three steps are necessary: (i) collection of EEG raw data while participants read or listen to linguistic input; (ii) preprocessing these raw data (e.g. filtering and artifact rejection); (iii) averaging across multiple events of the same type and comparing the resulting ERPs between conditions.

In the past decades, ERPs analysis also has been successfully applied to language research, since language processing has been regarded as the most mysterious cognitive process in human brain research [13]. Therefore, large amount of discoveries was carried on language processing, and researchers discovered several far-reaching potentials such as N400 and P600. In particular, we focus on two main potentials, i.e. N400 [14–16] and N270 [17, 18].

The N400 potential was first discovered by [14], which indicates the semantic mismatch when human prime languages. The N270 potential, which pertains to N2 family potentials, is mainly used to tribute the conflicts on the early semantic priming. That is to say, when the stimuli are out of subjects' expectation, then N270 affect would be observed on the parietal lobe.

4.2 Participants

Another 16 undergraduates (7 males and 9 females, aged from 19 to 25 and 20.7 in average) are paid to participate in the neuropsychological experiment. All the testers' mother language is Chinese, and they are all right-handed. Besides, all subjects have normal or corrected-to-normal vision and are reported no history of neurological surgery or drug.

4.3 Electroencephalography Recording

In this experiment, continuous electroencephalogram (EEG) is recorded using NeuroSCAN 4.3 with a 64-channel Quickcap and is amplified by a SynAmps2 DC-amplifier at 1,000 Hz. Before data recording, all channel impedances are maintained below 5 kΩ.

All the subjects are well seated in a comfortable chair placed in the same quiet room. And the stimuli are presented on a CRT monitor, whose screen resolution is 800 multiplied 600 at 30 Hz. When presenting the stimuli, the color of background is set to be white, whereas the color of all the characters, including the fixation symbol, are set to be black. Moreover, the distance between the eyes of subjects and the screen is 80 cm, and the screen is vertical to the ground.

4.4 Experimental Procedure

Before the formal experiment, all subjects are required to carefully read the instructions and participate in a test segment, which contains 20 trials whose words won't appear in the following formal experiment. Then, during the formal segment, 260 trials are divided into 10 groups randomly, and stimulus from each group is randomly presented to subject. Moreover, to avoid fatigue effect, we enforce subjects to take a break after they finish each group. When subject feel good about concentrating again during break, he or she can press any button to proceed. And, the average experimental time is controlled as 30 min or so.

Moreover, the procedure of each trial is shown in Fig. 4. Concretely, in the beginning, after 1,000 ms blank of screen, the duration of fixation is set to be randomized between 750 ms and 1,250 ms to avoid the expectation effect which might be raised by subjects. After that, the first word of a word pair will be shown for 400 ms on the following procedure, and then a blank screen followed for 1,000 ms. Then, the second word appears, and the subjects are required to press one of the two buttons to distinguish the degree of relatedness or similarity of the two words in 3,000 ms (button F for low and button J for high).

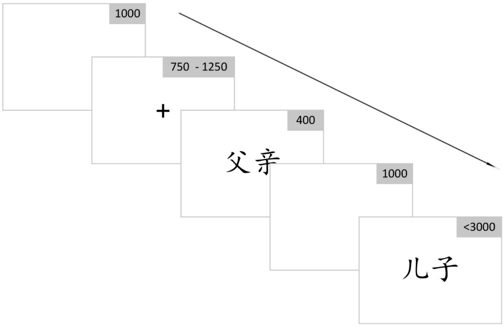


Fig. 4. The procedure of a trial (ms)

4.5 Data Analysis

Among all the data generated from subjects, 4 datasets are abandoned due to much signal noise or low accuracy. Thus, the remaining 12 data (male/female: 5/7) are available for analysis, which is performed with EEGLab.

Before the ERPs analysis, a filtering process is conducted, in which the filter is set as a band-pass from 0.1 Hz to 30 Hz and all channels are re-referenced to left mastoid (M1). After filtering, we use EEGLab to remove channel artifacts and ocular artifacts. Finally, the data from −100 ms to 800 ms are remained for further analysis.

As mentioned before, the dataset constructed in this paper consists of two subsets, i.e. the similarity subset and the relatedness subset. For data analysis, the similarity subset is further divided into three parts according to their psychological scaling scores. Therefore, we finally get four groups, as listed below:

- group A: word pairs from the relatedness subset
- group C3: word pairs from the similarity subset and with high most scaling scores
- group C1: word pairs from the similarity subset and with low most scaling scores
- group C2: the rest part of the similarity subset.

Here, we use C for categorical priming and A for associative priming, which are two types of semantic priming concluded in psycholinguistic research [19]. Actually, the categorical priming pattern is elicited by words whose relation is mainly categorical (e.g. apple - pear, tiger - lion), whereas the associative priming is induced by other relations beyond categorical information (e.g. broom - floor, car - wheel).

4.6 Result 1: Response Time Comparison

Figure 5 presents the response time, i.e. time that subjects averagely spend while making decisions, for each groups. From Fig. 5 differences among groups could be observed. Actually, the difference on response times among different groups is outstandingly significant (F(2, 22) = 21.782, p < .01).

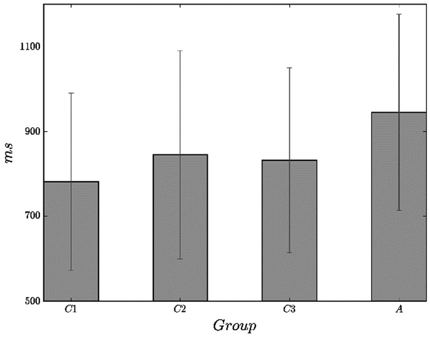
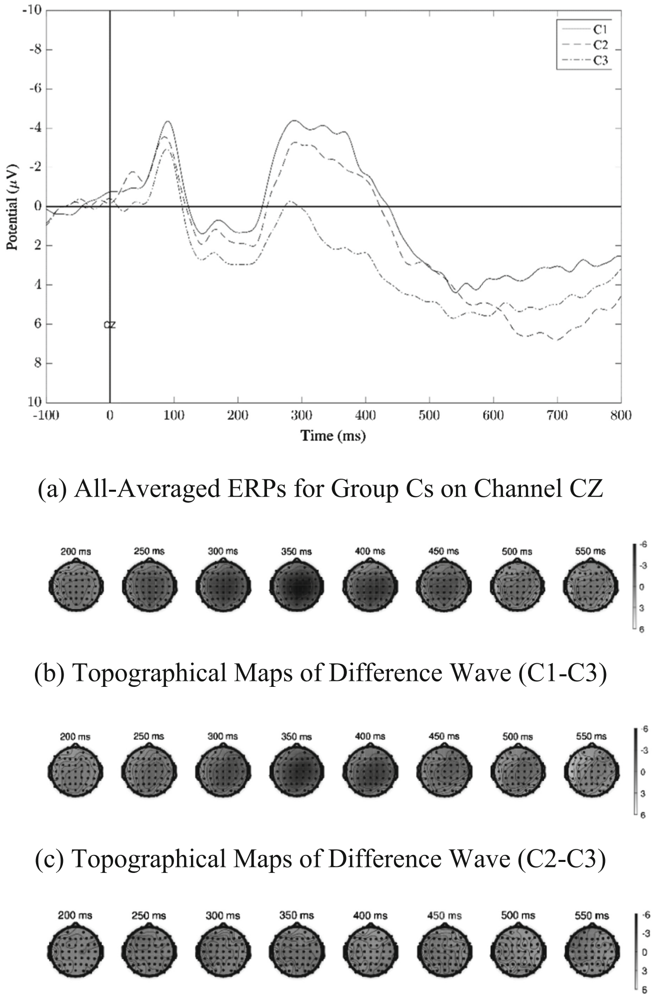


Fig. 5. Comparison on the response time

Moreover, two additional phenomena could be learned from Fig. 5. First, the response time for group C2 is larger than that of group C1 and group C3, which indicates that word pairs in group C2 are harder to be cope with. This is quite reasonable because word pairs in group C2 are vague between unrelated and highly similar. Second, the response time for group A is much larger than that of group C1, C2 and C3, which suggests that the related pairs are more difficult to be recognized than the similar pairs. This is also in line with the intuition.

4.7 Result 2: Comparison Among Different Similarity Groups

Figure 6 gives comparison among different similarity groups, i.e. group C1, C2 and C3. Concretely speaking, Fig. 6(a) shows the all-averaged ERPs curves for all three groups on channel CZ, which is one of the channels for frontal lobe. And, the topological map of the difference from every two different groups are shown in Fig. 6(b)–(d), respectively.



(a) All-Averaged ERPs for Group Cs on Channel CZ

(b) Topographical Maps of Difference Wave (C1-C3)

(c) Topographical Maps of Difference Wave (C2-C3)

(d) Topographical Maps of Difference Wave (C1-C2)

Fig. 6. Comparison among group Cs

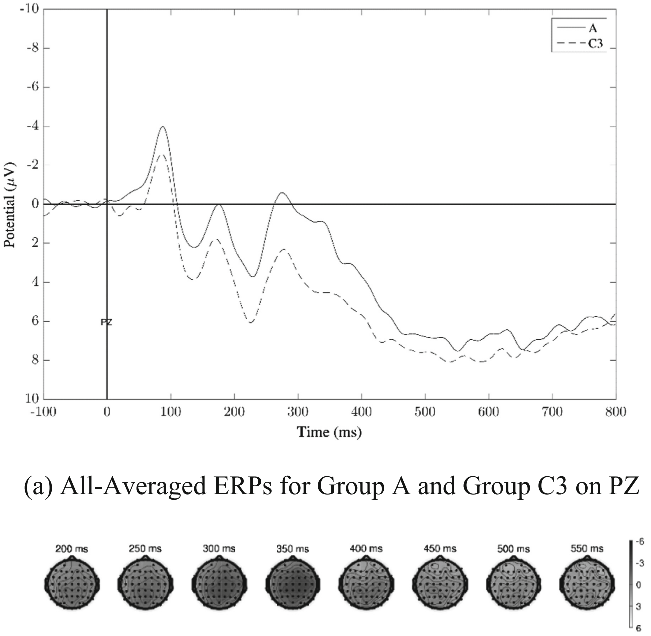
By carefully observing the curves in Fig. 6(a) between the time period from 300 ms to 500 ms, we can learn that the degree of the N400 potentials for word pairs from different groups are different significantly (Greenhouse-Geisser F(1.033, 11.368) = 70.936, $p < .01$). Precisely, word pairs from group C1 (M = −1.366, SD = 5.930) and group C2 (M = −0.466, SD = 5.278) show stronger N400 potentials than group C3 (M = 2.838, SD = 6.262) and are both significant (t(11) = −6.234, $p < .003$; t(11) = −4.173, $p < .003$). This is quite reasonable since word pairs in group C3 are highly similar while word pairs in group C1 or C2 are less similar or even unrelated.

Moreover, from these Fig. 6(b)–(d), we can see that the priming area of brain moves from parietal lobe to central lobe step by step. And it is obvious that the difference wave between group C1 and C3 is the more significant than any other one.

4.8 Result 3: Comparison Between the Similarity Group and the Relatedness Group

In Fig. 7, we present the curves for all-averaged value from group C3 and A on channel PZ, as well as the topological map of difference.

The associative priming word pairs are also related, so the ERPs taken from group A can be only compared to those from group C3. From Fig. 7, we can find that the word pairs with associative priming, i.e. those from group A, show significant



(a) All-Averaged ERPs for Group A and Group C3 on PZ

(b) Topographical Maps of Difference Wave (A-C3)

Fig. 7. Comparison between group A and group C3

difference from the word pairs with categorical priming, i.e. those from group C3, at a duration around 270 ms from parietal lobe. Actually, the difference is significant for the averaged value from 260 ms to 280 ms on channel PZ($t(11) = -3.220$, $p < .01$). That is to say, word pairs from group A were observed to arise significantly stronger N270 potentials.

As to group A and C3, the word pairs from both are possessed with strong co-occurrence frequencies. Nonetheless, the former group is a little out of semantic expectation when displayed to subjects due to their low hyponymy information. In fact, the words from group C3 can be known as semantic similarity, which means that both words are deeply related with each other than usual, because the two similar words have a high majority of attributes in common, e.g. 父亲 – 儿子 (fùqīn - érzi, father - son). When the word pairs from group C3 are presented to subjects orderly, the first word would help built a semantic conception, or "semantic space" for the second one. Whereas in terms of word pairs from group A, the second word from a word pair is slightly excluded by the semantic conception formed by the first word. And, this conflict raises a more obvious negative wave which peaks around 270 ms (i.e. N270).

5 Conclusion

In this paper, we proposed a multidisciplinary method for building semantic similarity validating datasets, which integrates tools or methods from computer science, psychology and brain science. Then the proposed method was used to construct a Chinese gold-standard word similarity dataset. The contributions of this paper are 3-folds. First, this paper introduces ERPs experiments, at the first time, to validate the soundness of the constructed data. Second, the proposed method distinguished similarity and relatedness, which is not so in the previous work. Third, we release a new Chinese gold-standard word similarity dataset.

Acknowledgments. This work was supported by National Natural Science Foundation of China (No. 61573294), National Social Science Foundation of China (No. 16AZD049) and Fujian Province 2011 Collaborative Innovation Center of TCM Health Management.

References

1. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of the International Conference on Learning Representations (ICLR), Scottsdale, Arizona, May 2013
2. Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. Commun. ACM **8** (10), 627–633 (1965)
3. Miller, G.A., Charles, W.G.: Contextual correlates of semantic similarity. Lang. Cogn. Process. **6**(1), 1–28 (1991)
4. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E.: Placing search in context: the concept revisited. In: Proceedings of the 10th International World Wide Web Conference (WWW10), Hongkong, China, pp. 406–414, May 2001

5. Wang, X., Jia, Y., Zhou, B., Ding, Z., Liang, Z.: Computing semantic relatedness using Chinese Wikipedia links and taxonomy. J. Chin. Comput. Syst. **32**(11), 2237–2242 (2011)
6. Jin, P., Wu, Y.: Semeval-2012 task 4: evaluating Chinese word similarity. In: Proceedings of the Joint Conference on Lexical and Computational Semantics, Montréal, Canada, pp. 374–377, June 2012
7. Hauk, O., Pulvermüller, F.: Effects of word length and frequency on the human event-related potential. Clin. Neurophysiol. **115**(5), 1090–1103 (2004)
8. Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., Soroa, A.: A Study on similarity and relatedness using distributional and WordNet-based approaches. In: Proceedings of North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL - HLT 2009), Colorado, pp. 19–27, June 2009
9. Dong, Z., Dong, Q.: Hownet, March 1999. http://www.keenage.com
10. Dong, Z., Dong, Q., Hao, C.: HowNet and its computation of meaning. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), Beijing, China, pp. 53–56, August 2010
11. Liu, Q., Li, S.: Word similarity computing based on HowNet. In: Proceedings of the Third Chinese Lexical Semantics Workshop, pp. 59–76 (2002)
12. Chen, C., Lee, S., Stevenson, H.W.: Response style and cross-cultural comparisons of rating scales among East Asian and North American students. Psychol. Sci. **6**(3), 170–175 (1995)
13. Kutas, M., Federmeier, K.D.: Thirty years and counting: finding meaning in the N400 component of the event related brain potential (ERP). Annu. Rev. Psychol. **62**, 621–647 (2011)
14. Kutas, M., Hillyard, S.A.: Reading senseless sentences: brain potentials reflect semantic incongruity. Science **207**(4427), 203–205 (1980)
15. Deacon, D., Hewitt, S., Yang, C., Nagata, M.: Event-related potential indices of semantic priming using masked and unmasked words: evidence that the N400 does not reflect a post-lexical process. Cogn. Brain. Res. **9**(2), 137–146 (2000)
16. Kiefer, M.: The N400 is modulated by unconsciously perceived masked words: further evidence for an automatic spreading activation account of N400 priming effects. Cogn. Brain. Res. **13**(1), 27–39 (2002)
17. Mao, W., Wang, Y.: Various conflicts from ventral and dorsal streams are sequentially processed in a common system. Exp. Brain Res. **177**, 113–121 (2007)
18. Bennett, M.A., Duke, P.A., Fuggetta, G.: Event-related potential N270 delayed and enhanced by the conjunction of relevant and irrelevant perceptual mismatch. Psychophysiology **51**(5), 456–463 (2014)
19. Moss, H.E., Ostrin, R.K., Tyler, L.K., Marslen, W.D.: Accessing different types of lexical semantic information: evidence from priming. J. Exp. Psychol. Learn. Mem. Cogn. **21**(4), 863–883 (1995)

Fuzzy Connected-Triple for Predicting Inter-variable Correlation

Zhenpeng Li, Changjing Shang(✉), and Qiang Shen

Department of Computer Science,
Institute of Mathematics, Physics and Computer Science,
Aberystwyth University, Aberystwyth, UK
{zhl6,cns,qqs}@aber.ac.uk

Abstract. Identifying relationship between attribute variables from different data sources is an emerging field in data mining. However, currently there seldom exist effective methods designed for this particular problem. In this paper, a novel approach for inter-variable correlation prediction is proposed through the employment of the concept of connected-triple, and implemented with fuzzy logic. By the use of link strength measurements and fuzzy inference, the job of detecting similar or related variables can be accomplished via examining the link relation patterns. Comparative experimental investigations are carried out, demonstrating the potential of the proposed work in generating acceptable predicted results, while involving only simple computations.

Keywords: Connected-triple · Fuzzy inference · Link analysis

1 Introduction

In the past decades, with the development of information technology, data plays an increasingly significant role in different fields of daily life. Data are collected and gathered at separate times, in diverse places, by different individuals. Also, data can, and in reality does exist in various sources, including mobile devices, public and private clouds, subscription-based services such as file sync and share, and virtual machines, to name just a few. As the amount of available data grows, the problem of managing and analysing the information embedded in the data becomes more difficult. Fortunately, the increasing growth of computational capability has enabled the handling of such large amount of data through a range of means, including the method of social network analysis (SNA) that has been increasingly gaining popularity. In particular, recently, SNA has become an important and effective technique in the study of sociology, economy, education [1], and even in the field of national defence, such as terrorism detection [2].

Social networks are fundamentally social structures including actors and relationships amid them [3]. These networks are represented by employing vertices and links. The links show types of relationship amongst the vertices including kinship, friendship, collaborations, and any other interactions between the people in the network [4]. However, the SNA models are not only restricted to the

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_5

use in networks regarding human beings, but can also be employed to analyse the networks in a wide variety of problem domains.

In SNA, link prediction is one of the most salient tasks, including the discovery of missing or developing links in a certain network [5]. Unlike previous research that focussed on identifying links between objects or entities in a specific region, this work is driven by the interests in searching for links between variables extracted from different data/information sources. When considering the task of link prediction, connected-triple, a graph model formed by three vertices and two undirected edges, with each edge connecting two distinct vertices out of the three respectively, has an intuitive appeal. This offers a potentially effective mechanism for dealing with the problem of link prediction, particularly when any given information contents are obtained from different data sources where parts of the information overlap. Such link prediction problems are obviously of general interest in many data mining applications.

The potential underlying links between variables or entities collected from different sources are usually hidden behind and not obvious to be discovered, making the task of link prediction from such data sources a challenge. Traditionally, this type of work has generally been handled by human experts. Thus, designing and implementing a predicting method which learns from human logical reasoning will be helpful to automate such prediction processes, especially when facing large and diverse data sources. Practically, when describing a link or a set of links, linguistic terms such as "Strong", "Medium" and "Weak" are natural adjectives to depict the link strength rather than crisp numerical values, to be consistent with human customs. Also, the common knowledge such as "if A has strong link to B, and B has strong link to C, then A may have strong link to C" perfectly matches human logical thinking. Inspired by this observation, fuzzy logic, a theoretical interface between Mathematical models and human reasoning, is adopted in the present to serve as the basis upon which to develop a multi-source link prediction model.

In particular, this paper presents two major contributions: (1) It proposes a novel approach to determining the correlation between attribute variables from distinct datasets with different entity references. (2) It proposes a fuzzy link prediction model which radically departs from conventional crisp representation of connected-triple-based link detection, resulting in models that resemble human inference and thereby, having the advantages of model interpretability and simplicity. The rest of this paper is arranged as follows. Section 2 introduces the proposed architecture for building a fuzzy connected-triple system for link prediction, describing details on issues such as model construction, link measures (including complexity analyses), and inference procedures. Section 3 shows the experimental results, supported by comparative studies with alternative predicting methods. Section 4 concludes the paper with suggestions for further development.

2 Predicting System

This section presents the proposed general framework for developing a system that predicts link strengths with data from multiple sources. It describes the system's components and their associated time complexity analyses.

2.1 Conceptual Framework

The structure of the predicting system is shown in Fig. 1. As can be seen, it comprises three distinct component subsystems, each of which implements the functionality of: triple extraction, link analysis, and fuzzy inference, respectively. These activities are integrated to construct the predicting model, whose implementation steps are detailed below.

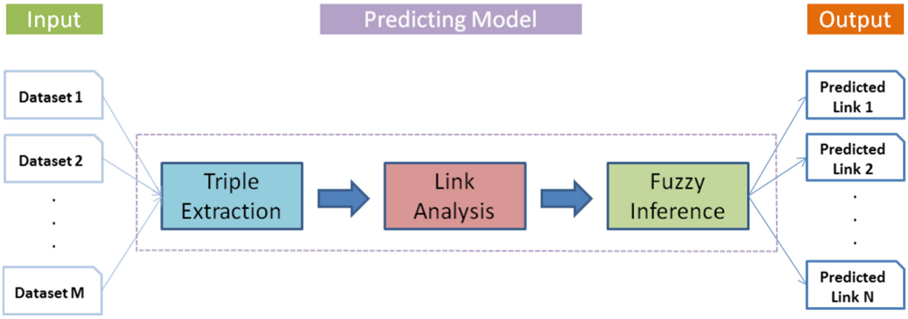


Fig. 1. Predicting framework

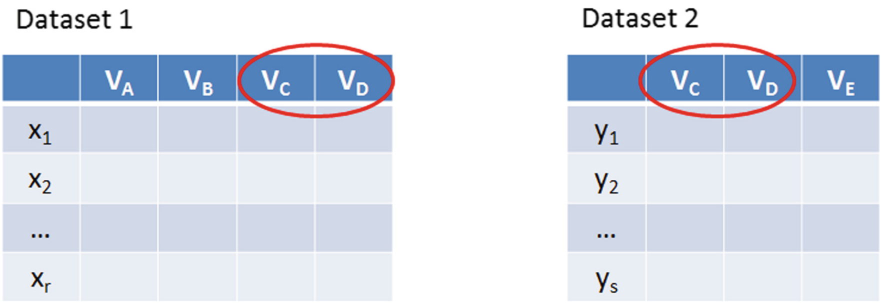
2.2 Connected-Triple Extraction

Concept of Connected Triple. Connected-triple modelling, first introduced to analyse global clustering coefficient [6], is also referred to as a method for measuring network transitivity. For instance, it measures the extent to which a friend of someone's friend is also the friend of that person. Formally, a connected-triple, $Triple = \{V_{Triple}, W_{Triple}\}$, is a subgraph of $G(V, W)$, where V represents the set of vertices in the graph and W represents the set of edges connecting related pairs of vertices, containing three vertices $V_{Triple} = \{v_i, v_j, v_k\} \subset V$ and two edges $W_{Triple} = \{w_{ij}, w_{jk}\} \subset W$, with $w_{ik} \notin W$. The vertex v_j connecting the other two vertices is called the centre of the triple, and v_i or v_k is called an end of the triple (there being two ends per triple, of course).

Extracting Connected-Triples from Datasets. Extracting connected-triples from original datasets plays a fundamental role in the present work. An example of two distinct datasets is shown in Fig. 2, where the variables V_C and V_D co-occur in both datasets (encircled in red), whist the variables V_A and V_B

only appear in Dataset 1, and V_E only appear in Dataset 2. Importantly, an obvious but crucial point is that although there exist variables co-occurring in more than one dataset, these datasets cannot be easily merged into one since the instances in the datasets can be totally distinct. For example, the instances $x_1, x_2, \ldots, x_r$ in Dataset 1 and the instances y_1, $y_2, \ldots, x_s$ in Dataset 2 are completely different from each other, although they share the two aforementioned common variables.

An example of extracting connected-triples from original datasets is shown in Fig. 3, with each vertex representing a variable in the sample datasets given in Fig. 2. For instance, v_A in Fig. 3 denotes the variable V_A in Dataset 1 of Fig. 2. A link (represented in a solid line) between two distinct variables denotes that these variables are co-occurring in at least one of the sample datasets, and therefore, that they are to a certain extent related to each other. In Fig. 3, four triples, $Triple\_i, i = 1, 2, 3, 4$, are formed from the Datasets 1 and 2 in Fig. 2, where $V_{Triple\_1} = \{v_A, v_C, v_E\}$, $V_{Triple\_2} = \{v_A, v_D, v_E\}$, $V_{Triple\_3} = \{v_B, v_C, v_E\}$, and $V_{Triple\_4} = \{v_B, v_D, v_E\}$. The centre of these four connected-triples are v_C and v_D, respectively. The dash line between v_A and v_E and that between v_B and v_E represent the potential links between pairs of the variables V_A and V_E and those of V_B and V_E, respectively, which do not exist in the present given datasets.



Dataset 1

	V_A	V_B	V_C	V_D
x_1				
x_2				
...				
x_r				

Dataset 2

	V_C	V_D	V_E
y_1			
y_2			
...			
y_s			

Fig. 2. Sample datasets

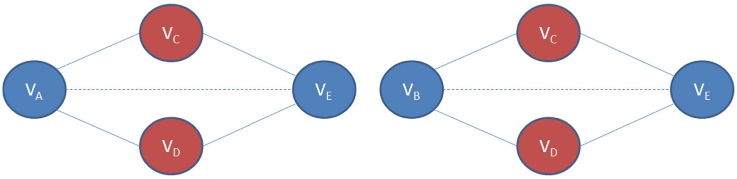


Fig. 3. Connected-triples of variables extracted from sample datasets

2.3 Link Strength Measurement

Having identified a new connected-triple from the source datasets, the task of determining correlation between a pair of variables that belong to two different datasets becomes to predict whether there exists a (hidden) link between the two end vertices. And if so, what may be the strength on such a link. To address this issue, prerequisites including the properties of the known links between pairs of vertices in the triple need to be obtained in advance.

In practice, the link property is generally described by its weight, which may correspond to a wide variety of aspects depending on the underlying application problem. For each connection between a given pair of distinct variables, different mechanisms may therefore be devised for estimating the strength of that connection. For instance, in a route graph or map, the weight of the link may indicate the route distance between two venues. In a graph of co-authorship, the weight of the link may denote the number of papers two authors collaborated to publish. In a graph of webpage linkage, the weight of the link may depict the popularity of people stepping from one to another. In the current study, the link between vertices represents the relationship between variables in datasets. Thus, the weight of a link is utilised to capture and reflect the closeness or correlation of the corresponding variables. For a particular variable pair in a dataset filled with binary or nominal attribute variables, the underlying characteristics can be described by the co-occurring frequency of the pairs of the variables concerned that take on a common term or value. From this observation, two indices for link strength measurement are adopted, named Normalised Mutual Information (*NMI*) and Frequency of Most Popular Term-Pair (*FMTP*), respectively.

Normalised Mutual Information. Mutual information is a symmetric measure to quantify the statistical information shared between two distributions [7]. The use of this measure in the present research provides a sound indication of the shared information between a pair of variables. In particular, for two discrete random variables V_A and V_B, the mutual information between them can be denoted as $MI(V_A, V_B)$ and computed by

$$MI(V_A, V_B) = \sum_{v_b \in V_B} \sum_{v_a \in V_A} p(v_a, v_b) \log(\frac{p(v_a, v_b)}{p(v_a)p(v_b)}) \tag{1}$$

where $p(v_a, v_b)$ is the joint probability distribution function of V_A and V_B, and $p(v_a)$ and $p(v_b)$ are the marginal probability distribution functions of V_A and V_B, respectively. It is obvious to see that there is no upper bound for $MI(V_A, V_B)$. Thus, for better facilitating interpretation and comparison, a normalised version of $MI(V_A, V_B)$ that ranges from 0 to 1 is desirable while describing the relationship strength between V_A and V_B.

Let $H(V_A)$ denote the entropy of V_A [8], which is defined by

$$H(V_A) = -\sum_{v_a \in V_A} p(v_a) \log p(v_a) \tag{2}$$

From this, the normalised mutual information between V_A and V_B [9], named $NMI(V_A, V_B)$, can be computed such that

$$NMI(V_A, V_B) = \frac{MI(V_A, V_B)}{\sqrt{H(V_A)H(V_B)}} \tag{3}$$

As desired, it is obvious that $0 \leq \mathit{NMI}(V_A, V_B) \leq 1$. The time complexity of computing *NMI* is $O(mnd)$, where d denotes the number of instances in the datasets, and m and n represents the number of variable terms for V_A and V_B, respectively. Typically, m and n are fixed to a small or medium number in advance. Therefore, this measurement has the linear time complexity proportional to the size of the datasets, namely $O(d)$.

Frequency of Most Popular Term-Pair. NMI may be a simple measurement computationally, for a data driven system, only taking it into consideration when modelling the link strengths between distinct variables may not be sufficiently effective. One aspect of particular interest to note is that the frequency of occurrence of different terms with regard to a certain variable within given datasets can be rather different. This is because datasets may be rather skewed; certain terms may have a very high occurrence frequency but one or more of the others may have a very low frequency. For instance, more than 90% of the primary school pupils are guarded by their parents and they are much less likely to be guarded by other relatives. The statistics of blood type distribution in the UK also shows this phenomenon, with 44% of the population have blood type O, and only 10% have blood type B [10], despite this is in a completely different problem domain.

When considering the link relationship between two variables V_A and V_B of such skewed datasets, suppose that V_A^1 and V_B^1 are each the most popular term to V_A and V_B, respectively. Then, even if most of the instances have the term V_A^1 for V_A and term V_B^1 for V_B simultaneously, the *NMI* score of the link between V_A and V_B might be low, since the number of other term-pairs and their proportion would largely affect the *NMI* score. In this case, judging the link strength between these two distinct variables by only calculating the *NMI* score may seriously distort the result, misinterpreting the closeness of the relationship between the two. This calls for the development of the so-called frequency of the most popular term-pair measure (*FMPT*).

Without losing generality, assume that a given dataset includes a total of d instances, and that V_A and V_B are two discrete variables describing the instances in the dataset, each containing m and n variable terms, respectively. Let V_A^i $(1 \leq i \leq m)$ and V_B^j $(1 \leq j \leq n)$ be the terms possibly taken by V_A and V_B, and $S_{V_A^i}$ and $S_{V_B^j}$ $(1 \leq j \leq n)$ be the set of instances which has the term V_A^i for V_A and V_B^i for V_B. The *FMPT* score of the link between the variable V_A and V_B is calculated as follows:

$$\mathit{FMPT}(V_A, V_B) = \frac{\max\limits_{1 \leq i \leq m, 1 \leq j \leq n} d_{(S_{V_A^i} \bigcap S_{V_B^j})}}{d} \tag{4}$$

where $d_{(S_{V_A^i} \bigcap S_{V_B^j})}$ denotes the number of instances which have the term V_A^i for the variable V_A and V_B^j for V_B simultaneously. Note that the *FMPT* score is also ranged from [0, 1]. The time complexity of computing *FMPT* is also $O(mnd)$, where m, n, d are the same as defined above.

Fusion of Link Properties. As indicated previously, both *NMI* and *FMPT* take values from the same range [0, 1]. It is therefore convenient to aggregate the results of applying them both. The fusion of these two measurements is useful because they capture different underlying properties of the datasets in general and the variables' terms in particular. For a particular link between two distinct discrete variables V_A and V_B, given the *NMI* and *FMPT* scores, the synthesised weight of the link $SYN(V_A, V_B)$ can be calculated in a straightforward manner by

$$SYN(V_A, V_B) = max(NMI(V_A, V_B), FMPT(V_A, V_B)) \qquad (5)$$

The synthesised link weight has the same real value range as either of the component weights, i.e., between 0 and 1. The complexity of this fusion step is extremely simple, being $O(N_l)$, where N_l denotes the number of individual strengths measured over the link. For the current investigation, as described above, N_l equals 2. The benefit of adopting the maximum operator is that it takes into consideration the most salient feature of the data while being simple in computation. Note that the strength fusion does not have to be implemented as above, but can be done in various alternative ways, e.g., by finding the arithmetic average of the component strengths. The implications of such alternative definitions remain as further research.

2.4 Fuzzy Inference Model

Having determined the weights for the known links in the connected-triples, the predicting system reaches its final step: logic deduction. A fuzzy inference model is employed to implement this task, providing a flexible means to perform human-interpretable reasoning by the use of linguistic terms rather than numeric values (although the linguistic terms still have their underlying numerical interpretations). For the problem of link prediction, linguistic labels such as "Strong", "Medium" and "Weak" are natural words that are commonly used to describe link strengths, according to human experience. The work follows this approach, with the following production rule applied to representing the concept of connected-triple:

IF $link_1$ *is* $(strong\backslash medium\backslash weak)$ AND $link_2$ *is* $(strong\backslash medium\backslash weak)$,
THEN $link_3$ *is* $(strong\backslash medium\backslash weak)$

where $link_1$ and $link_2$ represent the two known links in a certain triple, each of which connects the centre to one of the ends, and $link_3$ represents the link to be established with a (predicted) link strength score. Such a fuzzy system involves two key procedures as detailed below.

Link Weight Fuzzification. In order to enable the learning of fuzzy inference model, fuzzification of the link weights for all the connected-triples identified is necessary. A set of membership functions for link strengths is prescribed by domain experts. However, for applications where there is a sufficient amount of historical data, a clustering method may be employed to derive the required set of (potentially more objective) linguistic terms. In this work, especially for the experimental evaluation to be presented in the next section, the linguistic terms used are predefined without any optimisation and are shown in Fig. 4.

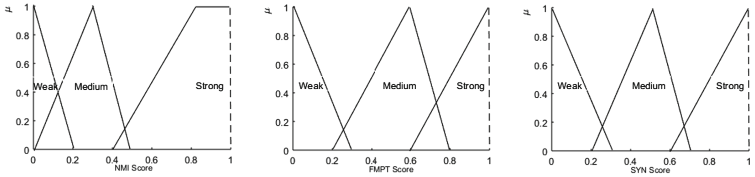


Fig. 4. Fuzzy membership value of link weight with respect to different measures

Fuzzy Inference. In the process of fuzzy inference, as with other applications of fuzzy systems, *t-norm* and *t-conorm* operators are adopted to interpret logic connectives over connected-triples, aggregating fuzzy values [11]. In general, for each pair of end vertices, there may exist several distinct centres connecting them to form different connected-triples. As such, each connection will lead to an intermediate inference outcome regarding the link strength, indicating the level that triple may contribute towards the final prediction result. Thus, an *t-conorm* operator is needed to aggregate all the intermediate predicted outcomes together.

Given a connected-triple CT, let $f^{L}_{link_1}$ and $f^{L}_{link_2}$ be the fuzzy membership values of the link strengths, or link weights on the links $link_1$ and $link_2$, where linguistic terms $L \in \mathcal{L}$, and $\mathcal{L}$ represents a collection of all fuzzy sets used to represent the linguistic labels (namely, the terms "Strong", "Medium" and "Weak" as given in the previous example). The predicted result of the triple can then be described as:

$$P_{CT} = \varDelta(\nabla_{L \in \mathcal{L}}(f^{L}_{link_1}, f^{L}_{link_2})) \tag{6}$$

where $\varDelta$ and ∇ represent a certain *t-conorm* and *t-norm*, respectively. Suppose that there are N connected-triples formed by a specific pair of end vertices with a common corresponding centre, the final predicted link strength $P_{linkweight}$ can be logically interpreted as the following:

$$P_{linkweight} = \varDelta(P^{1}_{CT}, P^{2}_{CT}, \ldots, P^{N}_{CT}) \tag{7}$$

This gives the membership value of the link weight calculated, $f^{L_p}_{P_{linkweight}}$. Then, the final step is to map such a numerical value onto the quality space of linguistic labels. The linguistic term which achieves the highest fuzzy membership value from this numerical link weight will be assigned to the link, as formulated in the following schema:

$$P_{linkstrength} = \arg\max_{L_p \in \mathcal{L}}(f^{L_p}_{P_{linkweight}}) \tag{8}$$

The time complexity of the proposed fuzzy inference model for a specific pair of attribute variables is $O(NM)$, with M denoting the number of linguistic terms for link strength description. Assume that a total number of p pairs of attribute variables are extracted from the whole corpus of datasets, the time complexity of this fuzzy inference model is $O(NMp)$.

3 Experimental Evaluation

This section presents experimental studies of the proposed approach, including a simple example of link prediction. The proposed inference model is applied to both the research field of student academic performance and that of individual credit evaluation.

3.1 Datasets

The proposed model has been firstly applied in the study of student academic performance. The provided method is utilised to deduce the relevance amongst attributes regarding student academic performance. Four corpus of datasets are used as examples, each containing hundreds of instances and collected from two schools (short-named as GP and MS hereafter) about their students in Maths and Language studies [12]. Each corpus comprises several datasets with respect to students' personal status, family background, normal study behaviour, and leisure activity. All the datasets in the corpus are logically connected with each other through the variables that indicate different aspects of student academic performance. The size of each dataset is different and the instances within different datasets are not necessarily referring to the same set of students. Before implementing the inference model, data preprocessing is carried out. Generally, each numeric variable is discretised in order to perform link analysis.

Another experimentation is on credit evaluation. A corpus of datasets is used here, each containing thousands of instances with variables indicating individual's personal identity, occupation status, residential status, and family relations. Each dataset is somewhat connected with others through several overlapping variables, thereby offering the potential for detecting the underlying links hidden behind. Again, data preprocessing is carried out as above, prior to the commencement of experimental investigation.

3.2 Implementation of Fuzzy Inference Model

When conducting the experiments, for simplicity and clarity, *t-norm* and *t-conorm* are initially implemented with minimum and maximum operators respectively. To reflect the flexibility of the proposed approach, and also to strengthen comparative studies, another type of operator combination, namely, algebraic product and bounded sum, are applied to form the Bounded Sum-Algebraic Product (BSAP) interpretation. In the step of aggregating the predicted outcomes of different connected-triples, maximum *t-conorm* is employed for both approaches.

3.3 Illustrative Example of Link Prediction

Consider two small sample datasets related to student academic performance, containing 14 and 10 distinct instances, respectively, as shown in Fig. 5. These two datasets share the common attribute variables "1^{st} semester grade" and "2^{nd} semester grade". The task is to discover the correlation between the attribute variable "Family support" in sample dataset 1 and "Family size" in dataset 2.

Two connected-triples can be extracted from the given datasets, namely:

$$Triple\_1 = \{\{v_{fsup}, v_{1sg}, v_{fsize}\}, \{w_{fsup-1sg}, w_{1sg-fsize}\}\}$$

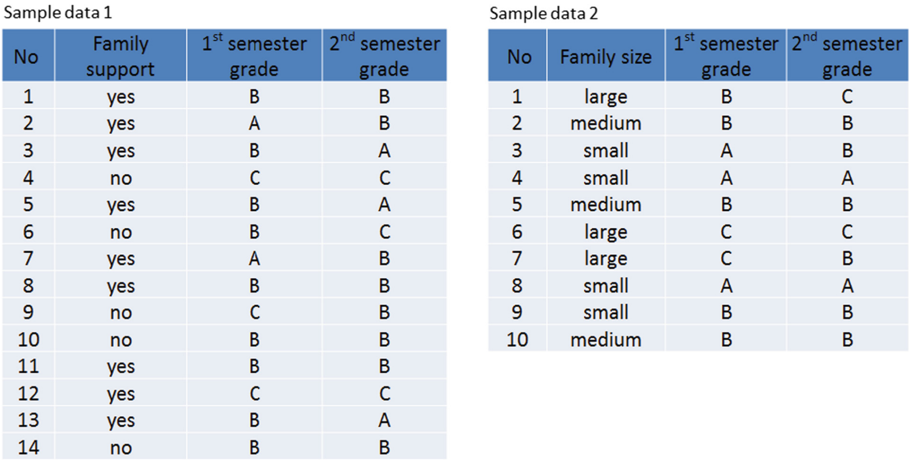
$$Triple\_2 = \{\{v_{fsup}, v_{2sg}, v_{fsize}\}, \{w_{fsup-2sg}, w_{2sg-fsize}\}\}$$

where "$fsup$", "$1sg$", "$2sg$", "$fzise$" stand for the variables "Family support","1^{st} semester grade","2^{nd} semester grade","Family size", respectively.

According to Eqs. (1), (2) and (3), the following can be computed: $NMI(V_{fsup}, V_{1sg}) = 0.139$, $NMI(V_{fsup}, V_{2sg}) = 0.172$, $NMI(V_{fsize}, V_{1sg}) = 0.58$, $NMI(V_{fsize}, V_{2sg}) = 0.474$ can be computed. Similarly, by Eq. (4), $FMPT(V_{fsup}, V_{1sg}) = 0.429$, $FMPT(V_{fsup}, V_{2sg}) = 0.357$, $FMPT(V_{fsize}, V_{1sg}) = 0.3$, $FMPT(V_{fsize}, V_{2sg}) = 0.3$. From these scores, the SYN scores can be obtained directly via Eq. (5) such that $SYN(V_{fsup}, V_{1sg}) = max(0.139, 0.429) = 0.429$, $SYN(V_{fsup}, V_{2sg}) = max(0.172, 0.357) = 0.357$, $SYN(V_{fsize}, V_{1sg}) = max(0.58, 0.3) = 0.58$, and $SYN(V_{fsup}, V_{2sg}) = max(0.474, 0.3) = 0.474$.

Having acquired the scores for the given links, the next step is to conduct fuzzy inference. For illustrative simplicity, apply the Max-Min aggregation method over the SYN scores. Thus, for $Triple\_1$, the P_{CT} score can be calculated by

$$max(min(f^S(0.429), f^S(0.58)), min(f^M(0.429), f^M(0.58)), min(f^W(0.429), f^W(0.58))) = 0.6$$



Sample data 1

No	Family support	1^{st} semester grade	2^{nd} semester grade
1	yes	B	B
2	yes	A	B
3	yes	B	A
4	no	C	C
5	yes	B	A
6	no	B	C
7	yes	A	B
8	yes	B	B
9	no	C	B
10	no	B	B
11	yes	B	B
12	yes	C	C
13	yes	B	A
14	no	B	B

Sample data 2

No	Family size	1^{st} semester grade	2^{nd} semester grade
1	large	B	C
2	medium	B	B
3	small	A	B
4	small	A	A
5	medium	B	B
6	large	C	C
7	large	C	B
8	small	A	A
9	small	B	B
10	medium	B	B

Fig. 5. Example of two small scale datasets

where f^S, f^M, f^W denote the fuzzification results of the link scores with respect to the linguistic terms "Strong", "Medium" and "Weak", respectively. Likewise, for $Triple\_2$, the P_{CT} score is computed to be 0.523. Hence, the link weight of $max(0.6, 0.523) = 0.6$ between the variable "Family support" and "Family size" results. Finally, a linguistic link weight between these two variables can be obtained, by mapping the numerical link weight back onto the corresponding linguistic quantity space and selecting the linguistic term that has the maximum fuzzy membership value.

3.4 Experimental Setup

In the experiments, for each corpus, its included datasets are split into subsets to enable 10-fold cross validation [13]. The following reported results are based on an average of running 10 times 10-fold cross validation. Note that the ground truth of the link strengths between variables is not a natural existence in such datasets. Thus, in these experiments, the ground truth is artificially computed by the testing data using the same NMI, FMPT or their synthesis as outlined in Sect. 2.3. That is, without losing fairness, the predicted results of each measurement are compared against those directly generated by the same underlying link strength measurement from the testing data. Note however that for real-world application, such ground truth values are of course unknown; else, the links are already established, be they weak or strong or otherwise.

Other well-known approaches for predicting similarity, including SimRank [14] and PageSim [15], are also implemented for comparison. Similarly, the link weights generated by different measures are set to these models as initial states. The decaying factors in these algorithms are set to 0.8 and 0.85, respectively, as being widely used in various applications. Furthermore, in PageSim, the maximum connecting-path is set to be of a length of 2 for computational simplicity.

3.5 Results and Discussion

The experimental results are shown in Table 1, with the best performance on each corpus of datasets highlighted in boldface.

Table 1. Comparison in terms of predicting accuracy (%)

Datasets	PA(Max-Min)			PA(BSAP)			SimRank			PageSim		
	NMI	FMPT	SYN	NMI	FMPT	SYN	NMI	FMPT	SYN	NMI	FMPT	SYN
Maths (GP)	83.4	74.5	84.1	82.2	70.5	**84.4**	77.2	70.7	79.9	74.3	67.9	77.4
Language (GP)	77.2	69.4	**80.6**	77.6	70.2	79.4	74.2	70.7	75.1	72.6	65.8	75.2
Maths (MS)	84.2	74.8	88.6	86.6	77.3	**89.4**	78.1	71.7	83.2	79.2	70.9	85.4
Language (MS)	74.2	64.3	78.6	79.6	71.2	**83.4**	71.6	64.4	77.4	70.2	60.1	74.9
Credit	79.2	73.4	81.5	84.6	75.9	**88.1**	72.2	65.8	76.1	70.2	66.1	73.3

*Predicting Accuracy: Number of correct predictions with regard to artificially computed ground truth over number of total predictions

**PA: Proposed Approach

From Table 1, it is obvious to see that the proposed model generates better predicted result than both SimRank and PageSim. Also, the model using BSAP for link strength measurement generally outperforms its Max-Min counterpart, while the model using synthesised link properties (SYN in the table) leads to the most accurate predicted results for most of the cases. The possible reason for this is that it takes the advantage of both component link property measurement methods. An interesting discovery is that although the *FMPT* measure itself does not seem to be able to produce good link prediction, its combination with *NMI* improves the performance significantly.

Note that the (artificially computed) ground truth of the link strength for a majority of the variable pairs is detected to be "Weak" no matter which strength measurement method is used. However, in real-world applications, it is the identification of any variable pair that is associated with a "Strong" link weight that is generally more attractive to the user. Thus, it is of practical importance to evaluate the capability of the proposed model in predicting "Strong" links. To conduct such an investigation, the criteria of precision and recall are utilised to examine the predicted results, where:

$$Precision = \frac{Number\ of\ Genuine\ Strong\ Links\ Disclosed}{Number\ of\ Disclosed\ Strong\ Links}$$

$$Recall = \frac{Number\ of\ Disclosed\ Strong\ Links}{Number\ of\ All\ Strong\ Links}$$

The precision and recall scores of the predicted results regarding "Strong" links are shown in Table 2. Note that all the results listed in this table are not so satisfactory as what was presented in Table 1. This is because the task of prediction for "strong" links is much more difficult compared to the task of general prediction [16]. Despite this fact, three major observations can be made: (1) The proposed model, using either Max-Min or BASP to implement fuzzy inference, outperforms both SimRank and PageSim in predicting "Strong" links. (2) BSAP-based fuzzy system that uses synthesised strength measurement generates, in four out of the five cases, the best predicated results (as highlighted in boldface). (3) Synthesised link measurement consistently outperforms either of the two single link measurements, showcasing its effectiveness in link prediction.

Table 2. Precision and recall rates over predicting strong links

Dataset	PA(Max-Min)			PA(BSAP)			SimRank			PageSim		
	NMI	FMPT	SYN	NMI	FMPT	SYN	NMI	FMPT	SYN	NMI	FMPT	SYN
Maths (GP)	.56/.49	.48/.44	.58/.60	.52/.71	.46/.62	**.64/.71**	.50/.58	.49/.59	.52/.61	.48/.56	.44/.55	.49/.58
Language (GP)	.58/.55	.41/.40	**.59/.61**	.53/.58	.46/.55	.57/.59	.49/.55	.42/.58	.52/.58	.50/.53	.41/.52	.50/.54
Maths (MS)	.61/.67	.52/.54	.64/.69	.64/.68	.58/.61	**.69/.73**	.54/.61	.48/.57	.55/.61	.50/.54	.45/.49	.52/.51
Language (MS)	.62/.68	.51/.52	.65/.69	.68/.67	.57/.55	**.70/.72**	.55/.62	.47/.50	.56/.65	.52/.59	.48/.53	.52/.57
Credit	.55/.72	.51/.68	.57/.68	.59/.72	.52/.70	**.60/.72**	.50/.55	.45/.54	.52/.55	.53/.52	.46/.49	.56/.54

4 Conclusion

This paper has proposed a novel approach to predicting the connections between variables that are hidden in different datasets. Assisted with the concept of connected-triple, the correlation between distinct variables can be naturally represented through the link notation, and the transitivity notion of "if xR_1y and yR_2z then xR_3z" (with R_1, R_2, R_3 each representing a certain relation) can be captured with logic interpretation. The use of fuzzy inference supports the link prediction process to be more consistent with human reasoning, with the predicted results being readily interpretable. Experimental results on different corpuses of datasets have shown that the proposed approach generates more accurate predicted outcomes, while involving simple computation.

Whilst promising, the proposed work opens up an avenue for further investigation. For the present predicting framework, only two types of link property measurement are considered; developing other types may help further improve the modelling performance. Similarly, alternative aggregating methods (e.g., arithmetic average and Ordered Weighted Averaging as employed in [17]) for fuzzy inference may also be worth investigating. Additionally, experimentation needs to be carried out on more datasets to testify its efficacy in different application domains.

References

1. Wasserman, S., Galaskiewicz, J.: Advances in Social Network Analysis: Research in The Social and Behavioral Sciences, vol. 171. Sage Publications, Thousand Oaks (1994)
2. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.-U.: Complex networks: structure and dynamics. Phys. Rep. **424**(4), 175–308 (2006)
3. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications, vol. 8. Cambridge University Press, Cambridge (1994)
4. Newman, M.E.J., Park, J.: Why social networks are different from other types of networks? Phys. Rev. E **68**(3), 036122 (2003)
5. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. J. Assoc. Inf. Sci. Technol. **58**(7), 1019–1031 (2007)
6. Luce, R.D., Perry, A.D.: A method of matrix analysis of group structure. Psychometrika **14**(2), 95–116 (1949)
7. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, Hoboken (2012)
8. Liang, J., Shi, Z.: The information entropy, rough entropy and knowledge granulation in rough set theory. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **12**(01), 37–46 (2004)
9. Strehl, A., Ghosh, J.: Cluster ensembles–a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**, 583–617 (2002)
10. Reid, M.E., Lomas-Francis, C., Olsson, M.L.: The blood group antigen factsbook. Academic Press, Cambridge (2012)
11. Deschrijver, G., Cornelis, C., Kerre, E.E.: On the representation of intuitionistic fuzzy t-norms and t-conorms. IEEE Trans. Fuzzy Syst. **12**(1), 45–61 (2004)

12. Cortez, P., Silva, A.: Using data mining to predict secondary school student performance. In: Proceedings of 5th Future Business Technology Conference (FUBUTEC 2008), pp. 5–12. EUROSIS, Porto, Portugal, April 2008
13. Bengio, Y., Grandvalet, Y.: Bias in estimating the variance of K-fold cross-validation. In: Statistical Modeling and Analysis for Complex Data Problems, pp. 75–95. Springer (2005)
14. Jeh, G., Widom, J.: Simrank: a measure of structural-context similarity. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 538–543. ACM (2002)
15. Lin, Z., King, I., Lyu, M.R.: Pagesim: a novel link-based similarity measure for the World Wide Web. In: 2006 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2006, pp. 687–693. IEEE (2006)
16. Boongoen, T., Shen, Q., Price, C.: Disclosing false identity through hybrid link analysis. Artif. Intell. Law **18**(1), 77–102 (2010)
17. Su, P., Shang, C., Chen, T., Shen, Q.: Exploiting data reliability and fuzzy clustering for journal ranking. IEEE Trans. Fuzzy Syst. (2017)

Data Integration with Self-organising Neural Network Reveals Chemical Structure and Therapeutic Effects of Drug ATC Codes

Ken McGarry(✉) and Ennock Assamoha

School of Pharmacy and Pharmaceutical Sciences,
Faculty of Health Sciences and Wellbeing, University of Sunderland,
City Campus, Sunderland, UK
ken.mcgarry@sunderland.ac.uk

Abstract. Anatomical Therapeutic Codes (ATC) are a drug classification system which is extensively used in the field of drug development research. There are many drugs and medical compounds that as yet do not have ATC codes, it would be useful to have codes automatically assigned to them by computational methods. Our initial work involved building feedforward multi-layer perceptron models (MLP) but the classification accuracy was poor. To gain insights into the problem we used the Kohonen self-organizing neural network to visualize the relationship between the class labels and the independent variables. The information gained from the learned internal clusters gave a deeper insight into the mapping process. The ability to accurately predict ATC codes was unbalanced due to over and under representation of some ATC classes. Further difficulties arise because many drugs have several, quite different ATC codes because they have many therapeutic uses. We used chemical fingerprint data representing a drugs chemical structure and chemical activity variables. Evaluation metrics were computed, analysing the predictive performance of various self-organizing models.

Keywords: Kohonen · Prediction · ATC codes · Chemical fingerprints

1 Introduction

In this paper we describe how self organizing feature maps can provide classification labels for the so called drug therapeutic and anatomical codes (ATC). Most drugs have these codes allocated/annotated manually by experts and they provide useful information pertaining to drug specifications and characteristics. Researchers developing new drugs or repositioning existing drugs for novel applications can use the guidance provided by ATC codes to assist their efforts [9,10]. The ATC classification system classifies active drug ingredients into different levels, based on the drugs chemical properties, therapeutic properties, pharmacological properties and the organ/anatomical group which they target. ATC codes are highly prevalent in drug utilisation studies but they also provide a lot

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_6

of information on the drugs pharmacological, chemical and therapeutic properties. The prediction of drug ATC codes can thus further be utilised in the fields of drug discovery, adverse drug effect prediction and drug repositioning.

The Kohonen self-organizing feature map (SOM) [5] is probably the best known of the unsupervised neural network methods and has been used in many varied applications. It is particularly suited to discovering input values that are novel and for this reason has been used in industrial, medical and commercial applications. The SOM has the ability to easily visualize difficult to interpret data, this is because of its topology-preserving mapping of the input data to the output units. Allowing a reduction in the dimensionality of the input data, making it more suitable for analysis and therefore also contributing towards forming links between neural and symbolic representations [12]. Furthermore, symbolic rules can be extracted from the code-book vectors and weights providing an explanation of the cluster boundaries [8], it is this feature we use to help uncover the relationships between independent variables that describe ATC drug boundaries.

In Fig. 1, the overall operation of data download preprocessing and statistical model development is shown. Data was downloaded from drugbank and chembl repositories to obtain the ATC codes and chemical structures. The side-effect information from SIDER4 database was used to augment the relationships between chemical structure and drug activity. In previous work we have related side-effect information with drug action similarity for drug re-purposing opportunities [9].

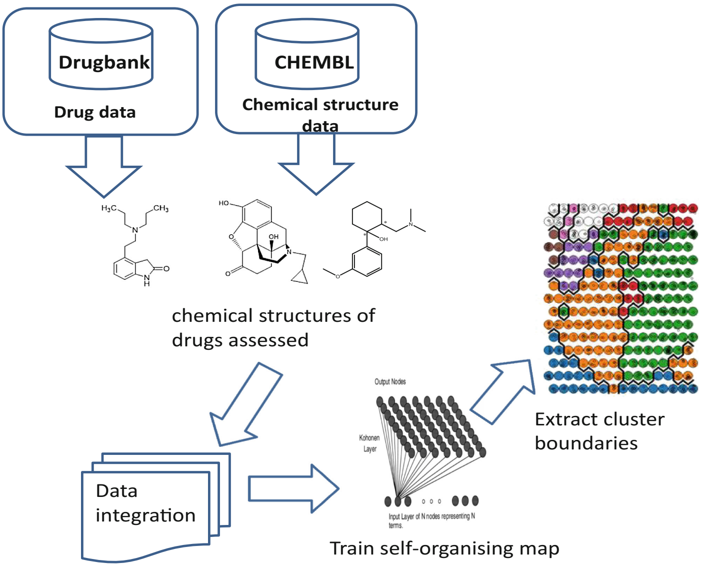

Drugbank
Drug data
CHEMBL
Chemical structure data
chemical structures of drugs assessed
Data integration
Output Nodes
Kohonen Layer
Input Layer of N nodes representing N terms.
Train self-organising map
Extract cluster boundaries

Fig. 1. System overview: data download, preprocessing and model building

1.1 Related Work

Dunkel et al. (2008) devised the *superpred* webserver, which constructs a structural fingerprint from a user defined molecule. The fingerprint was then compared to approximately 6000 drugs which had been enriched by approximate 7000 links to molecular drug targets; procured though text mining. However, Chen et al. extended this method to the lower levels of ATC classification. In a later study, Chen et al. introduced ontology information as well as chemical interaction and chemical structure information for the prediction of drug ATC codes [1,2]. Gurulingappa et al. combined the techniques of information extraction and machine learning in order to assign ATC codes to drugs [4]. The method had good predictive accuracy but was only tested on drugs which had an ATC classification code for the cardiovascular anatomical group.

Wang et al., utilized a Support Vector Machine (SVM) learning in a method named Netpred ATC [14]. The Netpred method utilised chemical structure information and drug target protein information The results from Wang et al.'s method was deemed to outperform the superpred method of Dunkel [3]. The method was able to successfully predict the ATC codes of unclassified and classified drugs. A web service called SPACE (Similarity-based Predictor of ATC CodE) was also developed to predict a range of ATC codes for a given drug compounds and their probability scores [17]. Other methods incorporate data from various sources such as gene ontology, chemical and compound structures [7].

The R code used to perform the analysis and the datasets we have used are freely available on GitHub from: https://github.com/kenmcgarry/ATC.

The remainder of this paper is structured as follows; section two describes our methods, indicating the types of data used and how we download and preprocessed it, along with the details of the self-organising feature map used to model this data, section three presents the results, section four provides the discussion and finally section five summarizes the conclusions and future work.

2 Methods

The ATC system is regulated by the World Health Organisation (WHO) and is the most renowned classification system in existence, used in a wide array of drug utilization studies. The ATC system consists of 5 grouping levels: with the detail of classification increasing as you progress from the top level (level 1) to through to the bottom level. The top level of the system classifies drugs based on the anatomical groups it targets. This level consists of 14 anatomical group which a drug may possibly target. The second level depicts the pharmacological/therapeutic sub group of a drug. The third level and the fourth level consist of the chemical/pharmacological/therapeutic subgroups. Moreover, the fifth level pertains to the chemical substance.

For example, the Drug Amlodipine has an ATC code of C08CA01 see Table 1. It is not necessary for a drug to only have one ATC code. A drug substance can have more than one ATC code assigned to it depending on whether it is available in different dosage formulations or strengths with therapeutic indication that are

Table 1. Example of ATC for the drug Amlodipine

Code	Description
C	Cardiovascular system (1st level, anatomical main group)
C08	Calcium channel blockers (2nd level, therapeutic subgroup)
C08C	Selective calcium channel blockers with mainly vascular effects (3rd level, pharmacological subgroup)
C08CA	Dihydropyridine derivatives (4th level, chemical subgroup)
C08CA01	Amlodipine (5th level, chemical substance)

clearly different from one another. This can be seen in the drug acetylsalicylic acid. Acetylsalicylic acid has three different ATC codes: B01AC06 for when it is used as a platelet aggregation inhibitor, A01AD05 for when it is used for local oral treatment and N02BA01 for when it is used as an analgesic and antipyretic. The level 1 code for all categories is shown in Table 2.

Table 2. Level 1 ATC codes

Code	Description	Code	Description
A	Alimentary tract and metabolism	L	Antineoplastic and immunomodulating
B	Blood and blood forming organs	M	Musculo-skeletal system
C	Cardiovascular system	N	Nervous system
D	Dermatologicals	P	Antiparasitic products, insecticides and repellents
G	Genito-urinary system and sex hormones	R	Respiratory system
H	Systemic hormonal preparations	S	Sensory organs
J	Anti-infectives for systemic use	V	Various

The basic Kohonen SOM has a simple 2-layer architecture. Since its initial introduction by Kohonen several improvements and variations have been made to the training algorithm. The SOM consists of two layers of neurons, the input and output layers. The input layer presents the input data patterns to the output layer and is fully interconnected. The output layer is usually organised as a 2-dimensional array of units which have lateral connections to several neighbouring neurons. The architecture is shown in Fig. 2.

The objective is to build a self-organising feature map in a two stage process of training the network and then passing a vector of test data through the network and observing the active neurons. A well trained network will have different neurons respond to specific input patterns. During training the network requires exposure to patterns which will modify the inter-neuron connections during the learning phase. Competitive learning is ensures that the best matching unit (neuron) will be activated (competes) in preference to the other units in the network. In each training step, the algorithm will calculate the changes to the synapses for every new input, D_j for each neuron:

The competitive learning process is presented in Eq. 1 and the best matching neuron is derived from Eq. 2.

$$D_j = \sum_{i=0}^{p} ||I_i - W_{ij}|| \tag{1}$$

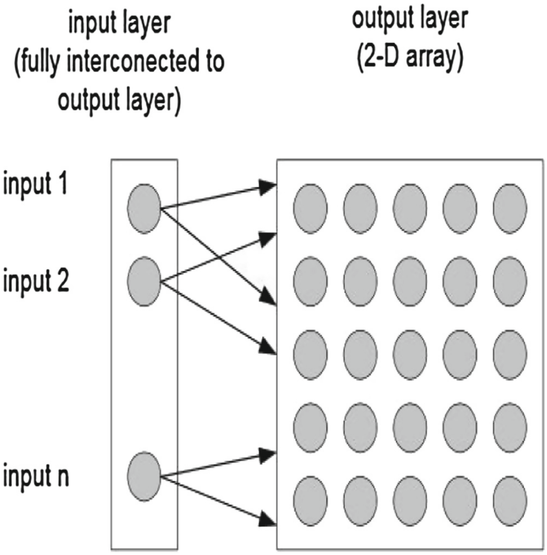


Fig. 2. Architecture of self-organising map

where: D_j is the distances for each neuron, I_i is the current input vector and W_{ij} is the weight vector.

We then select the Best Match Unit (BMU) and update the weight vectors of the map according to Eq. 2.

$$W_i(t+1) = W_i + h_{ci}(t) * (I(t) - W_i(t)) \tag{2}$$

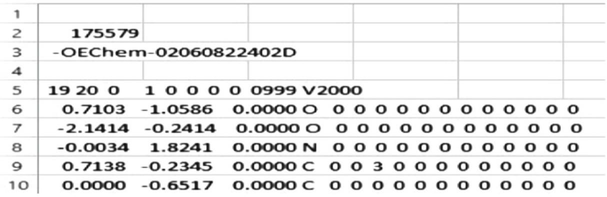
where t denotes the time and h_{ci} is the neighborhood kernel around the BMU. W_i is the weights attached to that unit.

However, key to understanding the kohonen network is the so called unified or U-matrix decomposition method. This enables the cluster boundaries to be made visible to the eye [13]. The matrix output uses relative distance between reference vectors to find cluster boundaries. Given an M×N lattice, the Euclidean distances associated with the reference vectors of the adjacent cells, such as M_{i-1}, M_{i+1} are summed, $M_{i,j}$ is designated as M adjacent (i, j), and d represents the Euclidean distance, then, according to U is now plotted using Eq. 3.

$$U_{(ij)} = \sum d(M_{adjacent}(i,j), M_{i,j})) \tag{3}$$

When the matrix is plotted, the cluster boundaries are generally dark colours while the clusters form lighter coloured spaces in between.

CHemBL is an important database containing chemical compounds usually represented as string data called the SMILES (Simplified Molecular Input Line Entry System) format, various algorithms have been developed that can generate numbers describing the compound [16]. The majority of our chemical data was downloaded and stored in SDF format, these are plain text files with a specific internal format. The format is extensible, a single file can contain a single chemical structure or millions of structures. The SDF text files have a fairly straightforward structure although can be inefficient in memory storage since a lot of whitespace and unnecessary characters are used to represent the chemical structures, see Fig. 3. This redundancy dates in part from earlier legacy file systems, programming languages and parsing techniques.



```
1
2      175579
3    -OEChem-02060822402D
4
5    19 20  0     1  0  0  0  0  0999 V2000
6       0.7103  -1.0586   0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
7      -2.1414  -0.2414   0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
8      -0.0034   1.8241   0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
9       0.7138  -0.2345   0.0000 C   0  0  3  0  0  0  0  0  0  0  0  0
10      0.0000  -0.6517   0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
```

Fig. 3. SDF chemical structure file, showing first few lines for the tramadol drug. The first line is the drug id, second line refers to the software package that created it. The next few lines describe the properties which state the number of atoms and bonds, each atom and bond on a separate line.

In Algorithm 1 the training data generation, processing and neural network processing is clarified, as a series of steps.

Algorithm 1. Data generation, processing and Kohonen training

```
procedure TrainKohonen($CheMBL$ chemical structures, ATC codes from $DrugBank$)
    do initialize
    $cStruc \leftarrow$ only get drugs with chemical structures[$CheMBL$]
    $aCodes \leftarrow$ only get drugs with ATC codes[$DrugBank$]
    $numStruc \leftarrow$ length[$cStruc$]          ▷ How many useful chemicals do we have?
    $Kohonen \leftarrow$ [N $x$ M]                  ▷ Setup small $5 \times 5$ matrix of nodes
    end initialize

    for $i \leq numStruc$ do                        ▷ process every drug with a chemical structure
        $FP_i \leftarrow Fingerprint(cStruc)$        ▷ Convert from SDF to binary fingerprints
        $FP_i \leftarrow Fingerprint(aCodes)$        ▷ Attach ATC code as class label to binary fingerprints
        $FP_{train} \leftarrow randomsample(FP_i, 80)$   ▷ split data 80/20% train/test
        $FP_{test} \leftarrow randomsample(FP_i, 20)$    ▷ split data 80/20% train/test
    end for
    repeat
        Train Kohonen [$FP_{train}$]
        Test Kohonen [$FP_{test}$]
        Modify Kohonen architecture, [$NxM$]         ▷ Manual intervention, increase until $15 \times 15$
    until $Kohonenerror \leq 0.01$                  ▷ stop training when error reaches cutoff point
    $GCs \leftarrow CalcNetworkStatistics(Cw)$       ▷ call
    inspect Mapping                                 ▷ visually inspect patterns to neurons
    inspect Umatrix                                 ▷ visually inspect umatrix
end procedure
```

The chemical database was then imported into the system using SDF format data. The database consisted of 7,759 drugs with 12 variables relating to chemical structure. Chemical fingerprints were created from these structures - each fingerprint is a binary matrix with 1 = structure present or 0 = structure absent. A random sample of 80% was taken from the dataset of 1,334 drugs and 20% for test, with chemical fingerprints assigned to each drug (representing the drug chemical structure). The training dataset consisted of 1,067 drugs with chemical fingerprints. This was presented to a Kohonen network with and architecture of 15 × 15 nodes. We use the Kohonen in a semi-supervised way, i.e. we have class labels (ATC codes).

We implemented the system using the R language with the RStudio programming environment, on an Intel Xenon 64-bit CPU, using dual processors (3.2 GHz) with six cores, and 128 GB of RAM. R is primarily a statistical data analysis package but is gaining popularity for various scientific programming applications and is very extendable using packages written by other researchers [11]. We used the following R packages: Kohonen [15]. Since it is an interpreted language, R is generally quite slow compared with a compiled language. However, we used the MicroSoft R Open system (https://mran.microsoft.com/) because it is optimized to take advantage of processor cores and the majority of its mathematics/matrix operations are rewritten in C++ to speed up operation. It is fully compatible with the oringinal CRAN version of R.

3 Results

The DrugBank database was integrated into our system because it contains the majority of drugs that are currently prescribed, or have been withdrawn or are at the clinical trial stage. This resource is widely used by those developing drugs, chemists, pharmacologists and others involved in pharmaceutics research [6]. Every drug is listed with its main targets, known off-targets along with chemical structure and other important characteristics.

Figure 4 contains the plot showing the training progress, during training, the codebook vectors are becoming more and more similar to the closest objects in the dataset.

When the test data is passed to the self-organising map we obtain the following confusion matrix based on the success or otherwise. It is evident that the ATC classes A, J, M and N consistently recorded high class accuracy, recall, precision and f1 scores for the self-organising map. The confusion matrix relating the accuracy of the various ATC codes is displayed in Fig. 5.

The correct classifications for the test data are on the upper left to bottom right diagonal, any misclassification's are located off-diagonal and reveal which class they were misclassified as. For example the ATC code 'A' has 12 correctlty identified test cases but one sample is misclassified as 'C', five misclassified as 'D' etc.

In Fig. 6 we reproduce the effects of the influence of the independent variables on the neurons. Here, for simplicity a 5 × 5 grid is extracted from the 15 × 15 grid of neurons. Each of the 12 independent variables will be colour coded and similar to a pie-chart the magnitude and orientation of the slice indicates its influence on the neuron.

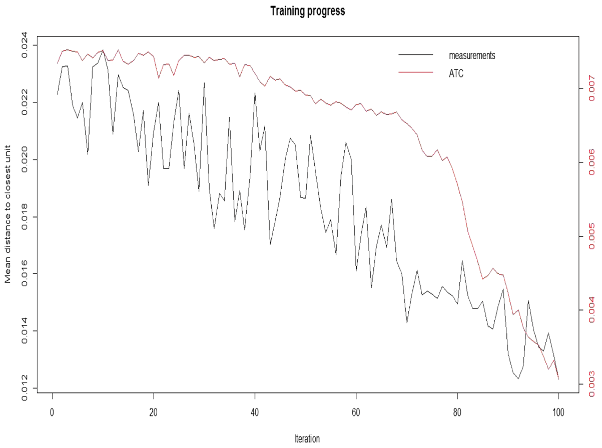
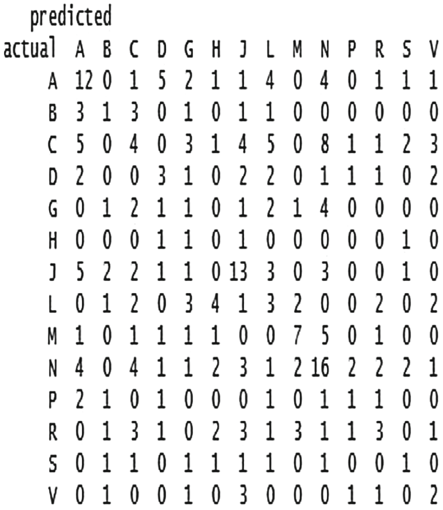


Fig. 4. Training progress of kohonen self-organizing map



```
   predicted
actual A  B  C  D  G  H  J  L  M  N  P  R  S  V
     A 12 0  1  5  2  1  1  4  0  4  0  1  1  1
     B 3  1  3  0  1  0  1  1  0  0  0  0  0  0
     C 5  0  4  0  3  1  4  5  0  8  1  1  2  3
     D 2  0  0  3  1  0  2  2  0  1  1  1  0  2
     G 0  1  2  1  1  0  1  2  1  4  0  0  0  0
     H 0  0  0  1  1  0  1  0  0  0  0  0  1  0
     J 5  2  2  1  1  0 13  3  0  3  0  0  1  0
     L 0  1  2  0  3  4  1  3  2  0  0  2  0  2
     M 1  0  1  1  1  1  0  0  7  5  0  1  0  0
     N 4  0  4  1  1  2  3  1  2 16  2  2  2  1
     P 2  1  0  1  0  0  0  1  0  1  1  1  0  0
     R 0  1  3  1  0  2  3  1  3  1  1  3  0  1
     S 0  1  1  0  1  1  1  1  0  1  0  0  1  0
     V 0  1  0  0  1  0  3  0  0  0  1  1  0  2
```

Fig. 5. Confusion matrix for the 14 ATC codes

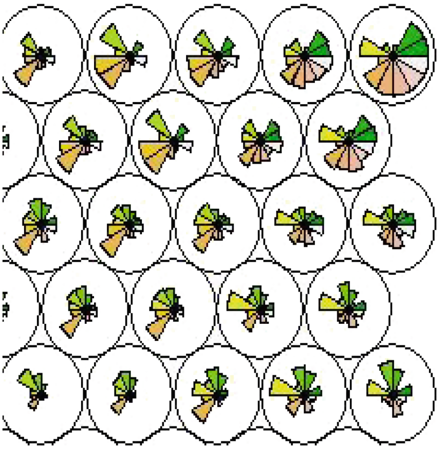


Fig. 6. Relative weights and effects of independent variables

The Jchem variables identified in Fig. 6 contained information on: acceptor count, average polarizability, donor count, ALOGPS LogP, Jchem LogP, ALOGPS LogS, number of rings, physiological charge, strongest basic pka, polar surface area, refractivity and rotatable bond count.

Plotting the training data to the 15 × 15 grid of neurons reveals why a classification algorithm would have difficulty. In Fig. 8 we can see that only a few neurons have unique ATC codes mapped to them. Most units have 2–3 sometimes more ATC codes, ideally for 100% accuracy would would need to have each ATC code mapped to its own unique neuron. There are also several neurons that

JCHEM_ACCEPTOR_COUNT
JCHEM_AVERAGE_POLARIZABILITY
JCHEM_DONOR_COUNT
ALOGPS_LOGP
JCHEM_LOGP
JCHEM_NUMBER_OF_RINGS
JCHEM_PHYSIOLOGICAL_CHARGE
JCHEM_PKA_STRONGEST_BASIC
JCHEM_POLAR_SURFACE_AREA
JCHEM_REFRACTIVITY

Fig. 7. Colour scheme for identifying independent variables shown in Fig. 6

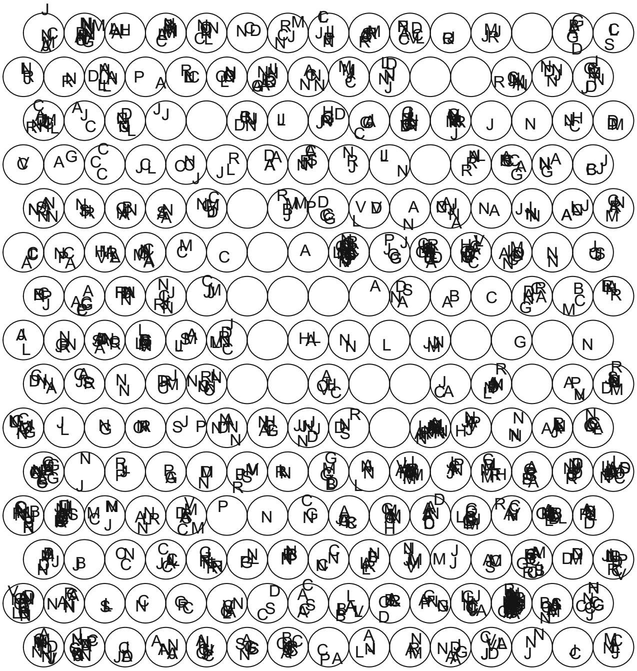


Fig. 8. Mapping ATC codes to 15 × 15 self-organising map

could not get codes mapped to them. These seem to form a border or boundary isolating the upper right part of Fig. 8. The locations of the circles indicate the neurons to which the samples have been mapped. The relationship between Figs. 8 and 9 involves the mapping of particular input vectors to specific neurons thus enabling the cluster boundaries to be made visible (Fig. 7).

Despite their excellent capability of visualization, SOMs cannot provide a full explanation of their structure and composition without further detailed analysis. One method towards filling this gap is the unified distance matrix or U-matrix technique of Ultsch [12]. The U-matrix technique calculates the weighted sum of all Euclidean distances between the weight vectors for all output neurons. The resulting values can be used to interpret the clusters created by the SOM. The rather confusing picture presented by Fig. 8 indicates that several neurons have many different ATC codes assigned to them (in fact the preference is for one ATC code for each neuron). The unmatrix shown in Fig. 9 indicates that several large boundaries exist.

Although, primarily used for situations where there are no class labels, and hence the natural structure of the data is important - the self-organizing feature map can also be used where class labels exist and semi-supervised operation is useful. This will provide more information to explain the key features of a dataset and a limited explanation of why the Kohonen organized the data they way it did [8].

The U-matrix in Fig. 9 is colour coded for each main cluster and has the umatrix boundary drawn to emphasize these. Any neuron near a class boundary can be expected to have higher average distances to their neighbours, indicating structural consistencies in the data. Interpreting the u-matrix diagram with Fig. 8 which highlights the mapping of the ATC labels has produced an area of 'no mans land' denoted by the empty neurons not allocated to a code appears to run counter to the u-matrix in the sense that it does not seem to be part of a boundary. It is probably a sign that the 15×15 map is too large for the available data but changing the map to smaller grids did not improve the training or classification issues. The majority of the boundaries can be explained by the simple fact that although the drugs have different ATC classifications, their

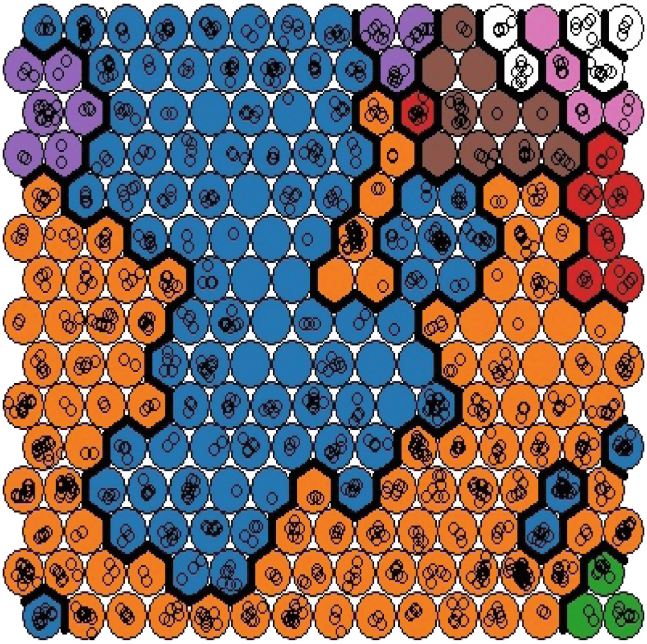


Fig. 9. Umatrix revealing cluster boundaries

chemical structure is very similar in many cases. In fact only slight changes to chemical structure are required for very different pharmacological properties to be exhibited by a drug. Therefore, some neurons think they doing a good job of identifying drugs based on the input vectors, however since we have access to the ATC labels we know this is not quite the case.

4 Discussion

The training of the Kohonen map clearly showed a decrease in the mean distance to the closest unit, as should be the case in a successfully trained Kohonen neural network. Even so, an improvement in the training process can be made as even after 100 iterations, it could be seen that the mean distance to the closest unit was still decreasing. Thereby indicating that the training sequence may not have been fully complete and that further iterations were needed. The optimum number of iterations needed can be known when the mean distance to the closest unit becomes relatively stable (flat) and begins to merely fine tune the value. The mean distances of objects that were mapped to a unit to the codebook vectors (of the units these objects were mapped to) was generally very low across all units; with only one unit/neuron displaying a mean distance above 1.0. In most cases, this meant that the objects were well represented by the codebook vectors and thus highlights the success of the Kohonen neural network method. Future work must consider adding further information to the chemical data to resolve the mapping issues.

5 Conclusions

The novel contribution of this work relates to the explanatory ability of the Kohonen network to reveal the internal structure of the clusters and input to output class label mapping. Future work will address integrating other sources of drug/chemical information to improve accuracy. The issue of drugs with multiple ATC codes in different therapeutic areas also needs to be resolved as it is a source of bias and class fuzziness.

References

1. Chen, F., Jiang, Z.: Prediction of drug's anatomical therapeutic chemical (ATC) code by integrating drug-domain data. J. Biomed. Inf. **58**, 80–88 (2015)
2. Cheng, X., Zhao, S., Xiao, X., Chou, K.: iATC-mISF: a multi-label classifier for predicting the classes of anatomical therapeutic codes. Bioinformatics **33**(3), 341–346 (2016)
3. Dunkel, M., Gunther, S., Ahmed, J., Wittig, B.: Superpred: drug classification and target prediction. Nucleic Acids Res. **36**, W55–W59 (2008)
4. Gurulingappa, H., Kolarik, C., Hofmann-Apitius, M., Fluck, J.: Concept-based semi-automatic classification of drugs. J. Chem. Inf. Model. **49**(8), 1986–1992 (2009)

5. Kohonen, T., Oja, E., Simula, O., Visa, A., Kangas, J.: Engineering applications of the self-organizing map. Proc. IEEE **84**(10), 1358–1383 (1996)
6. Law, V., Knox, C., et al.: Drugbank 4.0: shedding new light on drug metabolism. Nucleic Acids Res. **42**, D1091–D1097 (2014)
7. Liu, Z., Guo, F., Gu, J., Wang, Y., Li, Y., Wang, D., Li, D., He, F.: Similarity-based prediction for anatomical therapeutic chemical classification of drugs by integrating multiple data sources. Bioinformatics **31**(11), 1788–1795 (2015)
8. Malone, J., McGarry, K., Bowerman, C., Wermter, S.: Rule extraction from kohonen neural networks. Neural Comput. Appl. J. **15**(1), 9–17 (2006)
9. McGarry, K., Daniel, U.: Data mining open source databases for drug repositioning using graph based techniques. Drug Discov. World **16**(1), 64–71 (2015)
10. McGarry, K., Slater, N., Amaning, A.: Identifying candidate drugs for repositioning by graph based modeling techniques based on drug side-effects. In: The 15th UK Workshop on Computational Intelligence, UKCI-2015. University of Exeter, UK (7th-9th September 2015)
11. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2015). https://www.R-project.org/
12. Ultsch, A., Korus, D.: Automatic acquisition of symbolic knowledge from subsymbolic neural nets. In: Proceedings of the 3rd European Conference on Intelligent Techniques and Soft Computing, pp. 326–331 (1995)
13. Ultsch, A., Mantyk, R., Halmans, G.: Connectionist knowledge acquisition tool: CONKAT. In: Hand, J. (ed.) Artificial Intelligence Frontiers in Statistics: AI and statistics III, pp. 256–263. Chapman and Hall, London (1993)
14. Wang, Y., Chen, S., Deng, N., Wang, Y.: Network predicting drug's anatomical therapeutic chemical code. Bioinformatics **29**(10), 1317–1324 (2013)
15. Wehrens, R., Buydens, L.: Self and super-organising maps in r: the Kohonen package. J. Stat. Softw. **21**(5) (2007). http://www.jstatsoft.org/v21/i05
16. Weininger, D.: Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. J. Chem. Inf. Model. **28**(1), 316 (1988)
17. Wu, L., Liu, N., Wang, Y., Fan, X.: Relating anatomical therapeutic indications by the ensemble similarity of drug sets. J. Chem. Inf. Model. **53**, 2154–2160 (2013)

A Modified Approach to Inferring Animal Social Networks from Spatiotemporal Data Streams

Pu Zhang and Qiang Shen(✉)

Department of Computer Science, Institute of Mathematics,
Physics and Computer Science, Aberystwyth University, Aberystwyth, UK
{puz,qqs}@aber.ac.uk

Abstract. Animal social networks offer an important research mechanism for animal behaviour analysis. *Inferring social network structures in ecological systems from spatiotemporal data streams* [1] presents a procedure to build such networks based on animal's foraging process data which consists of time and location records. The method clusters the individuals into different gathering events and links up the individuals that appear in the same events, and subsequently filters coincident links. However, the original model does not perform well in many aspects, including time and space complexity and not-unique coincident link filtering threshold. To modify this method, fuzzy c-means is employed in this work to cluster all links into two groups, strong links or weak links. The work presented here also experimentally compares the performance of the proposed modification against the original method, demonstrating the efficacy of the modified version.

Keywords: Animal social networks · Coincident links · Spatiotemporal data · Fuzzy c-means

1 Introduction

Network analysis owns a long history, the original mathematical networks could be tracked back for centuries. Nowadays, network analysis has been increasingly applied to solving more and more problems such as Web link examination and animal behaviors analysis. For animal behaviors analysis, animal social networks have three distinct properties [2]:

1. Supporting the analysis of complex networks whose individuals have many features to consider.
2. Permitting the exploration of network structures at different levels such as individuals, dyad, group and population, enabling the generation of larger networks from pairwise interactions [2].
3. Helping discovery of the influence of individual behaviors on a population and that of the fitness of population on individuals [2].

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_7

The work on *inferring social network structures in ecological systems from spatiotemporal data streams* [1] has presented an approach to extracting such an animal social network from an animal tracking dataset that consists of time and location records (which has been widely applied in animal behavior analysis [3,4]).

The procedure is based on a hypothesis named *the Gambit of Group* [5], indicating that there are social connections between animals who spend more time in the same location with each other. The procedure consists of three steps. First, the records are clustered into different 'gathering events', then the animals (where only birds are concerned) which appear in the same gathering events are linked as friends forming a social network. In such a social network, there are many birds appearing in the same events by chance. Therefore, a so-called null model is designed to filter those coincident links in the third step. It is based on the assumption that all individuals have no relationship with each other and appear in gathering events randomly. All links generated in this case will be regarded as coincident links. Thus, the links with weak strengths over these coincident links can be filtered while retaining just the ones with greater strengths.

There are two significant limitations in the null model: that both the time complexity and space complexity are generally very large, and that the threshold used to perform filtering is not unique. To address these limitations, fuzzy c-means [6] is applied to modify the coincident link filter. It clusters links into two groups, strong links and weak links, where weak links are regarded coincident and strong links are used to build the final network. According to the experimental comparison, the filter based on the use of fuzzy c-means has led to better results than the null model, while having lower time complexity.

The rest of this paper is structured as follows. Section 2 reviews the generation process of animal social networks. Section 3 describes two coincident link filtering methods, the conventional null model and the proposed one that utilises fuzzy c-means clustering. Section 4 details the setting of the experiments carried out and the results of comparative experimental evaluations. Finally, Sect. 5 concludes this paper with future research pointed out.

2 Network Generation

This section introduces the process of generating animal social networks between individuals from spatiotemporal data set. The presumption of links are based on the previously mentioned notion of *the Gambit of Group (GoG)* [5].

Presumption 1. *If individuals arrive in the same location at close time points, then they may be clustered into the same group. This process is called 'gathering event'. If two individuals appear in the same gathering event, then they can be linked together.*

An implication of this presumption is that locations are treated as independent of one another. This means that networks may be built for each location of interest separately and then the emerging network can be integrated over all the locations to obtain the final network. In this regard, a social network is inferred exclusively by time records in the data stream.

To implement the above presumption, the following techniques have been developed, which will be detailed below:

1. Selection of arrival time records from the dataset.
2. Clustering arrival time records to different gathering events.
3. Link-up of individuals if appearing in a common gathering event.

2.1 Arrival Time Record Selection

In general, a spatiotemporal data stream may include a large number of records. However, only arrival time records are valuable for the network generation procedure, with the concept of arrival time as given below:

Presumption 2. *If an individual is not recorded over a certain time period Δt, the next recorded time of the individual is regarded as the arrival time.*

To select arrival time from a large dataset, the time period Δt should be settled first. The size of Δt will influence the size of arrival time records and the accuracy of the subsequent gathering events. To obtain an appropriate Δt, a distribution of arrival time records is built up. A trial and error process is run to increase Δt until it clearly changes the distribution, and the point at which such a change takes place is returned as the maximum value of Δt. With the use of the maximum Δt, on the one hand, the accuracy of the process can be maintained, and on the other hand, the size of dataset is reduced which, in turn, also helps reduce the running time and storage space.

2.2 Gathering Events Clustering

Arrival time records are typically concentrated on several special areas as exemplified in Fig. 1. These groups are called ‘gathering events’.

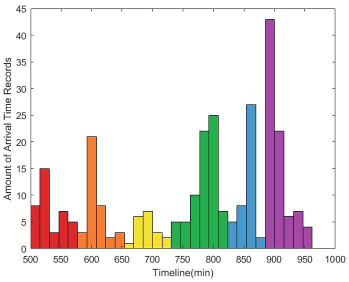


Fig. 1. An instance of arrival time records; as reflected by the distribution, records are concentrated on 6 groups that are termed ‘gathering events’.

To cluster the records into different gathering events, Gaussian mixture model (GMM) [7], a classical clustering algorithm is applied in the original approach. The result is described by a record-to-event matrix RE $\in R^{Z \times K}$, where Z is the number of arrival time records and K is the number of gathering events. Each record is only clustered into one gathering event while a single gathering event may include many records.

2.3 Link Generation

According to Presumption 2, to identify links between individuals, the relationship between individuals and gathering events needs to be generated first. Since arrival time records have been clustered into gathering events (in the last step) while the identities of individuals have been included in the records, a connection network among individuals, records and gathering events can be built up. Figure 2 shows a such network.

In a such network, most individuals may be involved in more than one record and the records may belong to different gathering events. Thus, many individuals may attend at more than one gathering event. Based on that, the connections between individuals and events may be generated as shown in Fig. 3.

The result is described by an individual-to-event matrix IE $\in R^{N \times K}$, where N represents the number of individuals. Elements in this matrix represent the number of records an individual attends at a certain gathering event.

However, considering that different individuals may involve in a different number of records, the elements need to be normalized with reference to the proportion of each event. These new elements are each called a 'preference', and this leads to a new individual-to-preference (IP) matrix. The elements in the matrix IP are:

$$p_{ij} = \frac{r_{ij}}{\Sigma_{j=1}^{K} r_{ij}} \tag{1}$$

where i stands for an individual an individual, j for a gathering event and $r_{ij} \in$ IE.

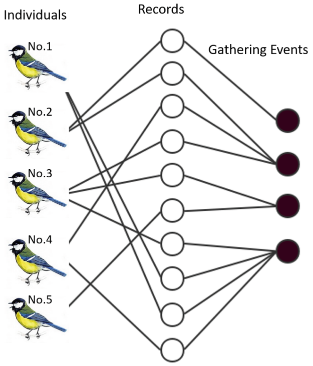


Fig. 2. Connection network consisting of two layers of nodes, with the left layer representing the records each individual has and the right the gathering event each record belongs to.

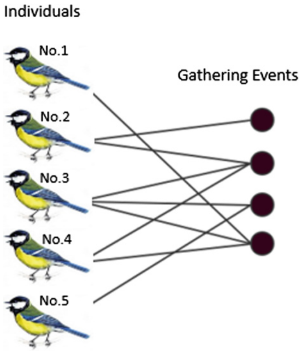


Fig. 3. Relationship between individuals and gathering events, representing which gathering events an individual appears in.

Table 1. Strength of links

	1	2	3	4	5
1	/	0	0.33	0.5	0
2	0	/	0.33	0.5	0
3	0.33	0.33	/	0.67	0.33
4	0.5	0.5	0.67	/	0.33
5	0	0	0.33	0.33	/

The strength of a link or relationship is determined by the preference similarity. In particular, the link between individual i and j is computed by the summation of the minimum preferences of individual i and j in the K gathering events:

$$a_{ij} = \Sigma_{k=1}^{K} \min(p_{ik}, p_{jk}) \tag{2}$$

Resulting strengths within the network can be described by an adjacency matrix NET $\in R^{N \times N}$. Table 1 shows such a matrix generated from the data given in Fig. 3.

3 Coincident Link Filtering

Coincident links are a problem caused by the presumptions made previously. It means that the linked individuals may appear together only by chance, or that the links with a low strength may exist in matrix NET. Setting a threshold helps define normal links and filter certain coincident links. That is, links with a strength larger than a given threshold will be retained to become a final link and those with a strength less than the threshold will be regarded as spurious ones and therefore, filtered out. Coincident filtering makes the structure of a generated network simpler, thereby making any subsequent reasoning computation simpler also.

3.1 Null Model

A null model is one that is derived on the basis of the so-called null hypothesis [8]. It assumes that all individuals have no relationship with each other and the foraging process is totally random. All links generated in this case should then be regarded as coincident links. For any element in matrix NET, whose strength is larger than the threshold are regarded as final links.

Null model generation consists of two procedures: random process simulation and threshold selection. The random process simulation shuffles each row's elements in the IP matrix, generating a new matrix IP', which breaks up the original connections. Then, a new link matrix NET' can be generated (by the same process as described in Sect. 2.3). In order to eliminate the coincident result in the shuffling process, this generation process repeats N times to obtain N strengths for each link. In applications of this technique, the number of N needs to be decided on the basis of trial and error.

For the threshold selection procedure, statistical methods are applied to identify an appropriate threshold; all links with a strength less than this threshold will be regarded as spurious links to a given significance level α. There are two probability-based methods that may be applied for this: empirical distribution and normal distribution. Empirical distribution is cumulative, measuring the proportion of those data objects which are less than or equal to a specific value *t*. For the dataset $X = x_1, x_2, \ldots, x_n$, the distribution function *F(t)* is defined by [9]

$$F(t) = \frac{1}{n} \Sigma_{i=1}^{n} 1(x_i \leq t) \tag{3}$$

From this, the *threshold* for empirical distribution simply satisfies the following:

$$F(threshold) \geq \alpha. \tag{4}$$

Another method is based on normal distribution or Gaussian distribution, which is defined by

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{5}$$

where μ is the expectation and σ the standard deviation.

For such a distribution, the threshold is set as $(\mu + n\sigma)$, where $(\mu \pm n\sigma)$ is associated with a given significance level α. Once the α is determined, the threshold is determined as well. From which the spurious links can be filtered out in a straightforward manner, by comparing the elements in the link strength matrix against it.

3.2 Problems with Null Model

Null model provides an approach to filter coincident links. However, there are two limitations in using it:

1. **Time complexity is very large.** This model requires repeatedly shuffling T times (where T is typically a large number) to obtain a set of strengths for each links. Empirically, T is larger than 50 to make the thresholds stable even for a moderate-sized problem. Thus, according to the description of the null model, the time complexity is $O(NT^2)$, where N is the number of links.
2. **Thresholds maybe different for different links.** Differences between individuals may require different thresholds to be used, but this problem has been addressed previously (in Sect. 2.3) by normalizing the IE matrix to the IP matrix. Nevertheless, the model is hard to extend when multiple thresholds are employed. For any new link discovered, its threshold has to be calculated.

Therefore, in order to modify spurious link filtering and to exploit the information contained within the links, clustering is applied to group the links into two distinct categories: strong links and weak links. The particular clustering algorithm employed in this work is fuzzy c-means, which is outlined below.

3.3 Fuzzy C-Means Filter

The purpose of coincident link filtering is to reduce spurious links. Through dividing links into two different groups, strong and weak, those weak links can be filtered while retaining the strong ones. Unlike conventional clustering algorithms which only allow one cluster for each instance, fuzzy c-means allows for a instance to belong to different clusters with a different membership degree each.

Fuzzy c-means is originally developed in [10] and subsequently improved in [11]. It clusters data based on computing object distance to each emerging cluster centre. However, instead of using a boolean distance metric directly as k-means [13], it calculates the membership of an object to decide on the clustering result. A minimum distance represents maximum membership. Here, the membership M is defined by

$$M = (\Sigma_{j=1}^{C}(\frac{\|x_k - v_i\|}{\|x_k - v_j\|})^n)-1 \tag{6}$$

and the objective function is defined by [6]

$$J_m(U,V) = \Sigma_{k=1}^{N}\Sigma_{i=1}^{C}(u_{ik})^m\|x_k - v_i\|^2 \tag{7}$$

where x_k is an object, m is the weighting exponent that controls the weight of each component, C is the number of centres, N is the number of objects, V is the set of centres and $v_i \in V$, U is the set of membership and $u_{ik} \in U$ representing the membership of the object k belonging to the centre i, and $\|x_k - v_i\|$ represents the similarity between the object k and the centre i.

To minimize the objective function J_m, fuzzy c-means updates the membership function u_{ik} and the centre v_i iteratively by

$$v_i = \Sigma_{k=1}^{N}(u_{ik})^m x_k / \Sigma_{k=1}^{N}(u_{ik})^m \tag{8}$$

$$u_{ik} = (\Sigma_{j=1}^{C}(\frac{\|x_k - v_i\|}{\|x_k - v_j\|})^{2/m-1})^{-1} \tag{9}$$

The procedure of fuzzy c-means algorithm can be summarized in Algorithm 1.

Algorithm 1. Fuzzy C-Means

```
1 Initialize $u_{ik} \in U^{(0)}$ randomly.
2 Set max iteration number L, termination condition $\varepsilon$.
3 Update centres $v_i \in V^{(k)}$ by $U^{(k)}$ and Eq. (8).
4 Update membership $u_{ik} \in U^{(k+1)}$ by $V^{(k)}$ and Eq. (9).
5 If $\|U^{(k+1)}$ - $U^{(k)}\| < \varepsilon$ or k+1 = L, then stop; otherwise, set $U^{(k)} = U^{(k+1)}$
  and return to step 3.
```

Comparing with the null mode, in general, the time complexity of fuzzy c-means is $O(NKT)$, where N is the number of links, K is the number of link clusters and T is the number of iterations. Herein, K is set to 2 since links are grouped into two categories. However, the complexity of null model is $O(NT^2)$ as mentioned previously. Since the value of T for both methods will increase along with the increase in the size of the dataset. Therefore, fuzzy c-means has lower time complexity, and the real running time between these two methods will be experimentally compared later.

4 Experimental Evaluation

The experiments reported herein have two purposes. The first is to compare the performance of using fuzzy c-means and Gaussian mixture model in the link generation process, and the second is to compare the performance of using null model and fuzzy c-means in coincident link filtering.

To compare the performance, ground truth has to be provided. However, there is no information regarding the ground truth available for the datasets used in the literature on animal social networks that are ideally used to facilitate comparative studies. Blood relation is employed to evaluate the result in [1], but only blood relation can not represent all types of relationship and is not sufficient to act as the ground truth. Therefore, the present investigation is based on the use of different datasets whose ground truth is given while the underlying inference process remains the same.

Based on the above, datasets which are suitable for both classification and clustering problem are employed, the results of the clustering process is evaluated by comparing the underlying class labels. In particular, for coincident link filtering, the ground truth for the dataset is utilised such that if two linked individuals have the same label, then this link is regarded as positive (i.e., the link is retained in the emerging network); otherwise, the link is negative (i.e., the links is removed). In this case, the number of positive links N can be calculated as following:

$$N = \Sigma_{i=1}^{k} \frac{1}{2} n_i (n_i - 1) \tag{10}$$

where k is the number of classes and n_i is the number of objects belonging to class i.

The construction process of ground truth is similar with the inference process of the original work and the structure of employed datasets is also similar to that of the spatiotemporal datasets. The only difference between these two datasets is that unlike the dataset used in original work [1] where individuals will have similar memberships to multiple clusters, the individuals in the datasets used for evaluation can only have one membership close to 1 while others close to 0. Thus, the distribution of link strengths will be concentrated on 0 and 1. It can be regarded as a special condition of the original work when each individual only belongs to one cluster. Note that the empirical distribution method in the null model is not suitable for these dataset.

For completeness, experiments are also carried out using the original dataset that was adopted in the original paper on animal social networks [2].

4.1 Seed Data Set and Experimental Setup

The Seeds Data Set [12] is employed here to evaluate the performance of the improved work. It consists of 210 instances, 7 features and 3 classes. All features are numerical and each class includes 70 instances. According to Eq. (10), there are 7245 links that are positive and that form the ground truth for testing. Apart from the ground truth, there are a number of parameters that need to be set up in order to conduct the experiments. In particular, for the generation of gathering events, according to the given class labels, the number of events K is set to 3. For coincident links filtering, in the original null model, the number of times of shuffling T is empirically set to 50, and 68% is employed as the significant level α (with normal distribution selected for threshold determination). In the modified method, the number of link clusters K is set to either 2 or 3 to facilitate comparison, where $K = 2$ represents two link clustering: strong and weak links, while $K = 3$ represents three link clustering: strong, general and weak links.

4.2 Results on Seeds Data Set

The original and modified approaches are applied to generate and filter links from seeds dataset and they will be compared with regard to six aspects: the amount of links remain in each method, the amount of true positive links in these remain links, precision, recall, the F1-score and the running time of filtering. F1-score is a combination of precision and recall and is employed to indicate the overall performance. The original links without involving any filtering process are also shown here as a control group. The results of experiments are displayed in Table 2. It can be seen from this table that the coincident filtering process can successfully reduce the total number of links while retaining most valuable links, thereby helping decrease storage space and retrieve time significantly.

Table 2. Results of original and modified methods

Cluster	FCM		GMM	All links without filter
Coincidence	FCM (k = 2)	FCM (k = 3)	Null model	
Link amount	8663	6448	7269	21945
True positive	**6431**	5407	5627	7245
Precision	0.7426	**0.8384**	0.7741	0.3301
Recall	**0.8879**	0.7463	0.7767	1.0000
F1-Score	**0.8089**	0.7897	0.7754	0.4964
Running time	0.4340	**0.3600**	14.7685	/

It can also be seen from Table 2 that fuzzy c-means based filters have a larger F1-score than GMM based method. This indicates that the modified approach performs better than the original.

In particular, FCM ($k = 2$) offers the best performance, retaining the most positive links with the highest recall rate. However, due to the largest amount of links it keeps, certain coincident links fail to be removed, which leads to a low precision rate. For FCM ($k = 3$), since the links are clustered into three categories but only strong links are considered, it has the highest precision rate, but in same time, it retains the least amount of links and certain positive links maybe wrongly removed, leading to a low recall rate. Nevertheless, it still performs better than the original method according to the F1-score overall.

Regarding the running time performance, the table clearly shows that the modified method takes much less time than the original method. This also confirms that the time complexity of the modified approach is lower than the original method as discussed previously (in Sect. 3.3). Particularly, FCM ($k = 3$) takes least number of iterations to generate the results. Putting the F1-score and running time together, this set of experimental results indicate that FCM ($k = 2$) is an overall winner.

Figure 4 displays the network inferred through the use of the method that builds on fuzzy c-means, and Fig. 5 displays the network inferred through the

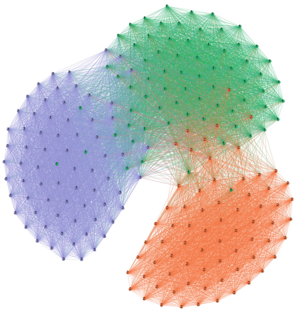


Fig. 4. Network of seeds dataset based on fuzzy c-means

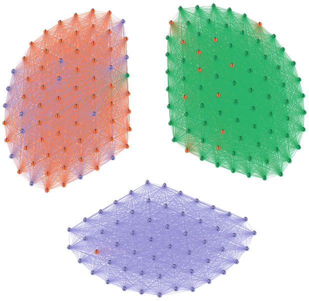


Fig. 5. Network of seeds dataset based on GMM and null model

original method. Objects with the same color have the same label. Comparing these two figures, it is clear that there are less objects being clustered into wrong groups in Fig. 4 than in Fig. 5. Further, even certain wrongly clustered can still be linked with the underlying true objects.

4.3 Results on Spatiotemporal Dataset (of No Ground Truth)

The dataset used in this experiment is the one adopted in the original work on animal social networks [2]. It came from a large amount of research into the foraging process in a population of Parus major at Wytham Woods, near Oxford, UK from 2007 to 2009 [1]. The dataset includes 1032797 records of 1241 different birds foraging in 69 different locations. Each record consists of three attributions: Bird ID, Time Stamp and Location ID.

It is hard to display the full network because of the huge scale. Therefore, as with the original work, 100 individuals are randomly selected to develop a network with both the original and the modified methods for comparison.

Figure 6 includes 1556 links, showing the result of running the modified method, and Fig. 7 includes 1599 links doing that of the original method. These two networks have resulted in visually similar structures and they have 1453 links in common (more than 90% of both methods).

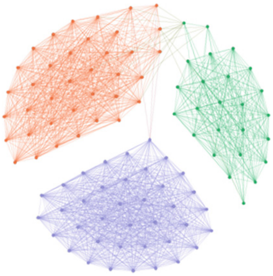


Fig. 6. Network of 100 individuals based on fuzzy c-means

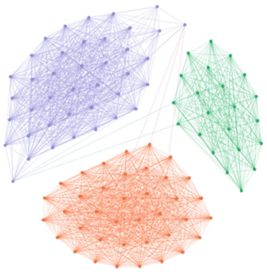


Fig. 7. Network of 100 individuals based on fuzzy c-means

For the entire dataset, the original method discovers 6857 strong links while the modified method generates 6637 strong links. Between the two networks, there are 6173 links in common. This confirms that the modified method can indeed work to build animal social networks (although for this particular dataset there is no ground truth to perform a more detailed comparison of both methods).

5 Conclusion

This paper has introduced a method to infer animal social networks from spatiotemporal data streams and a modified method which enhances coincident links filtering. The work has been motivated by the observation that the original method clusters individuals into different gathering events and those belonging to the same gathering event are treated as linked, thereby producing many spurious links. To effectively filter the coincident links, fuzzy c-means has been applied to modify this method by clustering the links into strong and weak links to enable the removal of weak ones efficiently. Experimented studies have demonstrated the potential of this work. The presented procedure has its generality: it can be applied not only to animal social networks but also to human social networks, and the features addressed can be other attributes rather than just time and location. For instance, people who have the same habits in sport, food, book, movie could be linked as friends.

The current work only deals with binary links within a social network. However, triple links may be inferred from such binary associations, e.g., with the support of link analysis [14,15]. This would be very useful in real world settings where missing information regarding a third party needs to be inferred to enrich the social networks from neighborhood binary relationships.

References

1. Psorakis, I., Roberts, S.J., Rezek, I., et al.: Inferring social network structure in ecological systems from spatio-temporal data streams. J. Roy. Soc. Interface (2012). rsif20120223

2. Krause, J., Lusseau, D., James, R.: Animal social networks: an introduction. Behav. Ecol. Sociobiol. **63**(7), 967–973 (2009)
3. Aebischer, N.J., Robertson, P.A., Kenward, R.E.: Compositional analysis of habitat use from animal radio-tracking data. Ecology **74**(5), 1313–1325 (1993)
4. White, G.C., Garrott, R.A.: Analysis of Wildlife Radio-Tracking Data. Elsevier, Amsterdam (2012)
5. Whitehead, H., Dufault, S.: Techniques for analyzing vertebrate social structure using identified individuals: review. Adv. Stud. Behav. **28**, 33–74 (1999)
6. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: the fuzzy c-means clustering algorithm. Comput. Geosci. **10**(2–3), 191–203 (1984)
7. Reynolds, D.A.: Gaussian mixture models. Encycl. Biom. **2009**, 659–663 (2009)
8. Moore, D.S., McCabe, G.P.: Introduction to the Practice of Statistics. WH Freeman/Times Books/Henry Holt and Co., New York (1989)
9. Van der Vaart, A.W.: Asymptotic Statistics. Cambridge University Press, Cambridge (2000)
10. Dunn, J.C.: A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters (1973)
11. Bezdek, J.C., Coray, C., Gunderson, R., et al.: Detection and characterization of cluster substructure I. Linear structure: fuzzy c-lines. SIAM J. Appl. Math. **40**(2), 339–357 (1981)
12. Brown, M.S., Pelosi, M.J., Dirska, H.: Dynamic-radius species-conserving genetic algorithm for the financial forecasting of Dow Jones index stocks. In: International Workshop on Machine Learning and Data Mining in Pattern Recognition, pp. 27–41. Springer, Heidelberg (2013)
13. MacKay, D.J.C.: Information Theory, Inference and Learning Algorithms. Cambridge University Press, Cambridge (2003)
14. Shen, Q., Boongoen, T.: Fuzzy orders-of-magnitude-based link analysis for qualitative alias detection. IEEE Trans. Knowl. Data Eng. **24**(4), 649–664 (2012)
15. Su, P., Shang, C., Shen, Q.: Link-based approach for bibliometric journal ranking. Soft. Comput. **17**(12), 2399–2410 (2013)

Optimisation

A Heuristic Approach for the Dynamic Frequency Assignment Problem

Khaled Alrajhi[1(✉)], Jonathan Thompson[2], and Wasin Padungwech[2]

[1] King Khalid Military Academy, Riyadh, Saudi Arabia
khalrajhi@kkma.edu.sa
[2] School of Mathematics, Cardiff University, Cardiff CF24 4AG, Wales, UK
{thompsonjml,padungwechw}@cardiff.ac.uk

Abstract. This study considers the dynamic frequency assignment problem, where new requests gradually become known and frequencies need to be assigned to those requests effectively and promptly with the minimum number of reassignments. The problem can be viewed as a combination of three underlying problems: the initial problem, the online problem, and the repair problem. In this study, a heuristic approach is proposed to solve this problem using different solution methods for each underlying problem. Moreover, the efficiency of this approach is improved by means of the *Gap* technique, which aims to identify a good frequency to be assigned to a given request. For the purpose of this study, new dynamic datasets are generated from static benchmark datasets. It was found that the performance of our approach is better than the state-of-the-art approach in the literature across the same set of instances.

1 Introduction

There has been an increasing interest in various dynamic optimization problems, in which some attributes of the problems change over time. Decisions have to be made at different points of time, and the quality of the solution depends on each of those decisions. Major difficulties of dynamic problems come from ignorance of how the problem will change in the future. Many real-life problems can be considered to be dynamic, but a majority of research has focused on static problems, where all the data is known in advance. Research into dynamic problems is growing in some areas such as graph colouring problems, scheduling problems and vehicle routing problems [1].

One of the dynamic problems is the dynamic frequency assignment problem (FAP), which was proposed in [2]. The FAP can be applied to wireless communication networks, which has many applications such as mobile phones, TV broadcasting, and Wi-Fi. In the dynamic FAP, new requests become known over time periods and frequencies need to be assigned to them effectively and promptly while satisfying a set of constraints (see Sect. 2). Hence, solving the dynamic FAP needs to deal with uncertain data as new data arrive in a dynamic process. There are two possible types of uncertain data: entirely accessible data and partially accessible data. The first type belongs to the area of robust optimization [3], where we need to find solution methods able to accommodate different realizations of data. The second type, which is considered in this study, corresponds to dynamic optimization [4], which has three features:

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_8

(1) new decisions are made one by one; (2) the decisions are non-adjustable unless necessary; (3) no information about the future is accessible. Each time period involves the static FAP, which can be represented as the well-known graph colouring problem (GCP) by representing each request as a vertex, each frequency as a colour and each constraint as an edge joining the corresponding vertices. In other words, the GCP can be seen as an underlying model to the FAP [5]. The FAP has been shown to be NP-complete [6], so it is common to use heuristics to find solutions to this problem.

During solving the dynamic FAP, if no feasible solution can be found, it is essential to change the previous decisions to improve the solution with the minimum number of changes. Although changing frequencies that have been assigned previously is technically allowed, in practice this can be time consuming and takes up human resources. Hence, the dynamic FAP states that changing frequencies of requests that are previously assigned should be avoided unless no other means of finding a feasible solution exists. Therefore, the objective of the dynamic FAP is to find a feasible solution with the minimum number of re-assigned requests. In this paper, the process of solving the dynamic FAP can be divided into several phases, and various approaches, including tabu search, are designed to solve each phase. The *Gap* technique [8] is also used to identify a good frequency to be assigned to a given request.

This paper is organised as follows: the next section presents an overview of the dynamic FAP. Section 3 provides a description of generating the new dynamic FAP datasets. Section 4 presents an overall structure of the proposed heuristic approach for the dynamic FAP. This includes a description of the initial assignment phase, the online assignment phase and the repair phase. In Sect. 5, results of various approaches are given, discussed and compared, and the best approach is compared with an existing approach in the literature. Finally, the conclusion is given in Sect. 6.

2 Overview of the Dynamic FAP

The dynamic FAP can be defined formally as follows: Given a set of requests $R = \{r_1, r_2, \ldots, r_{NR}\}$, where NR is the number of requests, a set of frequencies $F = \{f_1, f_2, \ldots, f_{NF}\} \subset \mathbb{Z}^+$, where NF is the number of frequencies. Let the frequency assigned to request r_i be denoted as f_{r_i}. The dynamic FAP has five types of constraints:

1. *Bidirectional constraints:* this type of constraint forms a link between each pair of (consecutively indexed) requests $\{r_{2i-1}, r_{2i}\}$, where $i = 1, \ldots, NR/2$. In these constraints, the frequencies $f_{r_{2i-1}}$ and $f_{r_{2i}}$ should be distance d_c apart, where d_c is a given constant. These constraints can be written as follows:

$$|f_{r_{2i-1}} - f_{r_{2i}}| = d_c \qquad \text{for } i = 1, \ldots, NR/2 \tag{1}$$

2. *Interference constraints:* this type of constraint forms a link between a pair of requests $\{r_i, r_j\}$. The frequencies f_{r_i} and f_{r_j} should be more than distance $d_{r_i r_j}$ apart, where $d_{r_i r_j}$ is a given constant. These constraints can be written as follows:

$$\left|f_{r_i} - f_{r_j}\right| > d_{r_i r_j} \qquad \text{for } 1 \le i < j \le NR \tag{2}$$

3. *Domain constraints:* the set of available frequencies for each request r_i is denoted by the domain $D_{r_i} \subset F$, where $\cup_{r_i \in R} D_{r_i} = F$. Hence, f_{r_i} must belong to D_{r_i}.
4. *Pre-assignment constraints:* for certain requests r_i, their frequencies are fixed and pre-assigned to given values p_{r_i}.
5. *Time period constraints:* this type of constraint gives the time period in which a request is known for the first time.

In the dynamic FAP, new requests appear in each time period and frequencies need to be assigned to them effectively and promptly. The objective is to find a feasible solution at each time period with the minimum number of re-assigned requests. An example of a dynamic FAP is illustrated in Fig. 1, where each node represents a request, each edge represents a bidirectional or an interference constraint, and each colour represents a time period in which a request becomes known for the first time.

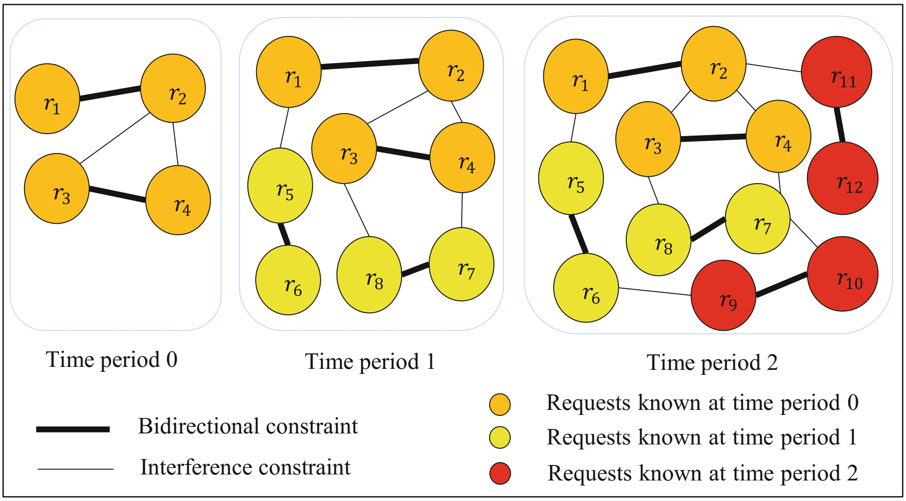


Fig. 1. A dynamic FAP instance over 3 time periods.

3 Generating the Dynamic FAP Datasets

To assess the proposed approaches in this study, new dynamic FAP datasets[1] are generated from the static FAP datasets (CELAR and GRAPH) by breaking down each static FAP instance into smaller sub-problems. To achieve this, each request is given an integer number between 0 and n, which indicates the time period in which it becomes known. In effect, the problem is divided into $n+1$ sub-problems $P_0, P_1, \ldots, P_n$, where n is the number of sub-problems after the initial sub-problem P_0, and each sub-problem P_i

[1] These datasets are available at https://dynamicfap.wordpress.com/.

contains a subset of requests which become known at time period i. It was found in preliminary study that the number of sub-problems does not impact on the performance of the approaches for solving the dynamic FAP. Here, the number of sub-problems is chosen to be 21 (i.e. $n = 20$).

Based on the number of the requests known at the initial sub-problem P_0, 10 different dynamic FAP instances are generated from each static FAP instance. These dynamic FAP instances are named using percentages which indicate the number of requests known at time period 0, namely 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%. Note that 100% means all the requests are known at time period 0 and so corresponds to the static FAP instance.

4 Heuristic Approach for the Dynamic FAP

Our heuristic approach consists of three solution phases:

1. *Initial assignment phase* aims to solve the initial sub-problem P_0, i.e. it assigns (feasibly, if possible) the requests known at time period 0.
2. *Online assignment phase* aims to solve the remaining sub-problems $P_1, \ldots, P_{20}$ in turn. In each sub-problem, the requests are assigned as they arrive dynamically based on their time period. At this stage, no existing assignments may be altered. If some request cannot be feasibly assigned, then we proceed to the repair phase.
3. *Repair phase* aims to solve the repair problem, i.e. attempting to feasibly assign frequencies to unassigned requests while trying to make as few changes as possible to the already assigned requests.

The process of our heuristic approach to solve the dynamic FAP is illustrated in Fig. 2. N_p denotes a given number of time periods in the problem.

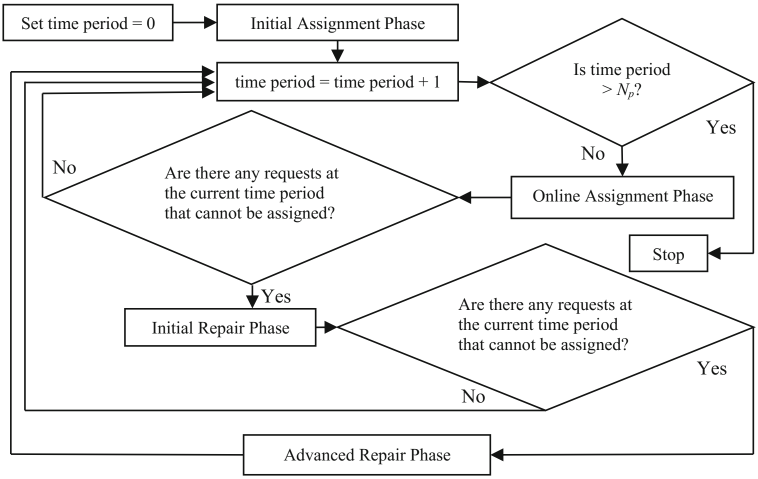

Set time period = 0
Initial Assignment Phase
time period = time period + 1
Is time period > N_p?
Yes
No
Online Assignment Phase
Stop
No
Are there any requests at the current time period that cannot be assigned?
Yes
Initial Repair Phase
Are there any requests at the current time period that cannot be assigned?
Yes
No
Advanced Repair Phase

Fig. 2. Overall structure of the heuristic approach for the dynamic FAP.

4.1 Initial Assignment Phase

An initial solution is generated by the following greedy algorithm: a request which has the smallest number of feasible frequencies is selected. Then, among those frequencies, the one which is feasible for most requests is assigned to the selected request. If there are no feasible frequencies for any request, a frequency is randomly selected. In case the initial solution is infeasible, a local search is used to reduce the number of violations.

4.2 Online Assignment Phase

The online assignment phase aims to solve the remaining sub-problems $P_1, \ldots, P_{20}$, consecutively. In each sub-problem, new requests arrive and need to be assigned feasibly to some frequencies. To do this, several decisions need to be made:

- In what order should the new requests be considered?
- For each request, which used feasible frequencies should be selected?
- If no used feasible frequencies are available, which unused feasible frequencies should be selected?

The following stages give answers to the above questions consecutively.

Request Selection Stage. 8 different techniques are considered:

- *Technique 1:* the request with the least number of feasible frequencies is selected. In case of a tie, the request that is involved in most (currently known) constraints is selected. If still there is a tie, one of them is randomly selected.
- *Technique 2:* the request that has the least number of feasible frequencies is selected. In case of a tie, one of them is randomly selected.
- *Technique 3:* the request that is involved in most (currently known) constraints is selected. In case of a tie, one of them is randomly selected.
- *Technique 4:* the request is randomly selected.

Techniques 5, 6, 7 and 8 are the same as techniques 1, 2, 3 and 4 respectively except that the requests are considered as pairs (instead of individuals) based on the bidirectional constraints (see Eq. 1).

Used Feasible Frequency Selection Stage. 3 techniques are considered:

- *The Ran technique:* one of the used feasible frequencies is randomly selected.
- *The Min technique:* the lowest value of the set of used feasible frequencies is selected. In case of a tie, one of them is randomly selected.
- *The Most technique:* among used feasible frequencies, one that is assigned to the most requests is selected. In case of a tie, one of them is randomly selected.

The *Min* and *Most* techniques aim to maximize the number of frequencies that are not selected from the set of used frequencies. This allows more choices of used frequencies for requests that will appear at later time periods.

Unused Feasible Frequency Selection Stage. 3 techniques are considered:

- *The Feas technique:* the frequency that is feasible for the most requests is selected. In case of a tie, one of them is randomly selected.
- *The Low technique:* the lowest value of the set of unused feasible frequencies is selected. In case of a tie, one of them is randomly selected.
- *The Gap technique* [8]: an unused frequency with the largest minimum gap between it and an already used frequency is selected. In case of a tie, one of them is randomly selected. The largest minimum gap leads to a greater probability that the interference constraints are satisfied.

Overall, these techniques aim to maximize the number of unused frequencies, which allows more choices of unused frequencies for requests that will appear later.

4.3 Repair Phase

The repair phase attempts to feasibly assign the unassigned requests from the previous phase while trying to minimize the number of re-assigned requests. This phase consists of two stages, which are the initial repair phase and, if needed, the advanced repair phase. The initial repair phase involves a simple method to find a feasible solution by making minor changes to the solution. If a feasible solution is found (i.e. all requires are now feasibly assigned), then the approach proceeds to the next sub-problem. Otherwise, the advanced repair phase is used, where the priority is to produce a feasible solution even if a large number of requests have to be re-assigned.

Initial Repair Phase. Given that the request r_i could not be feasibly assigned by the online assignment phase, then all the available frequencies for r_i (i.e. frequencies in the domain D_{r_i}) are ordered according to the number of violations that would result when r_i is assigned to each of them. After that, these frequencies are considered in turn starting with the frequency which would result in the smallest number of violations. Assume that such frequency is f_k. Then, each of the requests in those violations is re-assigned. If all of them can be feasibly re-assigned, then f_k is assigned to r_i. Otherwise, all the re-assignments are reversed and the next frequency is considered.

Advanced Repair Phase. This phase is only executed if some requests still could not be feasibly assigned after using the initial repair phase. In the advanced repair phase, all such requests are assigned frequencies that result in the minimum number of violations. Then, it tries to minimize the number of violations in the hope of finding a feasible solution using the tabu search with multiple neighbourhood structures [7].

5 Experiments and Results

This section compares the performance of various approaches for the dynamic FAP in the online assignment phase and the repair phase. After that, the best approach is compared with Dupont's approach in [2] using the generated datasets in this study. All the approaches were coded using FORTRAN 95 and all experiments were conducted on a 3 GHz Intel Core I3-2120 Processor with 8 GB RAM and 1 TB Hard Drive.

5.1 Online Assignment Phase

In this phase, different techniques in each selection stage (see Sect. 4.2) are compared. There are 8 potential techniques in the request selection stage, 3 potential techniques in the used feasible frequency selection stage, and 3 potential techniques in the unused feasible frequency selection stage. In total, this gives 72 different approaches. Testing all of them on 100 dynamic FAP instances would take excessive time. Hence, the best technique at each stage will be determined in turn by the following 3 experiments. To select the best technique in each experiment, firstly the techniques are ranked on each instance (the best one is given the lowest rank, ranks are averaged in case of a tie). Then, the one with the lowest rank sum over all the dynamic FAP instances is considered the best.

Experiment 1: This experiment compares 8 different techniques of the request selection stage, while selecting the *Ran* and *Feas* techniques for the used and unused feasible frequency selection stages, respectively. Fig. 3 shows that the minimum total rank is 247, which corresponds to technique 5 (consider requests as pairs based on the bidirectional constraints - see Eq. 1). In terms of the run time, based on the Wilcoxon signed-rank test, there is no significant difference between them.

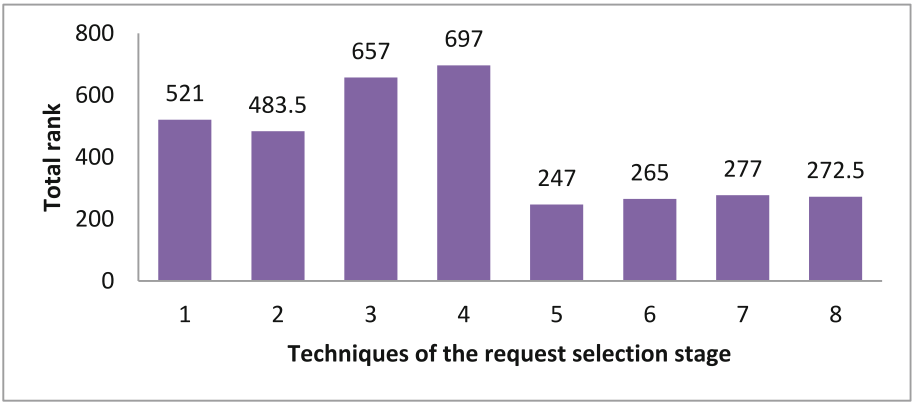


Fig. 3. The total rank for each approach based on Experiment 1.

Experiment 2: This experiment compares 3 different techniques of the used feasible frequency selection stage, while using technique 5 for the request selection stage (see Experiment 1) and the *Feas* technique for the unused feasible frequency selection stage. Fig. 4 shows little difference in the results achieved by the 3 different techniques in the used feasible frequency selection stage. Nevertheless, the best approach is the *Most* technique (with the total rank = 182.5). In terms of the run time, based on the Wilcoxon signed-rank test, there is no significant difference between them.

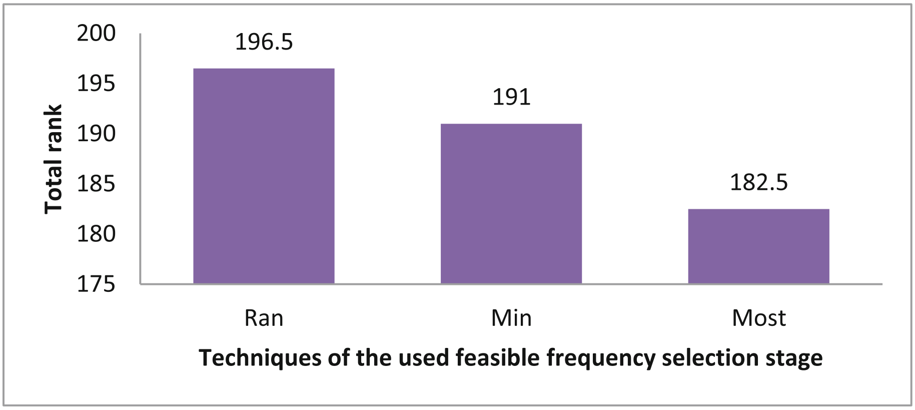


Fig. 4. The total rank for each approach based on Experiment 2.

Experiment 3: This experiment compares 3 different techniques of the unused feasible frequency selection stage, while using technique 5 for the request selection stage (see Experiment 1) and the Most technique for the used feasible frequency selection stage (see Experiment 2). Figure 5 shows that the minimum number of the total rank was 180, which resulted from the *Gap* technique. In terms of the run time, based on the Wilcoxon signed-rank test, there is no significant difference between them.

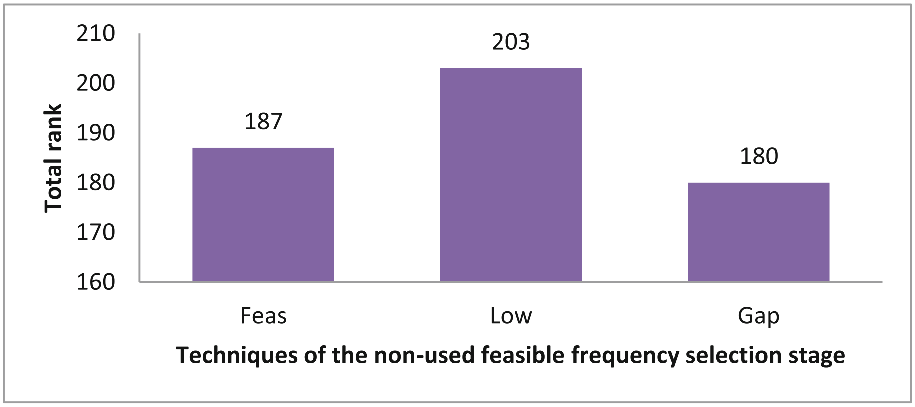


Fig. 5. The total rank for each approach based on Experiment 3.

Based on the above three experiments, the best approach for the online assignment phase uses technique 5 in the request selection stage, the Most technique in the used feasible frequency selection stage and the *Gap* technique in the unused feasible frequency selection stage.

5.2 Repair Phase

This section compares two approaches: the first uses the initial repair phase only and the second includes the advanced repair phase. The results shown in this section include the number of used frequencies in a feasible solution, the run time and the number of re-assigned requests in the repair phase (denoted by Repair). Note that a dash "-" means that no feasible solution is found.

Initial Repair Phase. The performance of the approach for the dynamic FAP using only the initial repair phase is given in Table 1. It can be seen that this approach achieved feasible solutions for all dynamic FAP instances for CELAR 02, CELAR 04 and GRAPH 14. In contrast, it could not achieve a feasible solution for all dynamic

Table 1. Results of the proposed approach for the dynamic FAP using the initial repair phase.

Instance	Number of requests known at time period 0									
	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
CELAR 01	26	28	–	30	–	–	–	26	–	30
Time	8.5 s	19.5 s	–	2 min	–	–	–	7 min	–	11 min
Repair	54	854	–	252	–	–	–	250	–	102
CELAR 02	16	18	18	20	16	16	18	18	16	14
Time	0.6 s	2.9 s	8.5 s	13 s	30 s	33 s	1 min	49 s	54 s	59 s
Repair	200	182	160	146	114	114	10	50	48	10
CELAR 03	24	26	20	24	–	26	26	24	28	–
Time	1.7 s	7.3 s	16.1 s	54.1 s	–	1 min	1.2 min	3.2 min	2.5 min	–
Repair	38	46	334	268	–	228	24	106	4	–
CELAR 04	46	46	46	46	46	46	46	46	46	46
Time	24 s	24.3 s	56.3 s	1.1 min	1.5 min	1.5 min	1.6 min	2.1 min	2.3 min	2.3 min
Repair	18	24	20	18	21	18	24	12	10	16
CELAR 11	–	–	–	–	–	–	–	–	–	–
Time	–	–	–	–	–	–	–	–	–	–
Repair	–	–	–	–	–	–	–	–	–	–
GRAPH 01	–	24	28	–	–	–	22	22	22	20
Time	–	2.8 s	7.5 s	–	–	–	47 s	1.1 min	1 min	2.1 min
Repair	–	18	16	–	–	–	72	52	48	20
GRAPH 02	22	–	26	–	–	28	20	26	18	22
Time	1.7 s	–	38 s	–	–	1.5 min	2.8 min	3.1 min	4.5 min	4.8 min
Repair	400	–	26	–	–	6	112	50	86	44
GRAPH 08	–	–	–	–	–	–	–	–	–	–
Time	–	–	–	–	–	–	–	–	–	–
Repair	–	–	–	–	–	–	–	–	–	–
GRAPH 09	–	–	–	–	–	–	–	–	–	–
Time	–	–	–	–	–	–	–	–	–	–
Repair	–	–	–	–	–	–	–	–	–	–
GRAPH 14	16	12	14	16	14	16	20	16	14	14
Time	7.5 s	42.5 s	58.5 s	2.1 min	3.6 min	4.2 min	6.9 min	19 min	13 min	18 min
Repair	916	808	760	634	564	464	152	280	166	76

FAP instances for GRAPH 08 and GRAPH 09. The number of re-assigned requests (labelled as Repair) fluctuates, and is noticeably high for most instances. Table 1 also shows that the run time increased with the number of requests known at time period 0.

Advanced Repair Phase. Table 2 shows that using the advanced repair phase in addition to the initial repair phase resulted in feasible solutions for all instances. The number of re-assigned requests (labelled as Repair) still fluctuates and does not have a clear relationship with the number of requests known at time period 0. However, the run time increased with the number of requests known at the time period 0.

Table 2. Results of the proposed approach for the dynamic FAP using the initial and advanced repair phases.

Instance	Number of requests known at time period 0									
	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
CELAR 01	26	26	34	26	32	30	30	26	32	26
Time	5.8 s	9.2 s	15.5 s	38.5 s	48.5 s	57.5 s	1 min	2.13 min	2.8 min	3.7 min
Repair	54	172	40	54	16	22	36	250	42	4
CELAR 02	16	16	18	16	18	16	18	16	16	14
Time	0.4 s	0.5 s	0.7 s	1.1 s	3.3 s	4.8 s	7 s	3.4 s	12 s	11 s
Repair	200	182	160	146	114	114	10	50	48	10
CELAR 03	24	24	22	28	22	26	22	20	24	18
Time	2.1 s	3.3 s	5.9 s	6.2 s	13.1 s	5.5 s	19.1 s	24.1 s	28.1 s	29.1 s
Repair	38	46	90	268	208	66	60	106	70	50
CELAR 04	46	46	46	46	46	46	46	46	46	46
Time	24 s	24.3 s	44 s	1.1 min	1.1 min	1.3 min	1.5 min	2.1 min	2.3 min	2.3 min
Repair	18	24	8	20	10	18	20	12	10	16
CELAR 11	42	42	40	42	44	32	42	42	42	40
Time	26.3 s	23.3 s	2.6 min	1.9 min	3.9 min	1.6 min	3.7 min	4 min	4.7 min	9.3 min
Repair	28	52	418	336	432	24	424	376	376	350
GRAPH 01	20	30	24	26	24	22	22	24	22	22
Time	0.5 s	0.5 s	2.1 s	1.1 s	11 s	3.8 s	11 s	4.2 s	15 s	6.7 s
Repair	52	6	10	14	10	34	72	52	48	20
GRAPH 02	22	22	28	20	24	24	24	28	22	20
Time	1.2 s	1.4 s	3 s	2.9 s	5.1 s	8.1 s	30.1 s	30.1 s	43.1 s	1.1 min
Repair	400	4	28	22	256	210	46	60	86	42
GRAPH 08	30	36	44	38	36	32	40	36	34	32
Time	10.3 s	17.3 s	15.3 s	14.3 s	19.3 s	26.3 s	59.3 s	1.1 min	1.5 min	2.1 min
Repair	24	24	44	24	38	78	28	16	34	30
GRAPH 09	40	44	42	38	42	42	44	42	40	30
Time	18.5 s	20.5 s	29.5 s	42.5 s	1.1 min	1.2 min	1.9 min	2.2 min	3.9 min	4.1 min
Repair	22	24	46	38	46	58	30	10	2	6
GRAPH 14	16	16	16	18	18	14	12	12	14	14
Time	4.5 s	5.5 s	7.5 s	16.5 s	31.5 s	1.1 min	2.2 min	3.1 min	7.3 min	15 min
Repair	916	916	760	634	564	6	344	286	166	76

5.3 Results Comparison with Other Approaches

Here, the existing algorithm [2] (which will be referred to as "Dupont's approach") is compared with our approach for the dynamic FAP. Private correspondence with the authors in [2] revealed that the datasets and the algorithm in their paper were not available, so we re-implemented Dupont's approach on the new dynamic FAP datasets (see Sect. 3). Table 3 shows that Dupont's approach achieved feasible solutions for all the instances. It can also been seen that the number of re-assigned requests (labelled as Repair) fluctuates and does not have a clear relationship with the number of requests known at time period 0, as seen previously for our approach.

Table 3. Results of Dupont's approach for the dynamic FAP.

Instance	Number of requests known at time period 0									
	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
CELAR 01	36	30	42	34	30	32	30	34	30	34
Time	7.2 min	8.9 min	9.2 min	11 min	11.5 min	21 min	24 min	26 min	30 min	31 min
Repair	230	912	620	340	322	320	240	430	234	420
CELAR 02	18	18	18	22	22	18	18	18	18	16
Time	2 s	4.8 s	6.2 s	10 s	18 s	38 s	55 s	57 s	1.3 min	1.8 min
Repair	200	190	190	160	70	30	90	40	80	30
CELAR 03	32	30	34	24	30	32	36	30	30	30
Time	20 s	45 s	50 s	54 s	52 s	1.8 min	2.1 min	4.1 min	4.7 min	5.2 min
Repair	70	200	430	330	204	220	110	98	60	40
CELAR 04	46	46	46	46	46	46	46	46	46	46
Time	1.7 min	2.3 min	4.1 min	6.1 min	6.2 min	8.1 min	10.2 min	11.9 min	20.1 min	30 min
Repair	42	80	32	22	52	24	92	88	22	72
CELAR 11	42	44	44	42	44	44	44	42	40	40
Time	8.1 min	8 min	9.5 min	9.7 min	10 min	11 min	13 min	14.7 min	17 min	18 min
Repair	356	556	312	400	334	366	342	322	398	338
GRAPH 01	40	40	42	32	36	40	42	44	38	38
Time	38 s	54 s	2.3 min	3.1 min	3.4 min	3.7 min	4.7 min	5.5 min	7.1 min	7.2 min
Repair	110	42	146	204	172	190	176	144	142	120
GRAPH 02	36	40	40	40	40	40	42	38	40	42
Time	1.7 min	2.8 min	2.9 min	3 min	3.1 min	4.2 min	4.8 min	5.2 min	5.3 min	6.8 min
Repair	212	76	252	302	294	306	258	256	256	242
GRAPH 08	48	46	48	44	44	44	48	48	42	44
Time	1.6 min	1.5 min	2 min	2.1 min	2.8 min	3.5 min	3.4 min	4.5 min	4.8 min	5.1 min
Repair	252	194	264	306	276	300	286	288	318	302
GRAPH 09	46	46	42	46	46	44	44	46	44	42
Time	4 min	4.1 min	5.3 min	6.2 min	6.7 min	8.5 min	8.8 min	10.2 min	12 min	17 min
Repair	308	178	300	298	280	278	274	290	308	302
GRAPH 14	26	26	36	34	34	32	34	34	34	28
Time	9.1 min	10 min	11 min	12 min	13.4 min	14 min	15.5 min	19.2 min	24.1 min	26 min
Repair	916	916	840	744	564	912	340	230	850	420

Figure 6 shows that our approach achieved a lower number of re-assigned requests on average on all the instances. It is found by the Wilcoxon signed-rank test at the 0.05 significance level that the difference is significant. Moreover, the average run times of our approach and Dupont's approach are shown in Fig. 7. It is also found by the Wilcoxon signed-rank test at the 0.05 significance level that the average run time of our approach is significantly better. This shows that our approach could generate better results than the state-of-the-art on the same instances.

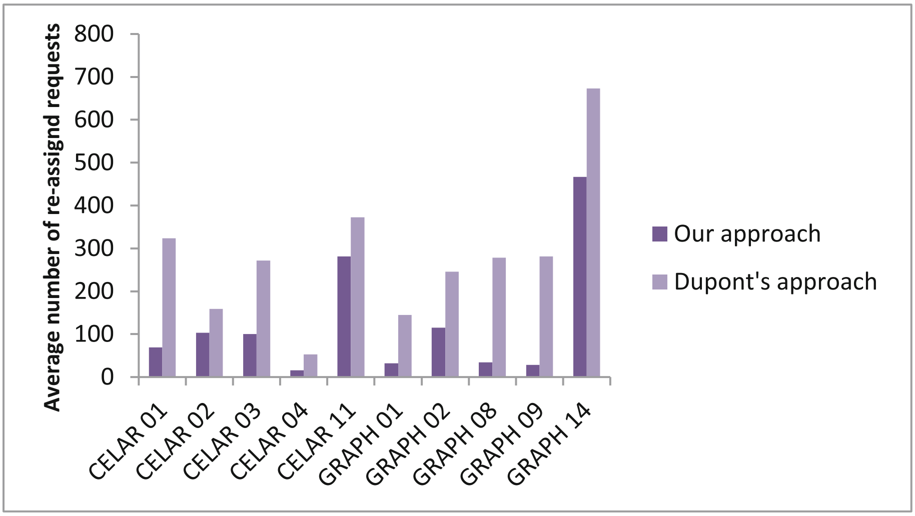


Fig. 6. Average number of re-assigned requests of our approach and Dupont's approach.

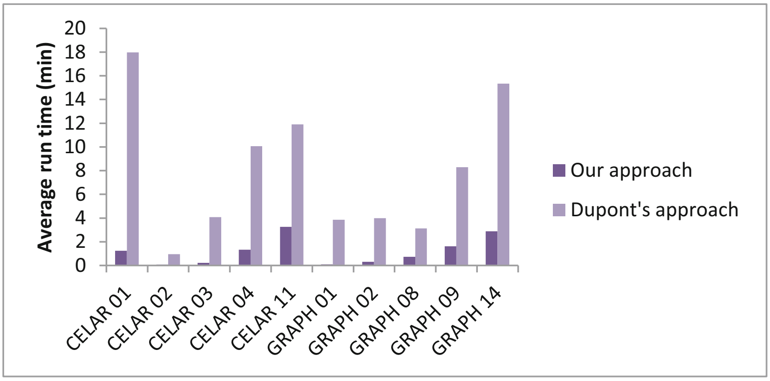


Fig. 7. Average run time of our approach and Dupont's approach.

6 Conclusions

This paper studied the dynamic FAP, where new requests become known over time. The objective of the dynamic FAP is to find a feasible solution with the minimum number of re-assigned requests. The proposed heuristic approach aims to solve the dynamic FAP through solving its three underlying problems, which are the static problem, the online problem and the repair problem. Hence, the heuristic approach consists of three solution phases: the first phase is the initial assignment phase (where the requests known at the start are feasibly assigned). The second phase is the online assignment phase (where the requests which dynamically arrive are feasibly assigned, if possible). The efficiency of the approach in this phase is improved by means of the *Gap* technique, which aims to identify a good frequency to be assigned to a given request. The third phase is the repair phase (where the unassigned requests from the previous phase are feasibly assigned, if possible). The repair phase includes two stages, which are the initial repair phase and the advanced repair phase.

New dynamic datasets are generated from the static benchmark datasets (CELAR and GRAPH), and various techniques in each phase of the approach were tested on those datasets. It was found that the best variant of our approach could outperform the state-of-the-art approach in the literature across the same set of instances.

References

1. Dupont, A., Vasquez, M.: Solving the dynamic frequency assignment problem. In: Proceedings of the 6th International Conference of Meta-heuristics (2005)
2. Dupont, A., Linhares, A., Artigues, C., Feillet, D., Michelon, P., Vasquez, M.: The dynamic frequency assignment problem. Eur. J. Oper. Res. **195**(1), 75–88 (2008)
3. Kouvelis, P., Yu, G.: Robust Discrete Optimization and its Applications. Kluwer Academic Publishers, Boston (1997)
4. Halldorsson, M., Szegedy, M.: Lower bounds for on-line graph coloring: SODA1992. In: Proceedings of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 211–216 (1992)
5. Hale, W.: Frequency assignment: theory and applications. Proc. IEEE **68**, 1497–1514 (1980)
6. Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman W.H. and Company, San Francisco (1979)
7. Alrajhi, K., Thompson, J., Padungwech, W.: Tabu search hybridized with multiple neighbourhood structures for the static frequency assignment problem. In: Proceedings of the 10th International Workshop on Hybrid Meta-heuristics, HM 2016. Lecture Notes in Computer Science, vol. 9668, pp. 157–170. Springer (2016)
8. Alrajhi, K., Padungwech, W.: A dynamic tabu search approach for solving the static frequency assignment problem. In: Advances in Computational Intelligence Systems, pp. 59–70. Springer (2017)

Applying ACO to Large Scale TSP Instances

Darren M. Chitty(✉)

Department of Computer Science, University of Bristol,
Merchant Venturers Building, Woodland Road, Bristol BS8 1UB, UK
darrenchitty@googlemail.com

Abstract. Ant Colony Optimisation (ACO) is a well known metaheuristic that has proven successful at solving Travelling Salesman Problems (TSP). However, ACO suffers from two issues; the first is that the technique has significant memory requirements for storing pheromone levels on edges between cities and second, the iterative probabilistic nature of choosing which city to visit next at every step is computationally expensive. This restricts ACO from solving larger TSP instances. This paper will present a methodology for deploying ACO on larger TSP instances by removing the high memory requirements, exploiting parallel CPU hardware and introducing a significant efficiency saving measure. The approach results in greater accuracy and speed. This enables the proposed ACO approach to tackle TSP instances of up to 200K cities within reasonable timescales using a single CPU. Speedups of as much as 1200 fold are achieved by the technique.

Keywords: Ant Colony Optimisation · Travelling Salesman Problem · High performance computing

1 Introduction

Ant Colony Optimisation (ACO) [8] is a metaheuristic which has demonstrated significant success in solving Travelling Salesman Problems (TSP) [7]. The technique simulates ants moving through a fully connected network using pheromone levels to guide their choices of which cities to visit next to build a complete tour. However, ACO has two drawbacks the first being significant memory requirements to store the pheromone levels on every edge. Secondly, simulating ants by making probabilistic decisions at each city to determine the next city to visit makes ACO computationally intensive. Therefore, ACO will struggle when applied to larger TSP instances. Consider, the pheromone matrix which requires an n by n matrix whereby n is the number of cities. As the number of cities increases linearly, a quadratic increase in memory requirements is observed. The same is true for probabilistically simulating ants to construct a tour. This paper will address these issues enabling ACO to be applied to larger TSP instances.

The paper is laid out as follows; Sect. 2 will describe ACO, Sect. 3 will present a scalable version of ACO to apply to large scale TSP instances whilst Sect. 4 will demonstrate its effectiveness on well known TSP instances. Finally Sect. 5 demonstrates the approach on TSP instances of up to 200,000 cities.

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_9

2 ACO Applied to the TSP

The Travelling Salesman Problem (TSP) is a task where the objective is to visit every city in the problem once minimising the total distance travelled. The symmetric TSP can be represented as a complete weighted graph $G = (V, E, d)$ where $V = \{1, 2, \ldots, n\}$ is a set of vertices defining each city and $E = \{(i, j)|(i, j) \in V \times V\}$ the edges consisting of the distance d between pairs of cities such that $d_{ij} = d_{ji}$. The objective is to find a Hamiltonian cycle in G of minimal length.

Ant Colony Optimisation (ACO) applied to the TSP involves simulated ants moving through the graph G visiting each city once and depositing pheromone as they go. The level of pheromone deposited is defined by the quality of the tour the given ant finds. Ants probabilistically decide which city to visit next using this pheromone level on the edges of graph G and heuristic information based upon the distance between an ant's current city and unvisited cities. An *evaporation* effect is used to prevent pheromone levels reaching a state of local optima. Therefore, ACO consists of two stages, the first *tour construction* and the second stage *pheromone update*. The tour construction stage involves m ants constructing complete tours. Ants start at a random city and iteratively make probabilistic choices as to which city to visit next using the *random proportional rule* whereby the probability of ant k at city i visiting city $j \in N^k$ is defined as:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} \tag{1}$$

where $[\tau_{il}]$ is the pheromone level deposited on the edge leading from city i to city l; $[\eta_{il}]$ is the heuristic information consisting of the distance between city i and city l set at $1/d_{il}$; α and β are tuning parameters controlling the relative influence of the pheromone deposit $[\tau_{il}]$ and the heuristic information $[\eta_{il}]$.

Once all ants have completed the tour construction stage, pheromone levels on the edges of graph G are updated. First, evaporation of pheromone levels upon every edge of graph G occurs whereby the level is reduced by a value ρ relative to the pheromone upon that edge:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} \tag{2}$$

where p is the *evaporation rate* typically set between 0 and 1. Once this evaporation is completed each ant k will then deposit pheromone on the edges it has traversed based on the quality of the tour it found:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^k \tag{3}$$

where the amount of pheromone ant k deposits, $\Delta\tau_{ij}^k$ is defined by:

$$\Delta\tau_{ij}^k = \begin{cases} 1/C^k, & \text{if edge } (i, j) \text{ belongs to } T^k \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $1/C^k$ is the length of ant k's tour T^k. This methodology ensures that shorter tours found by an ant result in greater levels of pheromone being deposited on the edge of the given tour.

3 Addressing the Scalability of ACO

A key issue with ACO is its memory requirements in the form of the pheromone matrix which stores the level of pheromone on every edge between each city. Thus an n by n size matrix is required in memory to store this information so for a 100,000 city problem, a 100,000 by 100,000 matrix is required. Using a float datatype requiring four bytes of memory, this matrix will need approximately 37 GB of memory, much more than typically available on CPUs. However, a variant of ACO exists which dispenses with the need for a pheromone matrix, Population-based ACO (P-ACO) [10]. With this approach, a population of tours are maintained ($k_{long}(t)$) whereby the best tour at each iteration t is added. Since $k_{long}(t)$ is of a fixed size tours are added in a First In First Out (FIFO) manner. Pheromone levels are calculated by using the $k_{long}(t)$ information. An ant at a given city calculates the pheromone levels by examining the edges that were traversed in $k_{long}(t)$ from the given city. Thus there is no pheromone matrix and no pheromone evaporation. If $k_{long}(t)$ is significantly less than the number of cities then this is a considerable saving in memory requirements.

In this paper, some modifications to P-ACO are implemented. Firstly, instead of using a store of best found tours updated in a FIFO manner, each ant has a *local memory* (l_{best}) containing the best tour that the ant has found, a *steady-state* mechanism. These l_{best} tours are used to provide pheromone level information to ants when probabilistically deciding which city to next visit. This is similar in effect to Particle Swarm Optimisation (PSO) [9] whereby particles use both their local best solution and a global best to update their position. Secondly, the amount of pheromone an edge from an l_{best} tour contributes equates to the quality of the *global best* (g_{best}) tour divided by that of the l_{best} hence a value between 0.0 and 1.0. These measures are taken to increase diversity.

Moreover, to gain the maximum available performance of the P-ACO approach from a CPU, an asynchronous parallel approach is used with multiple threads of execution and each thread simulates a number of ants. Moreover, the choosing of the next city to visit is decided by multiplying the heuristic information, the pheromone level and a random probabilistic value between 0.0 and 1.0 and the city with the greatest combined value is selected as the next to visit. This approach is known as the *Independent Roulette* approach [3]. This allows the utilisation of the extended Single Instruction Multiple Data (SIMD) registers available in a CPU through AVX for the probabilistic decision making process. These extra wide registers enable up to eight edge comparisons to be made in parallel. Using a parallel methodology with AVX registers improves the computational speed by approximately 30–40× when using a quad core processor.

3.1 Introducing *PartialACO*

The most computationally expensive aspect of the ACO algorithm is the *tour construction* phase. This aspect of ACO has an exponential increase in computation time cost as the number of cities increases. Moreover, as an ant repeatedly probabilistically decides at each city which to visit next it could be considered that the greater number of cities in a tour that requires constructing, the greater the probability an ant will eventually make a poor choice of city to visit next resulting in a low quality tour. Hence, it is hypothesised that perhaps it would be advantageous for ants to only change part of a known good tour. To do so would firstly reduce the computational complexity and secondly reduce the probability of an ant making a poor decision at some point of the tour construction. For the P-ACO approach detailed previously, the part of the tour that is not changed by an ant could be based upon its l_{best} tour. Essentially, at each iteration an ant randomly chooses a city to start its tour from and then a random number of cities to preserve from its l_{best} tour. The remaining part of the tour will be constructed as normal. This methodology is similar to crossover in Genetic Algorithms (GA) [11] for the TSP whereby a large section of a tour is preserved and the remaining aspect constructed from another tour whilst avoiding repetition.

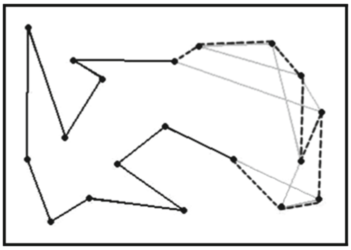


Fig. 1. Illustration of *PartialACO* whereby the l_{best} tour of an ant is partially modified by the ant. The dark part of the tour is retained and the lighter part discarded. *PartialACO* then completes the tour as normal creating a new tour (dashed line).

Algorithm 1. *PartialACO*

```
for each ant do
    Generate initial tour using P-ACO approach
end for
for number of iterations do
    for each ant do
        Select random starting city from current $l_{best}$ tour
        Select random number of cities in $l_{best}$ tour to preserve
        Copy $l_{best}$ tour from starting city for the specified random number of cities
        Complete remaining aspect of tour using P-ACO approach
        If new tour better than $l_{best}$ tour then update $l_{best}$ tour
    end for
end for
Output best $l_{best}$ tour (the $g_{best}$ tour)
```

Figure 1 visualises the concept whereby the dark part of the l_{best} tour is preserved and the rest discarded and then this partial tour is completed using ACO. Henceforth, this implementation of ACO will be referred to as *PartialACO*. A high level overview of the technique is shown in Algorithm 1.

4 Experiments with *PartialACO*

To test the effectiveness of the *PartialACO* approach experiments will be conducted using five standard TSP problems of increasing size from the TSPLIB library. Sixteen ants, two per parallel thread of execution, will be simulated for 100,000 iterations with the α and β parameters both set to a value of 5.0 to reduce the influence of heuristic information and increase the influence of pheromone from good tours. Results are averaged over 100 random runs and experiments are conducted using an Intel i7 processor using eight parallel threads of execution and the AVX registers. Table 1 shows the results from the standard P-ACO approach whereby full length tours are constructed by each ant at every iteration. The average accuracy ranges from between 4 and 13% of the known optimum. Table 2 demonstrates the results from the *PartialACO* approach described in this paper whereby only a portion of each ant's best found tour is exposed to modification. From these results it can be observed that accuracy has been improved

Table 1. Results from using standard P-ACO approach with accuracy expressed as percentage difference from known optimum. Results averaged over 100 random runs.

TSP instance	Accuracy (% error)			Execution time (in seconds)
	Average	Best	Worst	
pcb442	4.16 ± 1.37	1.58	8.78	40.53 ± 0.42
d657	8.02 ± 2.41	3.43	12.44	72.87 ± 0.53
rat783	4.08 ± 1.13	2.27	7.82	96.23 ± 0.81
pr1002	7.88 ± 1.49	5.18	12.47	145.59 ± 0.91
pr2392	13.47 ± 1.13	10.91	15.98	688.13 ± 2.86

Table 2. Results from using *PartialACO* approach with accuracy expressed as percentage difference from known optimum and relative speedup to using standard P-ACO approach. Results averaged over 100 random runs.

TSP instance	Accuracy (% error)			Execution time (in seconds)	Relative speedup
	Average	Best	Worst		
pcb442	2.72 ± 0.67	1.14	4.75	17.94 ± 0.26	2.26×
d657	4.33 ± 0.73	2.88	7.17	31.68 ± 0.31	2.30×
rat783	3.64 ± 0.60	2.18	5.99	40.01 ± 0.40	2.41×
pr1002	4.06 ± 0.57	2.61	5.24	58.27 ± 0.33	2.50×
pr2392	9.47 ± 2.09	5.01	12.84	245.96 ± 0.88	2.80×

Table 3. Results from using *PartialACO* approach with restrictions on the maximum permissible modification. Accuracy expressed as percentage difference from known optimum with relative speedup to using standard P-ACO approach reported. Results are averaged over 100 random runs.

TSP instance	Max. Modification	Accuracy (% error)			Execution time (in seconds)	Relative speedup
		Average	Best	Worst		
pcb442	50%	4.90 ± 1.31	2.31	9.48	9.01 ± 0.22	4.50×
	40%	6.79 ± 1.57	3.64	10.11	7.44 ± 0.18	5.45×
	30%	8.95 ± 1.71	5.05	13.13	6.05 ± 0.12	6.70×
	20%	11.91 ± 2.09	6.36	15.52	4.91 ± 0.10	8.26×
	10%	16.59 ± 2.16	9.70	21.71	3.97 ± 0.05	10.22×
d657	50%	7.29 ± 1.00	5.14	9.66	14.48 ± 0.22	5.03×
	40%	8.40 ± 1.34	5.34	11.30	11.62 ± 0.15	6.27×
	30%	10.23 ± 1.46	6.84	13.87	9.27 ± 0.16	7.86×
	20%	13.12 ± 1.97	6.99	17.87	7.16 ± 0.11	10.17×
	10%	17.12 ± 1.69	12.21	22.37	5.53 ± 0.06	13.17×
rat783	50%	7.67 ± 1.38	3.52	11.03	17.66 ± 0.16	5.45×
	40%	9.62 ± 1.20	4.73	12.41	14.08 ± 0.15	6.83×
	30%	11.34 ± 1.43	7.46	15.95	10.91 ± 0.18	8.82×
	20%	13.46 ± 1.68	9.45	16.77	8.27 ± 0.13	11.64×
	10%	17.09 ± 1.53	11.78	20.97	5.98 ± 0.08	16.10×
pr1002	50%	7.10 ± 1.32	4.24	11.54	24.36 ± 0.35	5.98×
	40%	9.24 ± 1.47	5.73	13.03	19.34 ± 0.27	7.53×
	30%	10.52 ± 1.70	6.12	13.41	14.70 ± 0.19	9.91×
	20%	12.75 ± 1.84	7.96	16.98	10.93 ± 0.16	13.33×
	10%	16.32 ± 1.57	11.43	20.44	7.83 ± 0.12	18.61×
pr2392	50%	13.98 ± 1.45	11.41	16.61	82.76 ± 0.37	8.31×
	40%	16.37 ± 1.18	13.32	18.97	61.25 ± 0.30	11.23×
	30%	18.01 ± 1.54	14.15	21.14	43.45 ± 0.17	15.84×
	20%	20.01 ± 1.56	15.33	22.22	29.84 ± 0.17	23.06×
	10%	21.92 ± 1.02	19.87	24.56	20.74 ± 0.18	33.18×

for all TSP instances by several percent. More importantly, the computational speed of the approach has been increased significantly. A speedup of up to 2.8× is observed with speedups increasing with the size of the TSP instance. Thus, *PartialACO* is demonstratively both faster and more accurate.

Although the initial results from *PartialACO* have demonstrated a speed advantage with improved accuracy, it is possible to increase the speed of the approach further. Currently, a random part of the local best tour of an ant is preserved and the rest exposed to ACO to modify it. However, the part that is modified could be restricted to a maximum percentage of the l_{best} tour. For instance, a maximum percentage modification of 50% could be used thus for

Table 4. Results from using *PartialACO* approach with 0.95 probability and restrictions on the maximum permissible modification. Accuracy expressed as percentage difference from known optimum with relative speedup to using standard P-ACO approach reported. Results are averaged over 100 random runs.

TSP instance	Max. modification	Accuracy (% Error)			Execution time (in seconds)	Relative speedup
		Average	Best	Worst		
pcb442	50%	2.55 ± 0.94	1.19	5.25	10.47 ± 0.21	3.87×
	40%	2.61 ± 0.94	1.01	5.22	9.00 ± 0.22	4.50×
	30%	3.26 ± 1.41	1.26	7.15	7.63 ± 0.16	5.31×
	20%	3.35 ± 1.34	1.43	7.95	6.37 ± 0.13	6.36×
	10%	3.96 ± 1.67	1.55	8.98	5.28 ± 0.11	7.67×
d657	50%	4.79 ± 1.25	2.61	7.71	17.08 ± 0.21	4.27×
	40%	5.28 ± 1.26	2.83	8.99	14.26 ± 0.22	5.11×
	30%	5.65 ± 1.33	2.95	9.03	12.05 ± 0.20	6.05×
	20%	6.34 ± 1.39	3.19	9.30	9.95 ± 0.17	7.32×
	10%	7.27 ± 1.64	3.38	11.39	8.15 ± 0.13	8.94×
rat783	50%	4.46 ± 1.45	1.81	8.60	21.23 ± 0.29	4.53×
	40%	5.04 ± 1.30	2.37	7.53	17.92 ± 0.22	5.37×
	30%	5.89 ± 1.48	2.25	9.40	14.91 ± 0.20	6.45×
	20%	6.99 ± 1.30	3.10	10.26	12.27 ± 0.14	7.84×
	10%	8.14 ± 1.83	3.30	10.90	9.95 ± 0.16	9.67×
pr1002	50%	4.75 ± 1.33	2.28	8.43	30.05 ± 0.30	4.84×
	40%	5.28 ± 1.16	2.91	7.97	24.96 ± 0.28	5.83×
	30%	6.02 ± 1.23	3.27	9.41	20.66 ± 0.22	7.05×
	20%	6.58 ± 1.17	3.69	9.04	16.95 ± 0.21	8.59×
	10%	8.11 ± 1.46	2.76	10.55	13.88 ± 0.15	10.49×
pr2392	50%	10.58 ± 1.07	8.31	12.54	111.98 ± 0.49	6.15×
	40%	10.86 ± 1.11	7.69	13.10	90.58 ± 0.36	7.60×
	30%	11.11 ± 0.96	9.48	13.17	73.76 ± 0.38	9.33×
	20%	11.32 ± 1.12	8.50	13.65	60.03 ± 0.35	11.46×
	10%	11.80 ± 1.02	9.81	13.84	49.79 ± 0.29	13.82×

a 100 city problem at least part of the l_{best} tour consisting of 50 cities will be preserved. Reducing the degree to which the l_{best} tour of an ant can be changed could also improve tour quality by increased tour *exploitation* whilst also increasing the speed advantage of *PartialACO*. Table 3 demonstrates the results from restricting the maximum amount that an ant's l_{best} tour can be modified whereby it can be observed that by reducing the part of the l_{best} tour that can be modified, the average accuracy deteriorates with respect to the known optimum. A potential reason for this is that the ants become trapped in local optima, unable to improve their l_{best} tour without a greater degree of flexibility in tour construction. However, as expected, reducing the degree to which a l_{best} tour can be modified increases the speed of the approach with up to a 33 fold increase in speed observed when only allowing a maximum 10% of l_{best} tours to be modified at each iteration.

Clearly, restricting the maximum aspect of l_{best} tours that can be modified results in much faster speed but tour quality suffers considerably as a result of being trapped in local optima. A methodology is required to enable an ant to jump out of local optima. In GAs, crossover is used with a given probability so perhaps the same approach will benefit *PartialACO*. Therefore, it is proposed that an additional parameter is introduced defining the probability that an ant will only partially modify its l_{best} tour. In the case an ant does not partially modify its l_{best} tour then it will construct a full tour as standard P-ACO would. Table 4 shows the results from using a probability of an ant only partially modifying its l_{best} tour of 0.95. Comparing to Table 3, improvements in the average tour accuracy to the known optimum of each TSP instance are observed. However, aside from the pcb442 problem, accuracy remains worse than the results shown in Table 2 with no restriction on the degree by which l_{best} tours can be modified. More importantly, a significant reduction in the relative speedups is observed even when using such a small probability of constructing full tours.

4.1 Incorporating Local Search

Given that enabling ants to occasionally construct a full length tour to break out of local optima has some beneficial effect, a better alternative could be considered. Instead of using occasional full length tour construction, a local search heuristic such as 2-opt could be applied to tours with a given probability. Using 2-opt will improve tours and by the swapping of edges between cities at any point of the tour, ants could break out of local optima. To test this theory, standard P-ACO is tested once again this time using a probability of using 2-opt search of 0.001 with the other parameters remaining the same. These results are shown in Table 5 whereby significant improvements in accuracy are

Table 5. Results from using standard P-ACO approach with a probability of 0.001 of using 2-opt local search with accuracy expressed as percentage difference from known optimum. Results are averaged over 100 random runs.

TSP instance	Accuracy (% error)			Execution time (in seconds)
	Average	Best	Worst	
pcb442	3.87 ± 0.39	2.87	4.52	44.67 ± 0.41
d657	4.45 ± 0.30	3.42	5.05	83.97 ± 0.49
rat783	5.20 ± 0.29	4.24	5.83	110.43 ± 0.63
pr1002	5.56 ± 0.32	4.65	6.18	170.48 ± 0.95
pr2392	7.47 ± 0.27	6.50	7.90	834.08 ± 5.71

Table 6. Results from using *PartialACO* approach with a probability of 0.001 of using 2-opt local search. Results are averaged over 100 random runs.

TSP instance	Accuracy (% error)			Execution time (in seconds)	Relative speedup
	Average	Best	Worst		
pcb442	1.64 ± 0.30	0.74	2.32	21.26 ± 0.31	2.10×
d657	2.32 ± 0.31	1.20	3.10	39.65 ± 0.40	2.12×
rat783	3.35 ± 0.36	2.30	4.15	51.82 ± 0.62	2.13×
pr1002	3.40 ± 0.31	2.53	4.04	79.09 ± 0.64	2.16×
pr2393	5.90 ± 0.30	5.01	6.58	377.61 ± 4.76	2.21×

observed over not using 2-opt. However, there is an increase in execution time by as much as 33% as 2-opt is a computationally intensive algorithm of $O(n^2)$ complexity.

Table 6 shows the results of the proposed *PartialACO* approach with the same probability of using 2-opt and no restriction to the portion of an ant's l_{best} tour that can be modified. Improvements in accuracy are observed for all the TSP instances over standard P-ACO. A speedup of a little over two fold is also achieved, slightly less than that when not using 2-opt. This is because a significant amount of computational time is now spent within the 2-opt heuristic reducing the advantage of *PartialACO*.

Given the success of using 2-opt local search with *PartialACO*, the experiments restricting the degree to which an ant can modify its l_{best} tour can be repeated, the results of which are shown in Table 7. Now with 2-opt local search, the accuracies from Table 6 are all improved upon by restricting the degree to which ants can modify their l_{best} tours. The point at which the best accuracy is achieved favours smaller maximum modifications as the problem size increases with only 10% for the pr2392 problem although this still enables partial tour modification of up to 239 cities. A potential reason for the success of restrictive *PartialACO* when using 2-opt local search is that 2-opt derives high quality tours whereby the subsequent iterations by ants are effectively performing a localised search on these tours. The *exploitation* aspect of *PartialACO* stems from exploiting high quality tours which 2-opt local search assists but it should be noted that the speedups are significantly reduced when using 2-opt.

5 Applying *PartialACO* to Larger TSP Instances

Now that the suitability of *PartialACO* has been demonstrated against regular size TSP instances, it will now be tested against four much larger TSP instances with hundreds of thousands of cities. These four large TSP instances are based on famous works of art such as the *Mona Lisa* and the *Girl with a Pearl Earring*. With these TSP instances, the optimal tour when drawn in two dimensions as a continual line will resemble the given famous art work (see Fig. 2).

Table 7. Results from *PartialACO* approach using 0.001 probability of 2-opt and a range of maximum modifications. Accuracy expressed as percentage difference from known optimum with relative speedup. Results averaged over 100 random runs.

TSP instance	Max. modification	Accuracy (% error)			Execution time (in seconds)	Relative speedup
		Average	Best	Worst		
pcb442	50%	1.46 ± 0.28	0.80	2.21	11.49 ± 0.21	3.89×
	40%	1.48 ± 0.29	0.75	2.24	9.74 ± 0.17	4.58×
	30%	1.49 ± 0.33	0.76	2.42	8.15 ± 0.19	5.48×
	20%	1.47 ± 0.32	0.65	2.25	6.63 ± 0.11	6.74×
	10%	1.80 ± 0.37	0.87	2.86	5.12 ± 0.09	8.73×
d657	50%	1.97 ± 0.28	1.13	2.60	21.11 ± 0.27	3.98×
	40%	1.90 ± 0.27	1.24	2.65	18.10 ± 0.30	4.64×
	30%	1.80 ± 0.28	1.22	2.39	15.15 ± 0.29	5.54×
	20%	1.71 ± 0.28	0.94	2.38	12.28 ± 0.22	6.84×
	10%	1.86 ± 0.26	0.86	2.41	9.32 ± 0.22	9.01×
rat783	50%	3.07 ± 0.29	2.16	3.76	27.37 ± 0.42	4.04×
	40%	3.01 ± 0.31	2.00	3.78	23.56 ± 0.35	4.69×
	30%	2.95 ± 0.28	1.99	3.51	19.73 ± 0.36	5.60×
	20%	2.81 ± 0.26	1.90	3.29	16.10 ± 0.34	6.86×
	10%	2.86 ± 0.26	2.06	3.47	12.25 ± 0.28	9.01×
pr1002	50%	2.93 ± 0.31	1.77	3.44	42.47 ± 0.59	4.01×
	40%	2.81 ± 0.32	2.06	3.54	36.40 ± 0.55	4.68×
	30%	2.65 ± 0.27	2.07	3.37	30.86 ± 0.57	5.52×
	20%	2.58 ± 0.29	1.71	3.16	25.42 ± 0.51	6.71×
	10%	2.55 ± 0.28	1.92	3.13	19.62 ± 0.43	8.69×
pr2392	50%	5.46 ± 0.27	4.86	5.99	204.69 ± 3.67	4.07×
	40%	5.38 ± 0.28	4.73	6.06	179.68 ± 3.79	4.64×
	30%	5.13 ± 0.27	4.51	5.84	157.07 ± 3.48	5.31×
	20%	4.86 ± 0.23	4.25	5.28	135.54 ± 3.42	6.15×
	10%	4.45 ± 0.28	3.67	5.06	110.85 ± 2.82	7.52×

As previously, *PartialACO* will be run for 100,00 iterations with 16 ants and results averaged over 10 random runs. Moreover, given the large scale of the problems under consideration, the maximum aspect of the l_{best} tour that can be modified by an ant is now reduced to just 1% of the number of cities which for the Mona Lisa TSP is still 1,000 cities. 2-opt search will be performed with a probability of 0.001. However, 2-opt is a computationally expensive algorithm of $O(n^2)$ complexity. Consequently, 2-opt will be restricted to only considering swapping edges that are within 500 cities of each other in the current tour. This reduces the runtime of 2-opt significantly although slightly reduces its effectiveness.

Fig. 2. Four classical art based TSP instances (downloadable from http://www.math.uwaterloo.ca/tsp/data/art/), (a) da Vinci's *Mona Lisa* (100K cities), (b) Van Gogh's *Self Portrait 1889* (120K cities), (c) Botticelli's *The Birth of Venus* (140K cities) and (d) Vermeer's *Girl with a Pearl Earring* (200K cities)

Table 8. Results from testing *PartialACO* on four art TSP instances. Accuracy expressed as percentage difference from best known tour. Results averaged over 10 runs.

TSP instance	Accuracy (% error)			Execution time (in seconds)
	Average	Best	Worst	
mona-lisa100K	5.45 ± 0.07	5.36	5.58	1.07 ± 0.02
vangogh120K	5.82 ± 0.10	5.70	6.01	1.45 ± 0.03
venus140K	5.81 ± 0.14	5.60	6.05	2.09 ± 0.06
earring200K	7.20 ± 0.18	6.91	7.39	5.06 ± 0.14

The results of executing the *PartialACO* technique on the large scale TSP instances are shown in Table 8 whereby it can be observed that tours with an average error ranging between 5–7% of the best known optima are found. Regarding runtime, *PartialACO* finds these tours with just a few hours of computational time using a single multi-core CPU. To clearly demonstrate the effectiveness of *PartialACO* a comparison is made with the standard P-ACO approach. Given the likely increase in runtime it would not be feasible to execute for 100,000 iterations. Consequently, a time limited approach is used whereby the standard P-ACO approach is run for the same degree of time as the results from Table 8 and the number of iterations achieved recorded which provides a relative

Table 9. Results from testing standard P-ACO against the four large art based TSP instances for the timings reported in Table 8. Accuracy is expressed as the percentage difference from the best known solution. The number of iterations is shown enabling a speedup of *PartialACO* to be ascertained. Results averaged over 10 runs.

TSP instance	Accuracy (% error)			Average iterations	Relative speedup by *PartialACO*
	Average	Best	Worst		
mona-lisa100K	13.50 ± 0.49	12.62	14.46	248.10 ± 4.41	403.06×
vangogh120k	14.04 ± 0.54	13.25	15.07	157.00 ± 1.76	636.94×
venus140k	14.48 ± 2.24	13.04	20.71	105.30 ± 3.77	949.67×
earring200k	16.71 ± 3.25	13.97	21.46	83.40 ± 0.52	1199.04×

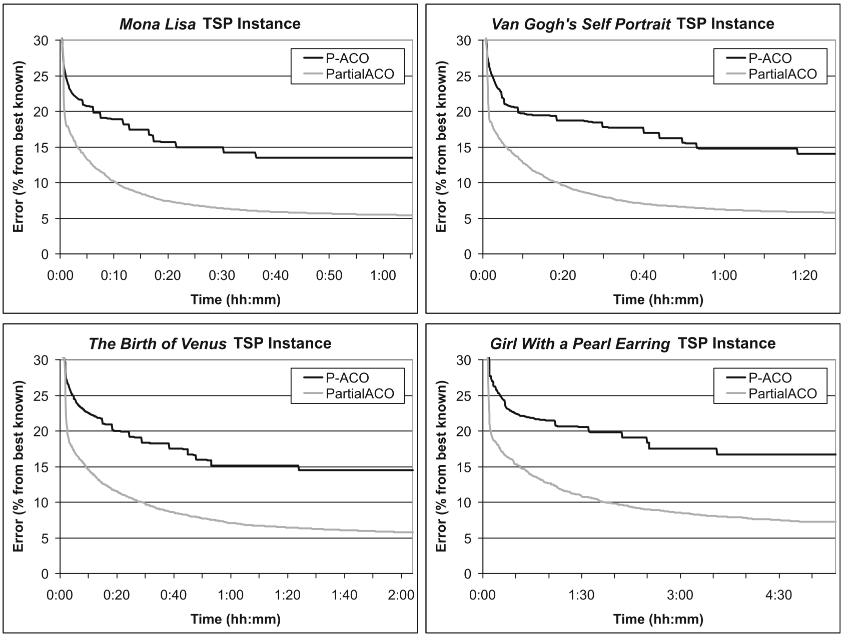


Fig. 3. The average convergence rates over time for the P-ACO and *PartialACO* techniques and each of the art based TSP instances

speedup achieved by *PartialACO*. These results are shown in Table 9 whereby it can be observed that much worse accuracy is achieved which is to be expected as P-ACO only ran for a few hundred iterations. Furthermore, a speedup of up to 1200× is demonstrated by *PartialACO*. To understand this speedup the number of required edge comparisons to unvisited cities to determine which city to visit next by an ant building a tour must be considered. The number of comparisons an ant will make relates to a triangle progression sequence defined by

$(n(n+1))/2$. Thus, an ant building a complete tour for a 100K city problem will perform approximately 5×10^9 edge comparisons. However, if only 1% of a tour is modified an ant will only perform 5×10^5 comparisons, a 10,000 fold efficiency saving. However, not all of this saving will be realised as a result of computational factors such as the use of 2-opt. The 10,000 fold efficiency only applies to the *tour construction* aspect of ACO hence the lower reported speedups.

Further evidence of the effectiveness of *PartialACO* is demonstrated by considering the convergence rates over time for each of the art based TSP instances as shown in Fig. 3. *PartialACO* clearly converges much faster than the P-ACO approach for all four problems. Indeed, inspection of the largest problem instance, *The Girl With the Pearl Earring*, shows that the *PartialACO* technique achieves the same accuracy in minutes that P-ACO takes hours to achieve. This convergence speed is simply as a result of *PartialACO* being able to perform many more iterations of the ant tours, indeed thousands, in a short space of time. In fact, it can be argued that in terms of iterations *PartialACO* converges slower but given that time is a more important factor, *PartialACO* is the better approach.

6 Related Work

It is acknowledged that ACO is computationally intensive. Indeed, even the original author of ACO was aware of the computational complexity proposing a variant known as Ant Colony System (ACS) [7] whereby the neighbourhood of unvisited cities is restricted. A *candidate list* approach is used whereby at each decision point made by an ant, only the closest cities are considered. If these have already been visited then normal ACO used. This approach significantly reduces the computational complexity. ACS is also similar to *PartialACO* in that with a high probability an ant takes the edge with the greatest level of combined pheromone and heuristic information improving the speed. However, ACS still requires a full pheromone matrix and to needs to search for the edge with the greatest level to choose the next city to visit.

The main area of research into speeding up ACO though has been through parallel implementations. ACO is naturally parallel such that ants can construct tours simultaneously. Early works such as Bullnheimer et al. [2], Delisle et al. [6] and Randall and Lewis [13] relied on distributing ants to processors using a master-slave methodology. In recent years the focus on speeding up ACO has been on utilising Graphical Processor Units (GPUs) consisting of thousands of SIMD processors. Bai et al. were the first to implement MAX-MIN ACO for the TSP on a GPU achieving a 2.3× speedup [1]. More notable works include DeléVacq et al. who compare parallelisation strategies for MAX-MIN ACO on GPUs [5], Cecelia et al. who present an *Independent Roulette* approach to better exploit data parallelism for ACO on GPUs [3] and Dawson and Stewart who introduce a *double spin* ant decision methodology when using GPUs [4]. However, ACO is not ideally suited to GPUs and these papers can only report speedups ranging between 40–80× over a sequential implementation.

7 Conclusions

This paper has addressed the issues associated with applying ACO to large scale TSP instances, namely reducing memory constraints and substantially increasing execution speed. A new variant of ACO was introduced, *PartialACO*, based upon P-ACO which dispenses with the pheromone matrix, the memory overhead. Moreover, *PartialACO* only partially modifies the best tour found by each ant akin to crossover in GAs. *PartialACO* was demonstrated to significantly improve the computational speed of ACO and the accuracy by reducing the computational complexity and the probabilistic chance of ants making poor choices of cities to visit. Consequently, *PartialACO* was applied to large scale TSP instances of up to 200K cities achieving accuracy of 5–7% of the best known tours with a speed of up to 1200 times faster than that of standard P-ACO. *PartialACO* is a first step to deploying ACO on large scale TSP instances and further work is required to improve its accuracy to compete with a GA approach [12] although it should be noted that this work uses a supercomputer. Further analysis of the parameters balancing speed vs. accuracy could help to improve the technique such as reducing the maximum permissible modification of tours for speed and increasing the iterations. Moreover, a dynamic approach may be best whereby initially only small modifications are allowed but as time progresses the permissible modification increases to avoid being trapped in local optima.

References

1. Bai, H., OuYang, D., Li, X., He, L., Yu, H.: MAX-MIN ant system on GPU with CUDA. In: 2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC), pp. 801–804. IEEE (2009)
2. Bullnheimer, B., Kotsis, G., Strauß, C.: Parallelization strategies for the ant system. In: High Performance Algorithms and Software in Nonlinear Optimization, pp. 87–100. Springer (1998)
3. Cecilia, J.M., García, J.M., Nisbet, A., Amos, M., Ujaldón, M.: Enhancing data parallelism for ant colony optimization on GPUs. J. Parallel Distrib. Comput. **73**(1), 42–51 (2013)
4. Dawson, L., Stewart, I.: Improving ant colony optimization performance on the GPU using CUDA. In: 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 1901–1908. IEEE (2013)
5. DeléVacq, A., Delisle, P., Gravel, M., Krajecki, M.: Parallel ant colony optimization on graphics processing units. J. Parallel Distrib. Comput. **73**(1), 52–61 (2013)
6. Delisle, P., Krajecki, M., Gravel, M., Gagné, C.: Parallel implementation of an ant colony optimization metaheuristic with OpenMP. In: Proceedings of the 3rd European Workshop on OpenMP (EWOMP01), Barcelona, Spain (2001)
7. Dorigo, M., Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput. **1**(1), 53–66 (1997)
8. Dorigo, M., Stützle, T.: Ant Colony Optimization. Bradford Company, Scituate (2004)

9. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS 1995, pp. 39–43. IEEE (1995)
10. Guntsch, M., Middendorf, M.: A population based approach for ACO. In: Workshops on Applications of Evolutionary Computation, pp. 72–81. Springer (2002)
11. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. MIT Press, Cambridge (1975)
12. Honda, K., Nagata, Y., Ono, I.: A parallel genetic algorithm with edge assembly crossover for 100,000-city scale TSPs. In: 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 1278–1285. IEEE (2013)
13. Randall, M., Lewis, A.: A parallel implementation of ant colony optimization. J. Parallel Distrib. Comput. **62**(9), 1421–1432 (2002)

A New Steady-State MOEA/D for Sparse Optimization

Hui Li[1](✉), Jianyong Sun[1], Yuanyuan Fan[1], Mingyang Wang[1], and Qingfu Zhang[2]

[1] School of Mathematics and Statistics, Xi'an Jiaotong University,
Xi'an 710049, Shaanxi, China
lihui10@xjtu.edu.cn

[2] Department of Computer Science, City University of Hong Kong,
Tat Chee Avenue, Kowloon, Hong Kong

Abstract. The classical algorithms based on regularization usually solve sparse optimization problems under the framework of single objective optimization, which combines the sparse term with the loss term. The majority of these algorithms suffer from the setting of regularization parameter or its estimation. To overcome this weakness, the extension of multiobjective evolutionary algorithm based on decomposition (MOEA/D) has been studied for sparse optimization. The major advantages of MOEA/D lie in two aspects: (1) free setting of regularization parameter and (2) detection of true sparsity. Due to the generational mode of MOEA/D, its efficiency for searching the knee region of the Pareto front is not very satisfactory. In this paper, we proposed a new steady-state MOEA/D with the preference to search the region of Pareto front near the true sparse solution. Within each iteration of our proposed algorithm, a local search step is performed to examine a number of solutions with similar sparsity levels in a neighborhood. Our experimental results have shown that the new MOEA/D clearly performs better than its previous version on reconstructing artificial sparse signals.

Keywords: Sparse optimization · Multiobjective optimization · Evolutionary algorithm · MOEA/D

1 Introduction

According to the compressive sensing theories [1], a sparse signal $x \in R^N$ can be reconstructed from a observation $y \in R^M$ $(M \ll N)$ by solving the following sparse optimization problem:

$$\min ||x||_0, \quad \text{subject to } y = Ax + \sigma \tag{1}$$

where

- N is the length of signal, and M is the number of observations;

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_10

- $A \in R^{M \times N}$ is a sensing matrix with $M \ll N$.
- $||x||_0$ denotes the l_0-norm of x, namely the number of the nonzero components of x.
- $\sigma \in R^N$ represents the noise level.

If the optimal solution x^* of (1) has k nonzero components, then it is called k-sparse. In this work, we mainly focus on the noiseless case, where σ equals to zero.

So far, the most popular sparse optimization methods for solving (1) use the penalty function methods to deal with the constraints in (1). In these methods, a regularization parameter is needed to strike the balance between the sparse term $||x||_0$ and the loss term $||y-Ax||_2^2$. The following regularization-based model should be minimized.

$$\min_{x \in R^N} ||y - Ax||_2^2 + \lambda ||x||_0 \tag{2}$$

where $\lambda(>0)$ is the regularization parameter.

The problem (2) is called l_0 regularization framework. It can be proved that the l_0 regularization problem is NP-hard [2]. The representative algorithms for l_0 problem include greedy algorithms [3,4] like MP and OMP, as well as iterative hard thresholding algorithm [6]. Note that the greedy methods can only find the approximate solution for the low-dimensional l_0 problem. In recent years, some iterative thresholding methods based on convex or nonconvex relaxation have received more attention due to their abilities in precise reconstruction or robustness in noise. The popular relaxation methods based on regularization are soft iterative thresholding method [7] and half iterative thresholding method [8], which use l_1-norm $\|x\|_1 = |x_1| + \cdots + |x_N|$ and $l_{0.5}$-norm $\|x\|_{0.5}^{0.5} = (\sqrt{|x_1|} + \cdots + \sqrt{|x_N|})^2$ to replace l_0-norm in (2). In the existing sparse optimization methods based on regularization, one of the major difficulties is the setting of regularization parameter λ. For the relaxation-based regularization problems, the sparse solution found by a large value of λ could be very sparse and has large error value, while that found by a small value of λ is not sparse enough. In practice, the cross validation method is often adopted in the setting of regularization. That is, multiple values of λ are individually considered in optimization. Moreover, the setting of λ may also need to know the true value of k or its estimation value. In fact, the prior knowledge on k is often unknown in advance.

To deal with the above difficulties, several multiobjective optimization algorithms have been developed for sparsity optimization [10,11]. In these algorithms, the following bi-objective optimization problem should be taken into consideration:

$$\min_{x \in R^N} F(x) = (f_1(x) = ||x||_0, f_2(x) = ||y - Ax||_2^2) \tag{3}$$

Note that the formulation of Problem (3) doesn't involve λ. The weakly Pareto front (PF) of (3) is plotted in Fig. 1. It is easy to show that the k-sparse solution is located at the knee part of the whole weakly Pareto front. It is also one of the Pareto solutions with $f_2 = 0$ (i.e., $y = Ax$). It is reasonable to believe that the knee region should be examined by multiobjective methods with more preference.

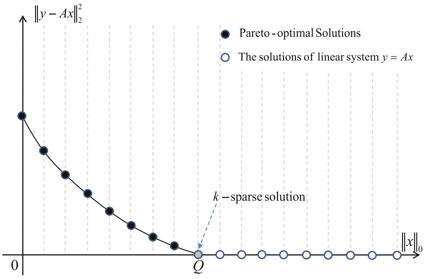


Fig. 1. The graphical illustration of the distribution of weakly Pareto front and the location of k-sparse solution. The solid dots stand for all Pareto solutions while the circles represents the solutions of $y = Ax$.

To approximate the knee part of the weakly PF, a multiobjective sparse optimization algorithm based on decomposition framework, i.e., MOEA/D [5], was proposed. It decomposes Problem (3) into multiple subproblems. Each of these subproblems has its own sparsity levels. During the search process, all subproblems are optimized by one of existing iterative thresholding methods in parallel. To bias the search on the knee region, the sparsity levels of subproblems are adjusted by changing the boundary sparsity levels. In sparse signal recovery, MOEA/D has shown its advantages for detecting the sparsity. However, its performance in solution precision is not competitive to those iterative thresholding algorithms, such as hard and half. It should also be mentioned that the multiobjective algorithm proposed in [10] approximates the whole PF based on the framework of NSGA-II. A spline curve fitting method is applied to find the knee solution in the PF. Compared with MOEA/D, NSGA-II could waste a lot of computational efforts for searching the regions far from the k-sparse solution.

In fact, the parallel searching mechanism in MOEA/D could also cause much waste in computational load. As a result, the knee solution found by MOEA/D is of low precision. In this paper, we present a new steady-state multiobjective algorithm for sparse optimization problems, named MOEA/D-II, in which only one solution or subproblem close to the knee region is selected for local search in each iteration. Our experimental results show that MOEA/D-II performs much better MOEA/D in precision. The rest of this paper is organized as follows. In Sect. 2, we briefly review MOEA/D for sparse optimization. Section 3 presents MOEA/D-II in detail. The results of experimental studies are reported in Sect. 4. The final section concludes this paper.

2 MOEA/D for Sparse Optimization

2.1 Problem Decomposition

The core idea in MOEA/D is to decompose Problem (1) into a number of single objective subproblems. The formulations of subproblems can be stated as follows:

$$\min_{x \in R^N} \|y - Ax\|^2 \quad \text{subject to } \|x\|_0 = k_i, i = 1, \ldots, S \tag{4}$$

where the sparsity level k_i is between k_{min} and k_{max}, which are set to $0.5 \times k$ and $2 \times k$ respectively in this work. S is the number of subproblems.

In each generation MOEA/D, all subproblems are optimized by one of existing iterative thresholding methods in parallel. After each generation, the sparsity levels of boundary subproblems corresponding to $\min k_i$ or $\max k_i$ are adaptively changed.

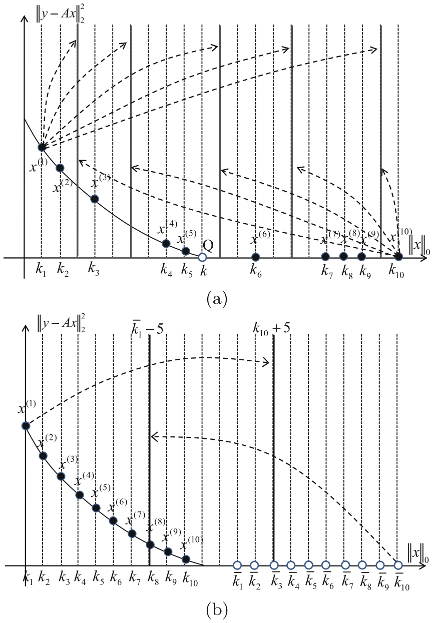
2.2 Framework of MOEA/D

The major data structures needed in MOEA/D are (1) a set of S sparsity levels $K = \{k_1, \ldots, k_S\}$ and (2) a population of S solutions $P = \{x^1, \ldots, x^S\}$.

The main procedures of MOEA/D are stated in Algorithm 1.

Algorithm 1. MOEA/D - approximating the weakly PF in a parallel mode

```
Step 1: Initialization
    Generate an initial set $T$ of $S$ sparsity levels, and produce a population $P$ of $S$ corresponding sparse solutions $x^i, i = 1, \ldots, S$ subject to $||x^i||_0 = k_i$.
Step 2: Local Search
    Improve all the solutions in $P$ by applying a certain iterative thresholding methods.
Step 3: Update of Sparsity & Solution
  – Step 3.1: Replace the boundary sparsity levels.
      Step 3.1.1: Sort the sparsity levels in $K$ by ascending order, i.e., $k'_1 \leq k'_2 \cdots \leq k'_S$. Find all the sparsity gaps $[k'_i, k'_{i+1}]$ with $k'_{i+1} - k'_i > 1, i \in 1, \ldots, S - 1$ and save them into $G$;
      Step 3.1.2: Compute the proportion $\alpha$ of non-dominated solutions in $P$;
      Step 3.1.3: If $G$ is not empty, go to Step 3.1.4, otherwise, go to Step 3.1.5;
      Step 3.1.4: Choose a sparsity level $k'$ in one gap of $G$ randomly. If $\alpha > 50\%$, let $k'_1 = k'$, otherwise, $k'_S = k'$;
      Step 3.1.5: If $\alpha > 50\%$, let $k'_1 = k'_S + 0.5 \times S$; otherwise $k'_S = k'_1 - 0.5 \times S$;
  – Step 3.2: Choose one solution $x'$ in $P$ that satisfies $f_2(x') < \frac{1}{2}(f_2^{min} + f_2^{max})$ to replace the solution corresponding to the new sparsity level.
Step 4: Stopping Criteria
    If a stopping criteria is reached, then stop and output the population $P$, otherwise, go to Step 2.
```



(a)

(b)

Fig. 2. The adjustment of sparsity levels in MOEA/D by two cases: (a) all sparsity levels are not consecutive (G is not empty); (b) all sparsity levels are consecutive (G is empty).

In the original MOEA/D, the iterative hard thresholding method [6] is adopted for the optimization of every subproblem in local search. It should be noted that all subproblems have different difficulties in minimizing the objective function $\|y - Ax\|^2$ due to the different setting of k_i. If all subproblems consume the same amount of iterations in thresholding method, the objective function values of some solutions with small sparsity levels will be decreased faster than those with large sparsity levels. It is reasonable to deduce that some solutions with small sparsity levels are located in the PF part and dominate those with large sparsity values. Therefore, the subproblems with both the maximum and the minimum sparsity levels should be changed.

The key steps for tuning sparsity levels lie in Step 3.1.4 and Step 3.1.5 of Algorithm 1, which are illustrated in Fig. 2. In the former step, a random sparsity level in the gap of two neighboring sparsity levels is selected to replace the minimum or the maximum of sparsity levels in K according to the value of α. When a majority of solutions in P are nondominated ($\alpha > 50\%$), the minimal sparsity level k'_1 is replaced. Otherwise, the maximal sparsity level k'_S is replaced. In the latter step, the boundary sparsity levels are moved towards their opposite direction when all sparsity levels are consecutive.

In Step 3.2, the solution associated with the modified sparsity level is replaced by one solution with the objective function values f_2 not too big. This strategy is helpful in encouraging the minimization of f_2.

3 MOEA/D-II

In this section, the main idea and the algorithmic framework of MOEA/D-II are presented in detail.

3.1 Main Idea

By analyzing the solutions in the final population found by MOEA/D, it is easy to observe that the solutions in weakly PF with similar sparsity levels are also similar in the decision space. More precisely, they only differ in a few components. This means it is not necessary to optimize multiple subproblems with similar sparsity levels in a parallel way. Instead, one solution among them is selected to be further improved by existing iterative thresholding methods. A notion called sparsity neighborhood is defined to diversify the search along the weakly PF.

Given a sparse solution x with $\|x\|_0 = \bar{k}(0 < \bar{k} < N)$, the sparsity neighborhood of x is defined as an interval of sparsity levels, i.e., $[\bar{k} - r, \bar{k} + r]$, where the setting of the radius r often relies on the length of signal N. From Fig. 1, the sparsity neighborhood of the k-sparse solution has a quite special distribution. On the one hand, the function values of f_2 for all solutions with sparsity levels in its left sparsity neighborhood $[\bar{k} - r, \bar{k}]$ are monotonically decreased as the sparsity level increases. On the other hand, all the solutions with sparsity levels in its right sparsity neighborhood $[\bar{k}, \bar{k} + r]$ have similar function values in f_2, which are very close to zero.

Based on the knee feature discussed above, we design a new version of MOEA/D with steady-state search strategy. In each iteration, a certain ITH method is used to improve a starting solution based on a sparsity value in its left neighborhood. If the f_2 value of the starting solution doesn't decrease, then the starting solution is further improved by the ITH method in its right neighborhood. In this way, the left neighborhood of any starting solution in local search is preferred during the improvement by the ITH methods (Fig. 3).

3.2 Framework of MOEA/D-II

Our proposed algorithm, MOEA/D-II, maintains two populations: $P1$ - the weakly Pareto solutions with small values of f_2; and $P2$ - the non-dominated solutions with large values of f_2. The main steps of MOEA/D-II are described in Algorithm 2:

The step of initialization includes Step 1.1 - the generation of initial population and sparsity levels, and Step 1.2 - applying local search for the non-dominated members of P. Since the initial solutions are generated randomly, the function values of these solutions on the loss term are usually big. The solutions obtained in Step 1.2 are of good quality regarding the loss term. To speed up the searching for 'knee' point, MOEA/D-II only selects one solution from $P1$ or $P2$ at each iteration. The detail of selecting solution for local search is implemented in Step 2.1 and Step 2.2, where the solutions close to the knee

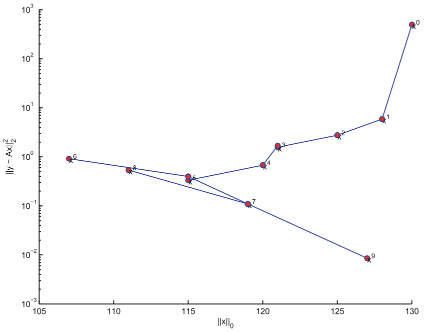


Fig. 3. An illustration of local search in MOEA/D-II. The left neighborhood is preferred in the beginning of local search since the f_2 value of starting solution can be easily improved.

Algorithm 2. MOEA/D-II with preference on the knee region

```
Step 1: Initialization
    Step 1.1: Generate $S$ initial sparsity levels $\{k_1, k_2, \ldots, k_S\}$ with $k_1 < k_2 < \ldots < k_S$ and corresponding sparse solutions $P = \{x^1, x^2, \ldots, x^S\}$. Let $P1 = P2 = \varnothing$.
    Step 1.2: For each non-dominated member of $P$, a certain ITH method is used to improve it based on a sparsity level in its left sparsity neighborhood and update $P1$ and $P2$ with the improved solution.
Step 2: Selection for Local Search
    Step 2.1: Select the solution with the minimal function value regarding loss term in $P1$ or the solution with the minimal value regarding sparsity levels in $P2$, denote the selected solution as $x_{ls}$.
    Step 2.2: If $x_{ls}$ is selected from $P2$, then mutate $x_{ls}$.
Step 3: Local Improvement with Thresholding
    Step 3.1: Select a sparsity level $k_l$ in the left sparsity neighborhood of $x_{ls}$, then apply the ITH method to improve $x$ based on $k_l$. The resultant solution is denoted as $x_l$.
    Step 3.2: If $f_2(x_l) > f_2(x_{ls}) + \beta$, then go to Step 3.3; otherwise, set $x' = x_l$ and go to Step 4.
    Step 3.3: Select a new sparsity level $k_r$ in the right sparsity neighborhood of $x_{ls}$, then apply the ITH method to improve $x_{ls}$ with $k_r$. The resultant solution $x_r$ is set to $x'$.
Step 4: Population Update
    Step 4.1: Let $f_2^{min}$ denote the minimal function value regarding the loss term in $P1$. If $f_2(x') < f_2^{min} + \beta$, add $x'$ to $P1$.
    Step 4.2: For two solutions in $P1$ with the same sparsity level, the one with bigger value regarding loss term is removed. When the number of solutions in $P1$ is more than the limit $S_{max}$, then the solution with the maximal sparsity level is removed.
    Step 4.3: If $x'$ is added into $P1$ or there are not solutions with smaller sparsity levels in $P1$, then go to Step 4.4; otherwise, go to Step 5.
    Step 4.4: If no solution in $P2$ dominates $x'$, then add $x'$ into $P2$ and remove the solutions dominated by $x'$. When the number of $P2$ is more than the limit $S_{max}$, remove the solution with the minimal sparsity level.
Step 5: Stopping Criteria
    If a stopping criteria is reached, then stop and output the population $P1$ and $P2$, otherwise, go to Step 2.
```

region are preferred. As we discussed above, the local search based on the left sparsity neighborhood is performed with a priority. If no improvement is made regarding the loss term, then the local search is performed for the right sparsity neighborhood. The major steps are illustrated in Steps 3.1–3.3. In Step 3.2, the parameter β is a small positive number. When $f_2(x_l) > f_2(x_{start}) + \beta$ holds, the local search fails to make an improvement.

Note that x_l or x_r generated in local improvement should be used to update $P1$ or $P2$. To encourage the convergence regarding the loss term, the update of $P1$ is performed before the update of $P2$. Once a solution is added into $P1$, it is no longer considered for the update of $P2$. The detailed step for updating populations by a certain solution x' are described as follows:

3.3 The Connections with MOEA/D

MOEA/D-II can be regarded as an improved version of MOEA/D for sparse optimization. This is because both algorithms decompose the Problem (3) into several subproblems and optimize them with preference on the knee region. Any existing ITH methods for sparse optimization can be applied in the two algorithms.

Compared with MOEA/D, the local search in MOEA/D-II is performed within sparsity neighborhood. At each iteration, MOEA/D-II uses steady-state strategy to approximate the weakly PF. In this way, MOEA/D-II consumes much less computational effort on searching the weakly PF part far from the knee region.

4 Computational Experiments

In this section, we conduct some experiments to compare the performance of MOEA/D and MOEA/D-II for sparse optimization on artificial sparse signals. Both algorithms are implemented in Matlab R2014a on the PC with the Intel Core i5-3220M CPU @2.60 GHz and 4 GB memory running Windows 8.1 operating system.

4.1 Experimental Setting

As suggested in [8], we generated six test problems P1–P6 shown in Table 1. The length of signals in P1–P4 are 512, and the true sparsity level is 130. The number of observations in these four test problems are 300, 270, 250, 240 respectively. P5 and P6 are two test problems with long signal, of which the configuration is 5 times and 10 times of P2. For each problem, we generate 100 test instances in the following way.

- Given a sparsity k, generate a true sparse solution x^* with $\|x^*\|_0$. Its nonzero components are sampled by the normal distribution with the mean - 0 and the std - 2.

Table 1. The setting of the test problems, N is the length of signal and M is the number of observations. T is the true sparsity value.

Problem sets	N	M	T
P1	512	300	130
P2	512	270	130
P3	512	250	130
P4	512	240	130
P5	2560	1500	650
P6	5120	3000	1300

- Generate a sensing matrix by setting $A = randn(M, N)$. Then, A is orthogonalized.
- Compute the observation vector $y = Ax^*$.

In MOEA/D, the hard ITH method is used for local search. Two versions of MOEA/D-II based on the hard ITH method and the half ITH method, denoted by MOEA/D-II/L0 and MOEA/D-II/L0.5, are involved in comparison. The major experimental parameters in MOEA/D-II are as follows:

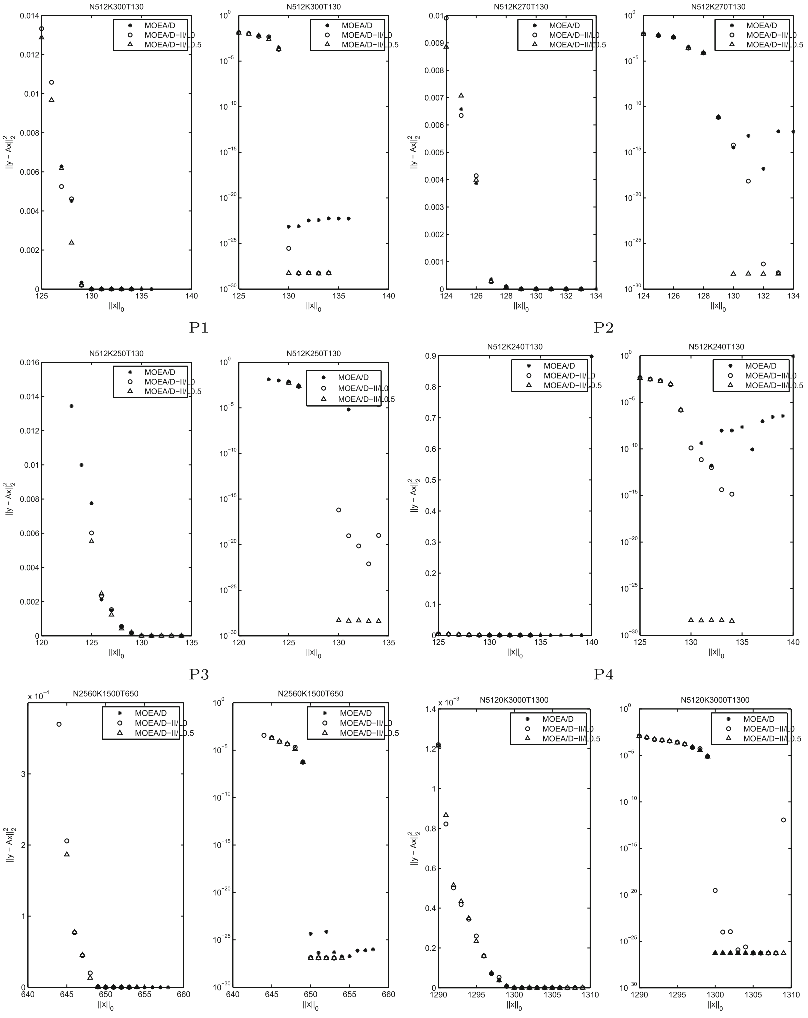
- $S = S_{max} = 10$ - the initial size of population and the maximal size of population;
- $ls = 10$ - the descending step of the ITH method in local search;
- $\beta = 0.01$ and the sparsity neighborhood size is set to 5;
- $c_{max} = 3000$ - the maximal number of iterations.

To measure the performance of MOEA/D and MOEA/D-II, the successful rate is computed by counting the number of successful runs. One successful run means the mean square error (MSE) between the solution found by the algorithm and the true sparse solution is less than a given precision. In this work, the precision is set to 10^{-6}.

4.2 Experimental Results

Figure 4 shows the PFs found by in one of its typical run on P1–P6 found by the three algorithms. We can see that all three algorithms are able to find the 'knee' region of the weakly PFs on four small scale problems P1–P4. Compared with MOEA/D, two versions of MOEA/D-II can perform better in solution precision. From the distribution of the final solutions found by MOEA/D and MOEA/D-II, it is easy to see that the location with true sparsity value is included in the final population.

The results plotted in Fig. 4 also indicate that both MOEA/D and MOEA/D-II can find the knee position for two large scale problems P5 and P6. But the



P1

P2

P3

P4

P5

P6

Fig. 4. The weakly Pareto solutions found by MOEA/D, MOEA/D-II/L0, MOEA/D-II/L0.5 on P1–P6

precision of the final solutions found by MOEA/D-II is not clearly better than that found by MOEA/D on these two problems. This can be explained by the use of the small value of sparsity neighborhood size is not suitable for dealing with large scale problem.

The convergence speed of the three algorithms in terms of MSE values on P1 is shown in Fig. 5. It is clear that the convergence speed of MOEA/D-II is much faster than the original MOEA/D. The errors of the signals found by

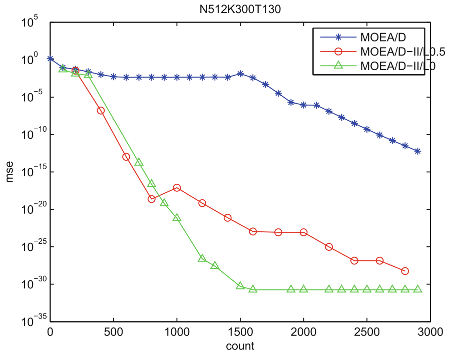


Fig. 5. The convergence speed of three algorithms

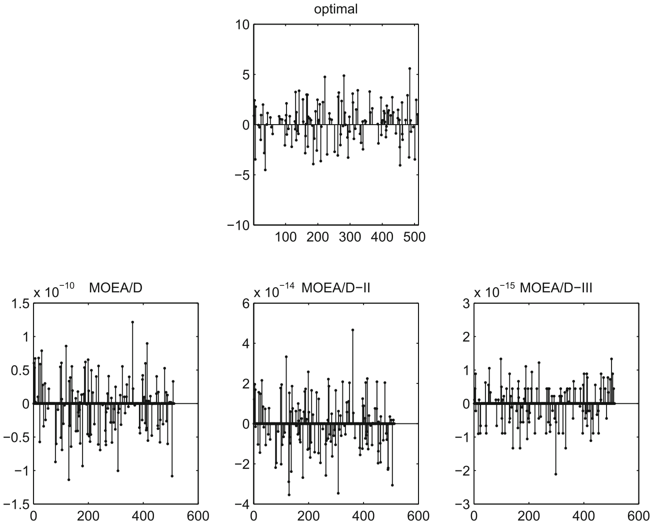


Fig. 6. The error of signals reconstructed by MOEA/D, MOEA/D-II/L0.5, MOEA/D-II/L0 from P1

three algorithms on P1 are plotted in Fig. 6. From this figure, we can observe that MOEA/D-II can reconstruct the signals with much less error values on P1.

Figure 7 plots the successful rates of P1–P6 in the 100 runs. It is clear that MOEA/D-II can improve the successful rate of MOEA/D on P1–P4, where the length of observation signal is relatively short. In this case, MOEA/D performs very badly while two versions of MOEA/D-II perform much better. For two large scale problems P5 and P6, the success rates found by MOEA/D-II are not clearly better than those found by MOEA/D.

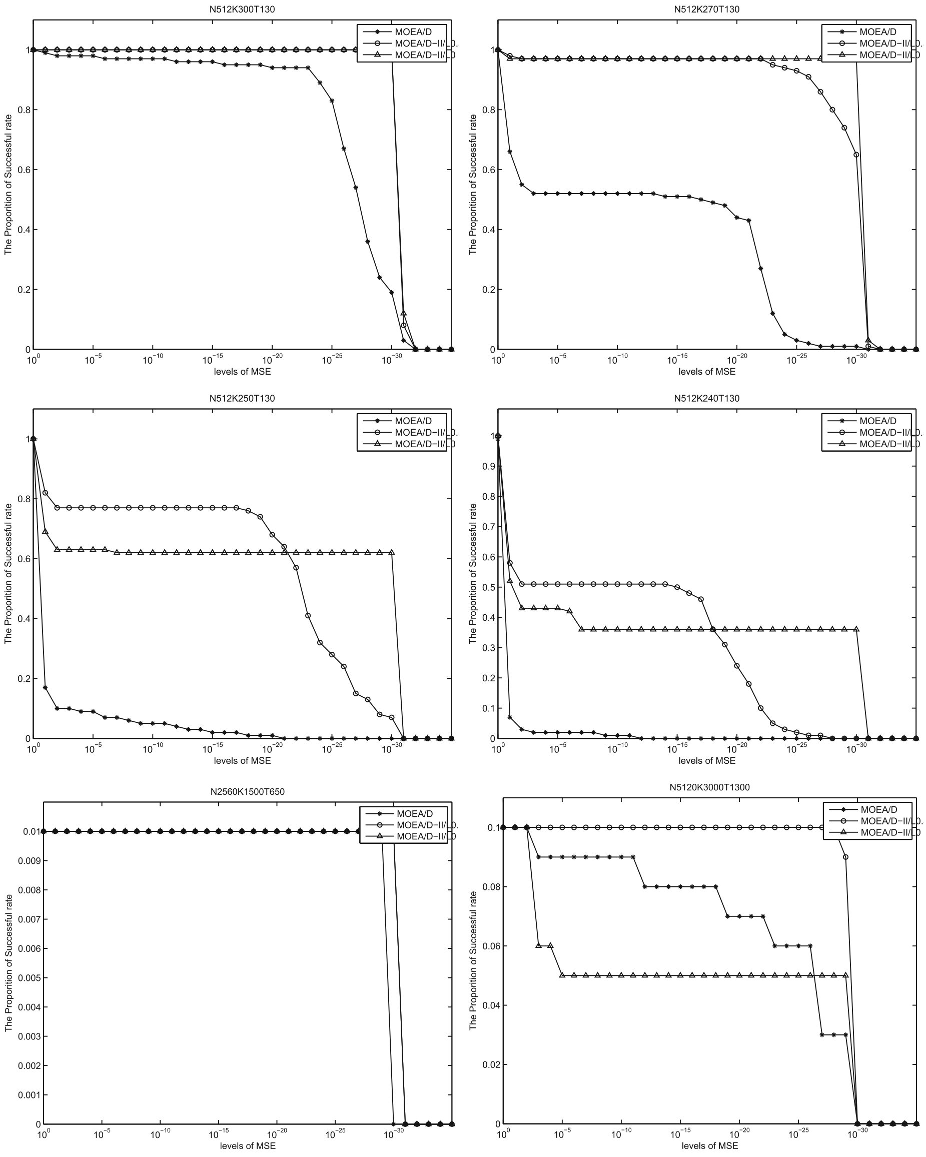


Fig. 7. The successful rates of MOEA/D, MOEA/D-II/L0,MOEA/D-II/L0.5 on P1–P6

5 Conclusion

Multiobjective methods for sparse optimization have the advantages since they do not need to set a regularization parameter or estimate the true sparsity. The true sparse solution is often located at the ‘knee’ position of the weakly

PF. MOEA/D consumes a lot computational cost due its parallel search mode. To accelerate the searching speed and improve successful rate, we proposed a new algorithms called MOEA/D-II, which uses steady-state search mode and performs local search within a sparsity neighborhood. Our experimental results on artificial sparse signals show that the new algorithm is efficient to recovery the sparse signal with better solution precision. Our future work will focus on the use of our new algorithm for the sparse optimization with noise.

Acknowledgments. The authors would like to thank the anonymous reviewers for their constructive comments and suggestions on the original manuscript. This work was supported by the National Science Foundation of China under Grant 61573279, Grant 61175063, and Grant 61473241, Grant 11626252, Grant 11690011, the National Basic Research Program of China under Grant 2017CB329404.

References

1. Donoho, D.: Compressed sensing. IEEE Trans. Image Process. **52**(4), 1289–1306 (2006)
2. Natraajan, B.: Sparse approximation to linear systems. SIAM J. Comput. **24**(2), 227–234 (1995)
3. Davis, G., Mallat, S., Avellaneda, M.: Adaptive greedy approximations. Constr. Approx. **13**(1), 57–98 (1991)
4. Temlyakov, V.: The best m-term approximation and greedy algorithms. Adv. Comput. Math **8**(3), 249–265 (1998)
5. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
6. Blumensath, T., Davies, M.: Normalized iterative hard thresholding: guaranteed stability and performance. IEEE J. Sel. Top. Sign. Process. **4**(2), 298–309 (2010)
7. Donoho, D.: De-noising by soft-thresholding. IEEE Trans. Inf. Theory **41**(3), 613–627 (1995)
8. Xu, Z., Chang, X.Y., Xu, F., Zhang, H.: L1/2 regularization: a thresholding representation theory and a fast solver. IEEE Trans. Neural Netw. Learn. Syst. **23**(7), 1013–1027 (2012)
9. Zeng, J., Lin, S., Wang, Y., Xu, Z.: L1/2 regularization: convergence of iterative half thresholding algorithm. IEEE Trans. Sig. Process. **62**(9), 2317–2329 (2014)
10. Li, L., Yao, X., Stolkin, R., Gong, M., He, S.: An evolutionary multiobjective approach to sparse reconstruction. IEEE Trans. Evol. Comput. **18**(6), 827–845 (2014)
11. Li, H., Su, X., Xu, Z., Zhang, Q.: MOEA/D with iterative thresholding algorithm to sparse optimization problems. In: Proceedings of 12th International Conference on Parallel Problem Solving from Nature (PPSN), pp. 93–101 (2012)

A Multiobjective Evolutionary Algorithm Approach for Map Sketch Generation

Şafak Topçu[1,3](✉) and A. Şima Etaner-Uyar[2]

[1] MEF University, Istanbul, Turkey
topcus@mef.edu.tr,topcus@itu.edu.tr
[2] Istanbul Technical University, Istanbul, Turkey
etaner@itu.edu.tr
[3] Graduate School of Science Engineering and Technology,
Istanbul Technical University, Istanbul, Turkey

Abstract. In this paper, we present a method to generate map sketches for strategy games using a state of the art many-objective evolutionary algorithm, namely NSGAIII. The map sketch generator proposed in this study outputs a three objective Pareto-front in which all the points are fair and strong in different aspects. The generated map sketch can be used by level designers to create real time strategy maps effectively and/or help them see multiple aspects of a game map simultaneously. The algorithm can also be utilised as a benchmark generator to be used in tests for various cases such as shortest path algorithms and strategy game bots. The results reported in this paper are very promising and promote further study.

Keywords: Procedural content generation · Games · Computational intelligence · Map sketch generation · Evolutionary multiobjective optimization · NSGAIII

1 Introduction

The rapidly growing market size of the video game industry, also increased investments in games. Each year, high-budget corporations release many games to get their share from this huge market. With increasing quantity, now players have more options than ever, thus their expectations from the games are getting higher.

To be able to meet player expectations, game studios need to produce new games with new standards. Some offer high quality graphics, some offer high-budget sounds and music etc. While this is attainable for high-budget game studios, indie game developers suffer from it and consequently use a lot of procedurally generated content in their games.

Procedural content generation (PCG) approaches are widely used in the video game industry for several reasons, such as increasing re-playability, lowering game size, speeding up development time etc. Even in some cases PCG methods

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_11

are a necessity. For example Rogue (1980) used procedurally generated dungeons to save valuable memory space. Hello Games' No Man's Sky[1] (2016) generated its entire game universe with PCG methods to save huge design and development time which exceed human life span.

PCG techniques can be used to generate almost every component of a game from *necessary content* (levels, maps, mechanics, story etc.) to *optional content* (weapons, buildings etc.) as is stated in the study of Togelius et al. [10]. In this paper, we focus on generating game map sketches, which are low-resolution representatives of actual game maps for strategy games such as Starcraft.[2] We present a MultiObjective Evolutionary Algorithm (MOEA) approach for generating such map sketches for strategy games. Our aim is to obtain optimal map sketches with respect to three objectives, which cover several aspects of map design to increase playability and game quality. The map sketch generator proposed in this study outputs a three objective Pareto-front in which all the points are fair and strong in different aspects, providing a selection of good quality map sketches to a game designer. Through the experiments in the paper, we also investigate how MOEAs can be applied to the map sketch generation problem. Our motivation is to show that MOEAs are decent alternatives to aggregation functions that are widely used in the area. In fact they may be better choices, because they are able to optimise multiple aspects of a problem simultaneously while generating a large variety of options to the designer.

As the MOEA of choice, in this study, we use NSGAIII introduced by Himanshu Jain [3,6] as an extended version of NSGAII [4]. Six criteria for a good quality map sketch and a direct representation for the solution candidates in the MOEA are modified and adapted from the work proposed by Antonios Liapis [7]. Also as part of our novel contributions in this study, we propose new genetic operators such as the *gaussian swap mutation* and the *one line matrix crossover* that work well with the chosen representation. We conduct experiments to explore the behaviour of the proposed approach as well as show how it scales up as game sizes increase.

The rest of the paper is organised as follows: Sect. 2 examines related work from literature in detail. Section 3 presents the proposed approach and shows how the genetic operators behave. Experimental results are given and discussed in Sect. 4. Section 5 concludes the paper with a brief summary and presents future research directions.

2 Related Work

PCG methods explore a high variety of possible contents while optimising important aspects of a game content, thus increasing both quantity and quality. In this section, we will focus on related work in two parts. In the first part, we will focus on PCG studies that formed a starting point for our study. Next, we will focus

[1] https://www.nomanssky.com.

[2] http://us.blizzard.com/en-us/games/sc/.

on a specific study on map sketch generation [7] because we adapt its solution candidate representation and six objective functions for our study.

2.1 Procedural Content Generation

First set of three studies we should discuss are Togelius et al.'s publications [9,11,12] about multiobjective generation of game maps. In this series of studies, Togelius et al. investigate MOEAs and their behaviour for game maps, particularly for real-time strategy games. In the first two studies [9,12], the authors propose many objective functions and do simultaneous optimisation of a selected two. In the third study [11], the authors expand their study by employing a three-objective optimisation approach. This series of studies is very important for our research because these are the only non-single-objective optimisation studies on game maps that we know of. Generally, PCG methods that are used to generate game levels or maps, focus on one objective or derive one objective function from multiple objectives via aggregation functions as in studies [7] and [1].

Another study we like to address is Aschlock et al.'s "Search-Based Procedural Generation of Maze-Like Levels" [1]. In this paper, the authors explain representation design and fitness function design in detail. This study is also very important for us since the experiments they have successfully done with direct representation is on a 30 × 30 grid in size. Since we only work on map sketches, we do not need to work with such large solution candidates. Thus, the fact that they were able to work with such large grids, the sizes of which are even more than we need, led our research towards using a direct representation instead of indirect representations or decoders.

Another study we would like to discuss in this category is Liapis et al.'s "Generating Map Sketches for Strategy Games" [7]. This work may be the most influential of all for us, since we adapted its representation and six objective functions for our research. In this work, the authors propose their objective functions which are inspired from game design patterns [2] and previous work on Starcraft maps [9]. The representation approach, metrics and the objective functions they proposed are explained in greater detail in the following Subsect. 2.2.

2.2 Map Sketches for Real Time Strategy Games

Representation: We used direct representation instead of the indirect representation scheme which is used in studies done by Togelius et al. [9,11,12]. Although direct representation increases the length of the chromosome, and hence the size of the search space, since we already adopted the sketch technique (which is also used to decrease chromosome length [7]), this won't pose a problem. In the indirect representation used in [9,11,12], the size of the chromosome is fixed. Because of this representation scheme, the number of resources is fixed. The direct representation allows us to optimise the number of resources and the impassable tiles as well as the structure of the map. The simplicity of the direct representation also makes the addition of new objectives very easy, which paves the way to further extend the study.

In the representation we used, each genome is an array of integers, where each integer corresponds to a tile's type, which can be passable, impassable, base, or resource. The genotype represents the phenotype, i.e. the layout of the map sketch directly, thus we refer to this representation as a direct representation.

Metrics: The *safety metric* is used to evaluate the safety of a tile t with respect to base i when compared to other bases. The calculation of this metric is done according to Eq. 1.

$$S_{t,i} = \underset{\substack{1 \le j \le N_B \\ j \neq i}}{min} \left\{ max \left\{ 0, \frac{d_{t,j} - d_{t,i}}{d_{t,j} + d_{t,i}} \right\} \right\} \tag{1}$$

where $d_{t,i}$ is the shortest path from tile t to tile i and N_B is the number of bases. While this shortest path can be calculated via various ways, we used a simple A^* algorithm.

The *exploration metric* is used to calculate the effort needed to discover all other bases from base i. $E_{j \to i}$ is a function that simulates user behaviour by using a four-way flood-fill algorithm and returns explored tiles while searching for i starting from j. This metric gives higher scores with longer distances and with more open areas between bases. The calculation of this metric is done according to Eq. 2.

$$E_i = \frac{1}{N_B - 1} \sum_{\substack{j=1 \\ j \neq i}}^{N_B} \frac{E_{j \to i}}{w_m h_m - N_I} \tag{2}$$

where w_m, h_m and N_I are width, height and number of inpassable tiles of the map respectively.

The *map coverage of safety* is a metric function that is used to calculate how many tiles are safe with respect to base i. If a tile t's safety value with respect to tile i ($S_{t,i}$) is more than the constant C_S ($C_S = 0.35$ is used throughout this paper since reported results were good with this value in [7]), tile t is declared as *safe*. The calculation of this metric is done according to Eq. 3.

$$A_i = \sum_{t=0}^{N_T} SB_{t,i} \text{ where } SB_{k,j} = \begin{cases} 1 \text{ if } S_{k,j} > C_S \\ 0 \quad \text{otherwise} \end{cases} \tag{3}$$

Objective Functions: The *resource safety function* (f_{res} in Eq. 4) calculates the safety of all resources on the map by using the safety metric (Eq. 1). Higher f_{res} values mean that resources are individually closer to one base than to the others, and are not contested in general.

$$f_{res} = \frac{1}{N_R} \sum_{i=1}^{N_R} \underset{1 \le j \le N_B}{max} \{S_{j,i}\} \tag{4}$$

where N_R is number of resources.

$$f_{saf} = \frac{1}{w_m h_m - N_I} \sum_{i=1}^{N_B} A_i \tag{5}$$

$$f_{exp} = \frac{1}{N_B} \sum_{i=1}^{N_B} E_i \tag{6}$$

The *base safety function* (f_{saf} in Eq. 5) is used to calculate the rate of safe tiles around every base. High scores on f_{saf} means that many tiles around the bases are safe, thus players have wider safety areas.

The *exploration function* (f_{exp} in Eq. 6) measures the hardness of finding other bases for each base. It uses the exploration metric (Eq. 2) which simulates this behaviour of base i.

Balance Functions: Each objective function in this sub-section corresponds to the fairness of the objective functions explained above. These functions are used to create fair and balanced maps for all players. There are three balance objective functions proposed by Liapis et al.'s [7] which are: *resource balance* (b_{res}), *base safety balance* (b_{saf}) and *exploration balance* (b_{exp}) calculated as given in Eqs. 7, 8 and 9 respectively.

$$b_{res} = 1 - \frac{1}{N_R N_B (N_B - 1)} \sum_{k=1}^{N_R} \sum_{i=1}^{N_B} \sum_{\substack{j=1 \\ j \neq i}}^{N_B} |S_{t_k,i} - S_{t_k,j}| \tag{7}$$

$$b_{saf} = 1 - \frac{1}{N_B (N_B - 1))} \sum_{i=1}^{N_B} \sum_{\substack{j=1 \\ j \neq i}}^{N_B} \frac{|A_i - A_j|}{max\,\{A_i, A_j\}} \tag{8}$$

$$b_{exp} = 1 - \frac{1}{N_B (N_B - 1))} \sum_{i=1}^{N_B} \sum_{\substack{j=1 \\ j \neq i}}^{N_B} \frac{|E_i - E_j|}{max\,\{E_i, E_j\}} \tag{9}$$

3 Proposed Approach

Our main aim is to extract a Pareto optimal set of strategy map sketches optimized using the objectives explained in Sect. 2.2. We use a state of the art MOEA approach, namely NSGAIII [3] to be able to achieve our goal, due to the fact that the results reported by Seada in [8] are successful for 3 to 10 objectives.

Constraint handling is not discussed in a separate section due to the simplicity of our approach. In our approach, we basically kill (setting objective values of the solution to zero) the solutions representing sketches with unreachable bases or resources since they occur relatively rare.

3.1 Objective Functions

While designing our objective functions, our aim was to obtain a Pareto optimal solution set where each solution corresponds to a map sketch that is strong on at least one design aspect and fair/balanced among all bases. We wanted to extract a three-objective Pareto-front while fully optimizing all three fairness-objectives. To be able to achieve that, first we derived an *overall balance objective* b_o by way of an aggregation function ($b_o = (b_{res} + b_{saf} + b_{exp})/3$).

Then, we derived the three objectives o_{res}, o_{saf} and o_{exp} as given in Eqs. 10, 11 and 12 respectively.

$$o_{res} = f_{res} + b_o \tag{10}$$

$$o_{saf} = f_{saf} + b_o \tag{11}$$

$$o_{exp} = f_{exp} + b_o \tag{12}$$

These three objective functions are used in the evaluation part of NSGAIII. By doing this, we tried to guarantee that individuals with high b_o values would be more likely to be selected by the selection mechanism, and thus, the population's average b_o value would be increased throughout the run. At the end of a run, the b_o value is subtracted from the used objective functions' values to retrieve their actual values which are f_{res}, f_{saf} and f_{exp}.

3.2 Crossover

Because of the nature of the problem, genes are interrelated with adjacent genes. Because of this, we proposed a new crossover technique called the *one line matrix crossover* which is the equivalent of *one point crossover* used on an array representation.

One line matrix crossover is very easy to implement and is a robust operator which works as follows: First a crossover direction (vertical or horizontal) is selected with equal probability of 0.5. Then a crossover line l_i is selected where $1 \leq i \leq w - 1$ and w is the width or height. Finally, two parent matrices are divided from the chosen line and the parts after the line are exchanged between the parents to create two children. The way this crossover operator works on two sample parent matrices can be seen in Fig. 1.

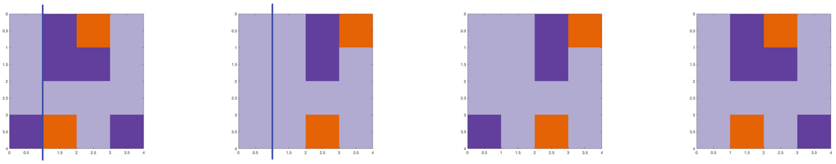


Fig. 1. One Line Matrix Crossover example on a 4 * 4 map sketch. Crossover occurs at vertical line 1 which is marked on the parent matrices. From left to right: parent 1 and 2, children 1 and 2

This crossover technique occasionally produces infeasible solutions that have less than the required number of bases. To fix this problem a repair function is applied right after the crossover operator. In order to satisfy the constraint enforcing a required number of bases in all solution candidates, this repair function simply transfers one base from the *greedy* child, which has more bases than required, to the *faulty* child, which has less bases than required.

3.3 Mutation

As for mutation, we proposed to use two mutation operators which are applied with equal probability to genes selected for mutation based on a mutation probability p_m. The proposed mutation operators are the *weighted random resetting* (WRR) mutation operator and the *gaussian swap mutation* (GSM) operator. But, to keep the feasibility of the genomes by preserving the required number of bases, only GSM is applied to base type tiles.

WRR is applied as follows: before the run, the targeted (required) base count B_N, maximum resource N_{Rmax}, and minimum resource counts N_{Rmin} are determined. p_r ($p_r = (N_{Rmax} + N_{Rmin})/(2 * w * h)$) and p_b ($p_b = N_B/(w * h)$) are calculated according to the expected rate of corresponding number of tiles. The selected gene is flipped to base or to resource with the calculated probabilities. In cases neither of the probabilities are met, the gene is flipped to either a passable or an impassable tile with equal probability.

GSM is the classic swap mutation but the probability of swapping one gene with another gene comes from the Gaussian distribution. With this approach we wanted to make swap mutation's effect less drastic, since swapping one gene with another gene could affect the objective functions largely and cause some high potential individuals to get lost in the selection stage.

4 Experiments

4.1 Experimental Design

All the reported experiments done in this study are listed in Table 1. The first column of the table shows the labels of the tests which will be used in the rest of this section. p_m is the mutation probability where L is the length of a chromosome. Crossover and Mutation columns show the operator(s) used in the corresponding test case. The crossover operator can either be the *one line matrix crossover (OLMC)* proposed in this study or the classical *one point crossover (OPC)* which is already widely used in the area. Mutation operators are the *gaussian swap mutation (GSM)*, classical *uniform swap mutation (USM)*, *weighted random resetting (WRR)* and *uniform random resetting (URR)*. These operators are always used in pairs with equal probability, except for the base tiles which are always mutated using a swap mutation (GSM or USM)).

Parameters not listed in Table 1 are fixed and listed as follows: Maximum number of iterations performed in each run is 100, the size of the generated

Table 1. Test cases and mean values of the comparison metrics averaged over 100 runs.

	p_m	Crossover	Mutation	hv	f_{res}	f_{saf}	f_{exp}	f_o
T1	$1/L$	OPC	GSM/WRR	0.6583	0.7889	0.8935	0.9714	0.8228
T2	$1/L$	OLMC	GSM/WRR	0.6793	0.8024	0.9002	0.9758	0.8359
T3	$0.5/L$	OLMC	GSM/WRR	0.6645	0.7955	0.8940	0.9734	0.8279
T4	$2/L$	OLMC	GSM/WRR	0.6971	0.8150	0.9046	0.9784	0.8421
T5	$3/L$	OLMC	GSM/WRR	0.6996	0.8167	0.9064	0.9796	0.8414
T6	$3/L$	OLMC	USM/WRR	0.7042	0.8156	0.9126	0.9796	0.8421
T7	$3/L$	OLMC	GSM/URR	0.6745	0.7916	0.9091	0.9770	0.8283
T8	$3/L$	OLMC	USM/URR	0.6645	0.7845	0.9052	0.9772	0.8230

map sketch is 8 by 8, the targeted number of bases is 2, number of resources are allowed to vary between 6 and 10, the crossover probability is 1.0, the population size calculated within the NSGAIII algorithm is 91 and the total number of runs performed for each test is 100. We compared the results of each test case based on five different metrics which are: Hyper volume indicator (hv), best f_{res}, f_{saf}, f_{exp} and best averaged solution (f_o) of the resulting Pareto set. We used Simon Wessing's Python implementation[3] for the *hyper volume indicator* calculation which is suggested by Fonseca et al. [5]. All statistical analysis tests discussed in the results section are computed using the one way ANOVA test.

Table 2. ANOVA significance test results.

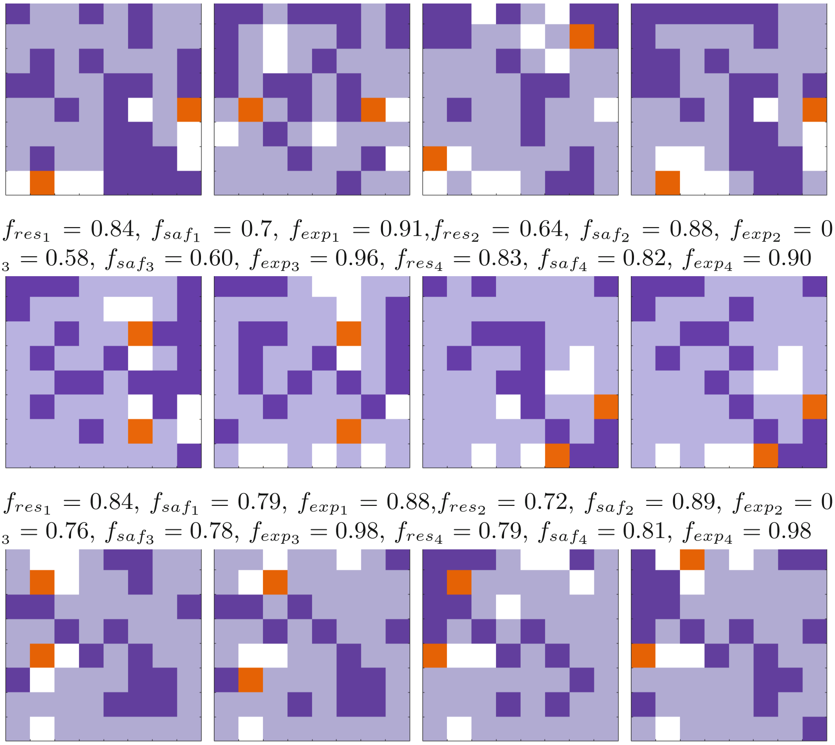
	hv	f_{res}	f_{saf}	f_{exp}	f_o
$T2, T1$	$>$	$>$	$+$	$>$	$>$
$T3, T2$	$<$	$-$	$-$	$-$	$<$
$T3, T4$	$<$	$<$	$<$	$<$	$<$
$T3, T5$	$<$	$<$	$<$	$<$	$<$
$T4, T2$	$>$	$>$	$+$	$>$	$>$
$T4, T5$	$-$	$-$	$-$	$-$	$+$
$T5, T2$	$>$	$>$	$>$	$>$	$+$
$T5, T6$	$-$	$+$	$<$	$=$	$=$
$T5, T7$	$>$	$>$	$-$	$>$	$>$
$T7, T8$	$>$	$+$	$>$	$-$	$>$

Symbols always define how good the first test case is compared to the second one. "$<$", "$>$", "$-$", "$+$" and "$=$" mean *significantly worse*, *significantly better*, *worse*, *better* and *almost equal* respectively. In cases where there is no statistically significant difference, we base the comparison on the actual value of the used metric.

4.2 Results

All the ANOVA comparison results are shown in Table 2 and the mean values of the corresponding comparison metrics averaged over 100 runs are listed in Table 1. It is important to state that all comparisons are done by changing one parameter at a time, thus comparisons are fair but parameters' interactions with each other are unknown. For example, in the first row, $T2$ and $T1$ are only different from each other on the crossover operator but it doesn't mean that all

[3] https://ls11-www.cs.uni-dortmund.de/rudolph/hypervolume/start.



(a) $f_{res_1} = 0.84$, $f_{saf_1} = 0.7$, $f_{exp_1} = 0.91$,$f_{res_2} = 0.64$, $f_{saf_2} = 0.88$, $f_{exp_2} = 0.70$, $f_{res_3} = 0.58$, $f_{saf_3} = 0.60$, $f_{exp_3} = 0.96$, $f_{res_4} = 0.83$, $f_{saf_4} = 0.82$, $f_{exp_4} = 0.90$

(b) $f_{res_1} = 0.84$, $f_{saf_1} = 0.79$, $f_{exp_1} = 0.88$,$f_{res_2} = 0.72$, $f_{saf_2} = 0.89$, $f_{exp_2} = 0.81$, $f_{res_3} = 0.76$, $f_{saf_3} = 0.78$, $f_{exp_3} = 0.98$, $f_{res_4} = 0.79$, $f_{saf_4} = 0.81$, $f_{exp_4} = 0.98$

(c) $f_{res_1} = 0.86$, $f_{saf_1} = 0.79$, $f_{exp_1} = 0.91$,$f_{res_2} = 0.82$, $f_{saf_2} = 0.88$, $f_{exp_2} = 0.87$, $f_{res_3} = 0.70$, $f_{saf_3} = 0.82$, $f_{exp_3} = 0.98$, $f_{res_4} = 0.83$, $f_{saf_4} = 0.85$, $f_{exp_4} = 0.97$

Fig. 2. Example outputs. From left to right best f_{res}, best f_{saf}, best f_{exp} and best mean of different Pareto optimal sets obtained from three different runs. (*orange: base, white: resource, purple: impassable, lilac: passable*)

comparisons between *OLMC* and *OPC* should result in the same way. Also the *balance function* values are not listed in the table because the reported results have mean b_o value of 0.99 (with standard deviation of 0.01) and as a result all the solutions are fair and balanced.

Comparisons can be divided into three different groups as comparisons of *crossover operators*, *mutation probabilities*, and *mutation operators*. The three groups are separated by a horizontal line in Table 2. We also ran tournament size tests for $t_s = 2, 3, 5$ and 7 but there was no significance, thus it is not included to this study. A visualisation of example solutions on the Pareto-fronts obtained from three different runs are shown in Fig. 2.

OLMC vs. OPC: The comparison between $T2$ and $T1$ shows that the proposed crossover operator *one line matrix crossover* is almost always significantly better than the widely used *one point crossover*. We believe the reason behind this

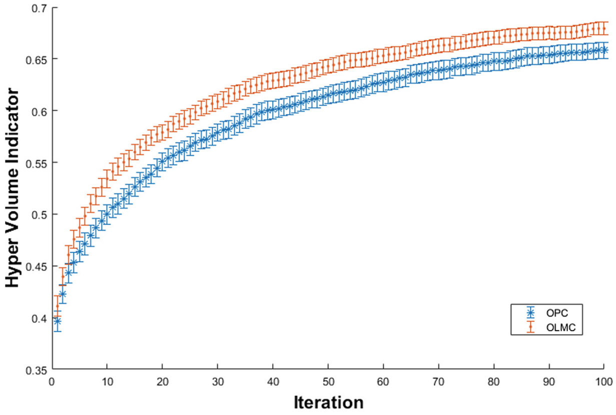


Fig. 3. Behaviour of *one line matrix crossover* and *one point crossover*. Means and standard errors of hyper volume indicator distributions.

significance comes from the nature of the *direct matrix representation*. All of the operators including objective functions are working on this representation and correlation between objective functions and tile types is also directly correlated with neighbour tiles. Because of that, it is important to transfer data from parents to children as groups.

Behaviour of the compared crossover operators based on the hyper volume indicator, can be seen in Fig. 3 over 100 iterations. As the figure shows, $OLMC$ is the better choice. This shows that the *one line matrix crossover* is a good and robust choice that works well with the *direct representation*, converging to better solutions very quickly while not getting stuck on local optima.

Effect of Mutation Probability p_m: In this comparison we tested values of p_m $0.5/L$, $1/L$, $2/L$ and $3/L$. L is the length of a chromosome (L) which is 64 ($8*8$) for all tests. High mutation probability $3/L$ is almost better than all of the other values except on f_o. Most significant difference occurs on f_{exp}. To increase the exploration function value of a solution, an algorithm should both organise and increase the number of impassable tiles. So a high mutation probability works good on f_{exp}. Another thing we should address is that $p_{r,s}$ (probability of applying random resetting over swap mutation) is equal to 0.5; and that means almost half of the mutation works as an organiser due to the GSM.

Since GSM acts like a local optimiser in this approach, we recommend half of the chosen $3/L$ which is $1.5/L$ as a starting point if the algorithm is using conventional mutation operators (for example *random resetting*).

Effect of Different Mutation Operators: Our test results show that (see Table 2, test cases $T5$, $T6$, $T7$ and $T8$) using WRR instead of URR makes a significant difference. In almost all objectives and hv WRR works better than URR. Although selecting GSM over USM makes a significant difference (see

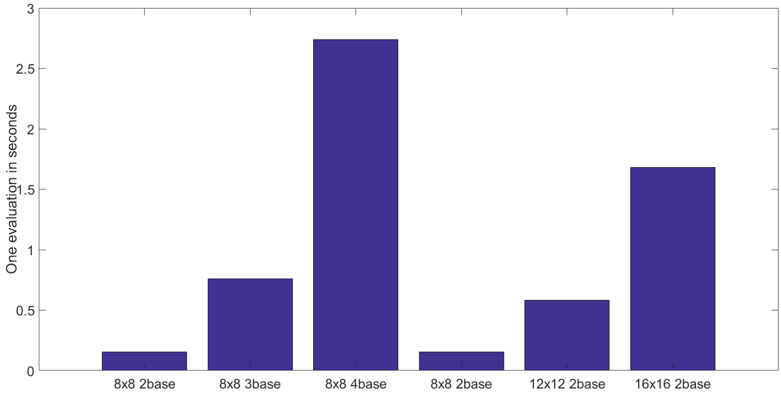
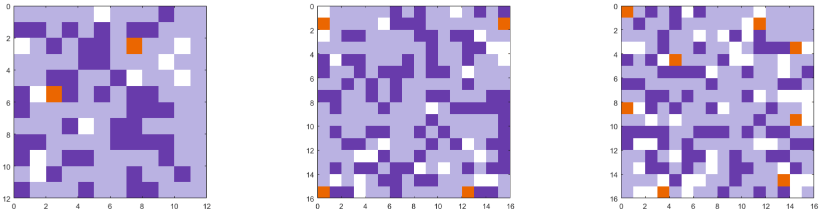


Fig. 4. Runtime comparison between different map sketch sizes and number of bases. Runtimes are based on the time it takes for the program to complete only one iteration of a run.



(a) a 12*12 with 2 bases ($f_{res} = 0.76$, $f_{saf} = 0.9$, $f_{exp} = 0.86$, $b_o = 0.98$)
(b) 16*16 with 4 bases ($f_{res} = 0.67$, $f_{saf} = 0.89$, $f_{exp} = 0.88$, $b_o = 0.95$)
(c) 16*16 with 8 bases ($f_{res} = 0.54$, $f_{saf} = 0.89$, $f_{exp} = 0.67$, $b_o = 0.91$)

Fig. 5. Example map sketches generated by the algorithm.

Table 2, row $T7$, $T8$), from the comparison between $T5$ and $T6$ we can say that GSM and WRR doesn't work well together.

We also compared different combinations of mutation operators (GSM with WRR, USM with WRR, GSM with URR and no mutation). Cases with WRR significantly better than others as expected but, interestingly GSM with URR is worse than the case without any mutation. Although it might be an unexpected result, the reason behind it is quite straightforward. URR generates relatively high number of infeasible solutions (due to high probability of resetting a gene to a resource) therefore it results in more *deaths* than any other mutation operators. And that prevents the algorithm to reach good quality solutions.

Scalability: We also performed some experiments to explore the scalability of our approach by using it on problems with different number of bases and different map sketch sizes. From Fig. 4 it is clear that mean evaluation time

(t_e) increases with both map sketch size and number of bases exponentially. Although evaluation time increases drastically, results are decent and show us that Liapis et al.'s objective functions work well even for 16 by 16 map sketches with 8 bases.

Visualisations of example solutions from Pareto-fronts obtained from sample runs with different base counts and map sketch sizes are shown in Fig. 5.

5 Conclusion

In this study, we proposed a many objective optimisation approach based on the NSGAIII algorithm for the map sketch generation problem in strategy games. We derived our objective functions from map sketch generation objectives proposed previously in literature [7]. We also extended the direct representation used in the same study to better represent a matrix, which is the most natural representation for our problem. We then proposed a new crossover operator and a set of new mutation operators which exploit the inherent nature of the used representation and work efficiently, thus increasing the locality of the representation with respect to the chosen objectives.

The results of the study in general show that MOEAs can be used as an alternative to aggregation functions that are commonly used in single-objective algorithms. The experiments we have done also demonstrate that MOEAs are decent methods to use in the field of procedural content generation. The chosen MOEA algorithm NSGAIII efficiently provided very good solutions optimised on different design objectives within a single run.

This approach can be used in several ways. It can be used as a suggestion mechanism to game level designers which was the main idea in Liapis' study [7]. Also it can be actively used online within the strategy games as a map (level) generator. In order to do this, sketches should be converted to actual game maps. Also, due to the slowness of MOEAs the program should be executed as a background process to avoid long loading screens. Another possible usage can be for benchmark test data generation for various purposes, such as testing shortest path algorithms or artificial intelligence bots etc. Our approach is a good choice particularly for this last case because many different maps can be generated in a single run.

As future work, we want to experiment with different, possibly more than three, objective functions to better represent the preferences of game level designers and to test the limits of NSGAIII in game map generation. For this reason, choke points can also be used as part of the evaluations as suggested in [9] and implemented in [11]. Also we would like to work on much bigger maps similar to open world game maps. To summarise, in future studies we want to push the limits of evolutionary computation on procedurally generated game levels/maps further.

References

1. Ashlock, D., Lee, C., McGuinness, C.: Search-based procedural generation of maze-like levels. IEEE Trans. Comput. Intell. AI Games **3**(3), 260–273 (2011)
2. Bjork, S., Holopainen, J.: Patterns in Game Design (Game Development Series). Charles River Media Inc., Rockland (2004)
3. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans. Evol. Comput. **18**(4), 577–601 (2014)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
5. Fonseca, C.M., Paquete, L., Lopez-Ibanez, M.: An improved dimension-sweep algorithm for the hypervolume indicator. In: 2006 IEEE International Conference on Evolutionary Computation, pp. 1157–1163, July 2006
6. Jain, H., Deb, K.: An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: handling constraints and extending to an adaptive approach. IEEE Trans. Evol. Comput. **18**(4), 602–622 (2014)
7. Liapis, A., Yannakakis, G.N., Togelius, J.: Generating map sketches for strategy games. In: Applications of Evolutionary Computation: 16th European Conference, EvoApplications 2013, Vienna, Austria, 3-5 April 2013, pp. 264–273. Springer, Heidelberg (2013)
8. Seada, H., Deb, K.: U-NSGA-III: a unified evolutionary optimization procedure for single, multiple, and many objectives: proof-of-principle results. In: Evolutionary Multi-Criterion Optimization: 8th International Conference, EMO 2015, Guimarães, Portugal, 29 March–1 April 2015, Part II, pp. 34–49. Springer International Publishing, Cham (2015)
9. Togelius, J., Preuss, M., Beume, N., Wessing, S., Hagelbäck, J., Yannakakis, G.N.: Multiobjective exploration of the starcraft map space. In: Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games, pp. 265–272, August 2010
10. Togelius, J., Yannakakis, G.N., Stanley, K.O., Browne, C.: Search-based procedural content generation: a taxonomy and survey. IEEE Trans. Comput. Intell. AI Games **3**(3), 172–186 (2011)
11. Togelius, J., Preuss, M., Beume, N., Wessing, S., Hagelbäck, J., Yannakakis, G.N., Grappiolo, C.: Controllable procedural map generation via multiobjective evolution. Genet. Program. Evolvable Mach. **14**(2), 245–277 (2013). http://dx.doi.org/10.1007/s10710-012-9174-5
12. Togelius, J., Preuss, M., Yannakakis, G.N.: Towards multiobjective procedural map generation. In: Proceedings of the 2010 Workshop on Procedural Content Generation in Games, PCGames 2010, pp. 3:1–3:8. ACM, New York (2010). http://doi.acm.org/10.1145/1814256.1814259

A Reference-Inspired Evolutionary Algorithm with Subregion Decomposition for Many-Objective Optimization

Xiaogang Fu[1], Jianyong Sun[2(✉)], and Qingfu Zhang[3]

[1] School of Electronic Engineering, Shanghai Dianji University,
PuDong, Shanghai 201306, China
fuxg@sdju.edu.cn

[2] School of Mathematics and Statistics, Xi'an Jiaotong University,
Xi'an 710049, China
jy.sun@xjtu.edu.cn

[3] Department of Computer Science, City University of Hong Kong,
Hong Kong, Hong Kong
qingfu.zhang@city.edu.hk

Abstract. In this paper, we propose a reference-inspired multiobjective evolutionary algorithm for many-objective optimisation. The main idea is (1) to summarise information inspired by a set of randomly generated reference points in the objective space to strengthen the selection pressure towards the Pareto front; and (2) to decompose the objective space into subregions for diversity management and offspring recombination. We showed that the mutual relationship between the objective vectors and the reference points provides not only a fine selection pressure, but also a balanced convergence-diversity information. The decomposition of the objective space into subregions is able to preserve the Pareto front's diversity. A restricted stable match strategy is proposed to choose appropriate parent solutions from solution sets constructed at the subregions for high-quality offspring generation. Controlled experiments conducted on a commonly-used benchmark test suite have shown the effectiveness and competitiveness of the proposed algorithm in comparison with several state-of-the-art many-objective evolutionary algorithms.

Keywords: Many-objective optimization · Reference-inspired · Domain decomposition

1 Introduction

Unconstrained multiobjective optimisation problem (MOP) can be stated as follows:

$$\min \quad \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x})) \text{ s.t. } \mathbf{x} \in \Omega \subset \mathbb{R}^n \tag{1}$$

where $\Omega = \prod_{i=1}^{n}[a_i, b_i]$ is the decision (variable) space, $\mathbf{x} = (x_1, \cdots, x_n) \in \Omega$ denotes candidate solution; and $\mathbf{F} : \Omega \rightarrow \mathbb{R}^m$ constitutes m objectives. A solution $\mathbf{F}^1 = (f_1^1, \cdots, f_m^1)$ is said to dominate another solution $\mathbf{F}^2 = (f_1^2, \cdots, f_m^2)$

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_12

(denoted as $\mathbf{F}^1 \preceq \mathbf{F}^2$) iff $f_j^1 \leq f_j^2$ for all $j \in \{1, \cdots, m\}$ and there exists at least one index i such that f_i^1 is strictly less than f_i^2. A solution $\mathbf{F}^*$ is Pareto optimal to (1) if there is no other $\mathbf{F}$ such that $\mathbf{F} \preceq \mathbf{F}^*$. The set of the Pareto optimal solutions is called the Pareto Front (PF), while the set of optimal solutions in Ω is called the Pareto set (PS). We consider $m \geq 3$ which is often named as many-objective optimization problems (MaOPs).

In recent years, interests on developing MOEAs for MaOPs grows fast [8]. Due to the increasing number of objectives, it becomes very challenging for MOEAs to find an approximate PF that well balance convergence and diversity. It has been found that the performance of MOEAs on two- and three-objective optimisation problems deteriorates significantly on MaOPs [14].

The decomposition-based framework provides an efficient way to balance convergence and diversity. It can keep the selection pressure toward the PF while maintain the population diversity through the construction of uniformly distributed weight vectors and through selection based on scalarising aggregation functions. The convergence to the PF can be controlled by minimising the aggregation function at each subproblem; while the diversity can be measured by minimising the distance to the weight vectors. Pareto-dominance and decomposition can be combined together to preserve diversity and promote convergence simultaneously. For examples, Yuan et al. [17] proposed a similar approach in terms of reference-points generation and adaptive normalisation as in [5,9]; while the diversity preservation is realised by a clustering operator and a proposed θ-dominance. Deb and Jain [5] used a set of reference points to partition the search space. In each subspace, Pareto-dominance is applied to rank individual solutions.

It is also worth mentioning MOEAs that are based on a set of predefined reference points in solving MaOPs. A set of reference points provides an external inspiration to measure diversity; and can be used to address the first two challenges. For examples, Wang et al. [15] also proposed to co-evolve a population of candidate objective vectors and a randomly-generated reference set, where the candidate objective vectors and the reference set are evaluated against each other to provide comparability among the objective vectors. The proposed algorithm demonstrated somewhat promising performance for MaOPs. Figueira et al. [6] developed a two-stage parallel multi-reference point approach which connects the generation of reference points and the solving for each reference point in every processor. The two-archive algorithm proposed by Kata and Yao et al. [13] constructs the reference set using historical or current solutions in the convergence archive and diversity archive. TC-SEA developed by Moen et al. [12] uses a similar way to construct the reference set, while the principle of selection criteria is based on the Manhattan distance.

Motivated by the success of the approaches that combine weight and reference points for MaOPs, we propose a reference-inspired multiobjective evolutionary algorithm (RIEA) for MaOPs where the decomposition idea is incorporated.

2 Fitness Assignment

In this section, we present a new fitness assignment method based on a set of *randomly* generated reference points in the original m-dimensional objective space. In the sequel, we use $\mathcal{N}_r$ and $\mathcal{N}_p$ to represent the set of reference points and the population of solutions, respectively.

In the following, we define a function $L : \mathcal{N}_r \rightarrow \mathbb{Z}_+$ to represent how many solutions in the population set $\mathcal{N}_p$ dominate a reference point $r \in \mathcal{N}_r$. $L(r) = 1$ indicates that there is only one solution that dominates r. By this way, we can cluster the reference points according to their L values, which can result in several layers. The layers of the reference points provide us some useful information on the dominance relationship and convergence-diversity. Further, we define a function $D : \mathcal{N}_p \rightarrow \{0, 1\}$ to characterise the non-dominance of a solution in $\mathcal{N}_p$. If a solution s is non-dominated, we assign its D value as one, otherwise zero.

Information about a solution's convergence can also be provided according to the number of reference points dominated by this solution. It is intuitive that more reference points dominated by this solution, the closer it approximates to the PF. Further, information regarding a solution's diversity can also be implied by the reference points dominated by it. Basically, if a reference point's L value is large, it gives us hardly any information.

$$V(s) = \frac{1}{|\mathcal{N}_r|} \sum_{r \in \mathcal{N}_r} \frac{M_r}{T_r} \tag{2}$$

where M_r is the number of reference points dominated by s, and T_r is the number of solutions in $\mathcal{N}_p$ that dominate r; $|\mathcal{N}_r|$ is the total number of reference points in $\mathcal{N}_r$. Note that if we define $\mathcal{L}_i = \{r : L(r) = i\}$ as the i-th layer, then we can rewrite Eq. (2) as follows:

$$V(s) = \frac{1}{|\mathcal{N}_r|} \sum_{r \in \mathcal{L}_i} \frac{M_r}{i}. \tag{3}$$

We propose to aggregate the non-domination and convergence-diversity information for ranking solutions in a population. That is, for each $s \in \mathcal{N}_p$, we define

$$R(s) = D(s) + V(s) \tag{4}$$

The immediate advantage of the proposed fitness assignment scheme is that the ranking of solutions can be automatically adapted to different evolution stages. At early stage, the D value dominates Eq. (4) since the non-dominated solutions have higher chances to survive than the dominated solutions. At later stage, solutions are almost non-dominated to each other (this means that all solutions will have $D = 1$), which is especially the case in MaOPs as stated in the first challenge. Therefore, the V value will dominate the ranking based on Eq. (4). In this case, solutions will be differentiated based on their respect convergence and diversity information.

3 The Algorithm

The framework of the proposed reference-inspired multiobjective evolutionary algorithm (in short, RIEA) is summarised in Algorithm 1. RIEA maintains a set of N individuals $\mathcal{S} = \{\mathbf{x}^1, \cdots, \mathbf{x}^N\}$, and their corresponding objective vectors $\mathcal{P} = \{\mathbf{F}^1, \cdots, \mathbf{F}^N\}$. An initialisation procedure first generates N initial solutions, K reference directions (grouped in set $\mathcal{V}$), M reference points (grouped in set $\mathcal{R}$) (line 2). The neighbourhood index set $\mathcal{B}$ for each solutions is identified in the initialisation procedure and remains fixed during the evolution procedure. Within the main while-loop, a parameter δ is used to decide where to select parent solutions (line 6–10) for offspring generation. An offspring is then generated by the variation of these parent solutions (line 11) taking two elite sets $\overline{\mathcal{C}}$ and $\overline{\mathcal{D}}$ into consideration (in the first generation, no elite sets are required). New population and elite sets are updated (line 12), by employing an elite-preserving mechanism.

Algorithm 1. The Framework of RIEA

```
Require: a population size $N$
Ensure: population $\mathcal{P}$
 1: $\mathcal{C} \leftarrow \emptyset, \mathcal{D} \leftarrow \emptyset, t \leftarrow 0$;
 2: $[\mathcal{P}, \mathcal{V}, \mathcal{R}, \mathcal{B}] \leftarrow$ INITIALIZATION();
 3: while termination not satisfied do
 4:     $\mathcal{P}' \leftarrow \emptyset$;
 5:     for $i \leftarrow 1$ to $N$ do
 6:         if $rand() > \delta$ or $t = 0$ then
 7:             $\overline{\mathcal{B}} \leftarrow \{1, \cdots, K\}, \overline{\mathcal{C}} \leftarrow \emptyset; \overline{\mathcal{D}} \leftarrow \emptyset$;
 8:         else
 9:             $\overline{\mathcal{B}} \leftarrow \mathcal{B}_i; \overline{\mathcal{C}} \leftarrow \mathcal{C}_i; \overline{\mathcal{D}} \leftarrow \mathcal{D}_i$;
10:         end if
11:         $\mathbf{p} \leftarrow$ REPRODUCTION($\mathbf{x}^i, \overline{\mathcal{B}}, \overline{\mathcal{C}}, \overline{\mathcal{D}}$);
12:         $\mathcal{P}' \leftarrow \mathcal{P}' \bigcup \mathbf{F}(\mathbf{p})$;
13:     end for
14:     $\overline{\mathcal{P}} \leftarrow \mathcal{P} \bigcup \mathcal{P}'$; $[\mathcal{P}, \{\mathcal{C}_i, \mathcal{D}_i\}_{i=1}^{K}] \leftarrow$ UPDATE_POPULATION($\overline{\mathcal{P}}, \mathcal{R}, \mathcal{V}$);
15:     $t \leftarrow t + 1$
16: end while
17: return $\mathcal{P}$
```

3.1 Initialisation and Decomposition to Subregions

The initialisation procedure is presented in Algorithm 2. Individuals are to be generated within the search space by randomly sampling from Ω (line 1). The individuals are evaluated and their objective values are stored in $\mathcal{P}$ (line 2). Next, we generate a set of randomly sampled points within $[0, 1]^m$ (line 3). The generation of the reference vectors follows the approach developed in [11]. In this approach, reference vectors $\mathbf{v}^i = (v_1^i, \cdots, v_m^i), 1 \le i \le K$ are generated on a unit hypersphere: $v_k^i \in \left\{\frac{0}{H}, \frac{1}{H}, \cdots, \frac{H}{H}\right\}$, s.t. $\sum\limits_{i=1}^{m} v_k^i = 1$ where H is a positive integer. Note that for different H and m, the number of reference vectors K generated by the method is $K = \binom{H+m-1}{m-1}$.

Algorithm 2. The initialisation procedure (Initialization)

```
Ensure: $\mathcal{P}, \mathcal{V}, \mathcal{R}, \mathcal{B}$
1: Generate an initial population of individuals, where $\mathbf{x}_j^i \in [a_j, b_j], 1 \le j \le N, 1 \le i \le n$ randomly.
2: Evaluate these individuals: $\mathbf{F}^i = \mathbf{F}(\mathbf{x}^i), 1 \le i \le N$.
3: Randomly generate a set of $M(= m \times 100)$ reference points, denoted by $\mathcal{R}$, in $[0,1]^m$.
4: Randomly generate a set of $K$ reference vectors, denoted by $\mathcal{V} = \{\mathbf{v}^1, \cdots, \mathbf{v}^K\}$, in a $(m-1)$-dimensional unit simplex.
5: For each $i \in \{1, \cdots, N\}$, set $\mathcal{B}_i = \{i_1, \cdots, i_T\}$ where $\mathbf{v}^{i_1}, \cdots, \mathbf{v}^{i_T}$ are the $T$ closest reference vectors to $\mathbf{v}^i$.
6: return $\mathcal{P}, \mathcal{V}, \mathcal{R}, \mathcal{B}$.
```

After generation of the reference vectors, we then identify the neighbourhoods of each reference vector (line 5). The reference vectors will be used to divide solutions into subregions in the objective space. That is, for each reference vector $\mathbf{v}^i$, the subregion, denoted as Λ_i, can be defined as

$$\Lambda_i = \left\{\mathbf{F} \middle| \langle \mathbf{F}, \mathbf{v}^i \rangle \le \langle \mathbf{F}, \mathbf{v}^j \rangle \text{ for } 1 \le j \le K, j \ne i \right\} \tag{5}$$

where $\langle \mathbf{F}, \mathbf{v}^i \rangle$ is the acute angle between $\mathbf{F}$ and the reference vector $\mathbf{v}^i$. That is, an objective vector $\mathbf{F}$ belongs to subregion Λ_i if it has the least acute angle value. We use Ω_i to denote the corresponding solutions in the search space, that is

$$\Omega_i = \left\{\mathbf{x} \middle| \mathbf{F}(\mathbf{x}) \in \Lambda_i \right\}. \tag{6}$$

The neighbourhood index set $\mathcal{B}_i$ of $\mathbf{v}^i \in \Lambda_i$ is also considered as the index set of Ω_i. The definition of subregion is the same as that in MOEA/DD [9] and MOEA/D-M2M [11]. However, in MOEA/DD, the subregions are used to facilitate local density estimation; while in MOEA/D-M2M, they are used to specify sub-populations for multiobjective subproblems. In this paper, the subregions are used to choosing parent solutions for offspring generation.

3.2 Reproduction Procedure

The differential evolution (DE) and polynomial mutation [3] are used to generate offspring. To generate an offspring, the mutation operator of DE variates an individual $\mathbf{x}^i$ using two parent individuals $\mathbf{x}^{r_1}$ and $\mathbf{x}^{r_2}$. A mating control parameter $\delta \in [0, 1]$ is used to decide where to choose the parent individuals. Specifically, with δ, parent individuals are selected from the neighbourhood of $\mathbf{x}^i$ (the neighbourhood indices are predefined in $\mathcal{B}_i$); otherwise the whole population is considered as the neighbourhood. This means to balance exploration and exploitation.

Moreover, to address the second challenge (i.e. the inefficiency of reproduction operators), we proposed a selection mechanism similar to the restricted mating selection mechanism proposed in [10] for choosing parent solutions from $\mathbf{x}^i$'s neighbourhood. That is, we first randomly select two subregions from the neighbourhood index set. Then the parent individuals are chosen from the convergence elite set and diversity elite set that are defined in the two selected subregions, respectively. How to construct the convergence elite set $\mathcal{C}_i$ and diversity elite set $\mathcal{D}_i$ for each subregion Λ_i will be presented later.

In our implementation, two combinations of DE parameters are applied. This is to address different search purpose (exploration if the whole population is considered as neighbourhood and exploitation otherwise).

3.3 Environmental Selection

In our environmental selection procedure, we need to consider two main issues. The first is on how to rank solutions according to their R values (cf. Eq. (2)). The second is on how to construct the convergence elite set and diversity elite set. To implement environmental selection, we first normalise the solutions in $\overline{\mathcal{P}}$ and the set of reference points in $\mathcal{R}$ to scale up the fitness assignment. Each $\mathbf{F}^i \in \overline{\mathcal{P}}$ is normalised as $\overline{\mathbf{F}}^i = (\bar{f}_1^i, \cdots, \bar{f}_m^i)$ with each element j computed as follows:

$$\bar{f}_j^i = \frac{f_j^i - f_j^{\min}}{f_j^{\max} - f_j^{\min}}, \forall j \in \{1, \cdots, m\} \tag{7}$$

where $f_j^{\min}$ and $f_j^{\max}$ are the minimum and maximum values of $\overline{\mathcal{P}}$ at the j-th objective. They will be used to normalise $\mathcal{R}$ as well.

Secondly, we assign the combined population of solutions to subregions (see Algorithm 3). During the assignment, we make sure that there are at least L solutions in each subregion. New population is then selected from the subregions by selecting N solutions with the largest R (Eq. (4)) values from these subregions (Algorithm 4). Finally, for each subregion Λ_i, we construct two elite sets $\mathcal{C}_i$ and $\mathcal{D}_i$ which will be used to generate offspring concerning convergence and diversity, respectively (Algorithm 5). In each subregion, we choose a solution from the combined population with the smallest acute angle to the reference vector in this subregion as the diversity elite set. Meanwhile, we select all non-dominated solutions in each subregion as the convergence elite set. If there is

Algorithm 3. Partition

```
Require: $\mathcal{P}, \mathcal{R}, \mathcal{V}$
Ensure: $\mathcal{P}$
Set $\Lambda_i \leftarrow \emptyset, \forall i \in \{1, \cdots, K\}$;
for $j \leftarrow 1$ to $N$ do
    $k \leftarrow \arg\min_i \langle \overline{\mathbf{F}}^j, \mathbf{v}^i \rangle$; $\Lambda_k \leftarrow \Lambda_k \bigcup \mathbf{F}^k$;
end for
for $j \leftarrow 1$ to $K$ do
    if the number of solutions in $\Lambda_j$ is less than $L$ then
        Set $U \leftarrow \emptyset$, $S \leftarrow \emptyset$;
        for $i \leftarrow 1$ to $T$ do
            $k \leftarrow \mathcal{B}_j(i)$;
            Compute the $R$ values for $s \in \Lambda_k$, store these values in $U_k$;
            $U \leftarrow U \bigcup U_k$ and $S \leftarrow S \bigcup \Lambda_k$;
        end for
        Select an index set $I$ proportionally from $U$ subject to $|I| = L - |\Lambda_i|$;
        $\Lambda_i \leftarrow \Lambda_i \bigcup S_I$;
    end if
end for
return $\Lambda$
```

no non-dominated solution, solutions that are closest to the reference vector in that subregion will be selected as the convergence elite. It is worth noting that a solution's D value is computed through the comparison within the entire population and its V value is computed with respect to the reference points within its subregion.

Algorithm 4. Population_Selection

```
Require: $\Lambda$
Ensure: $\mathcal{P}$
Set $\mathcal{P} \leftarrow \emptyset$;
for $i \leftarrow 1$ to $N$ do
    Randomly generate an integer $j \in \{1, \cdots, K\}$;
    $\bar{\mathbf{s}} \leftarrow \arg\max_{\mathbf{s} \in \Lambda_j} R_{\mathbf{s}}$;
    $\mathcal{P} \leftarrow \mathcal{P} \bigcup \mathbf{p}$ where $\mathbf{p} \in \Omega_j$ and $\mathbf{F}(\mathbf{p}) = \bar{\mathbf{s}}$;
    $\Lambda_j \leftarrow \Lambda_j \setminus \bar{\mathbf{s}}$;
end for
return $\mathcal{P}$
```

Algorithm 5 presents the construction of the convergence and diversity set at the subregions. To construct the convergence elite set for a subregion Λ_i, we first find all the non-dominated solution in that region (line 2). If there is no non-dominated solution, we search non-dominated solutions in its neighbourhood region and put these solutions in its convergence elite set (line 3 to 7). To construct the diversity elite set, we locate the solution that is closest to the reference vector $\mathbf{v}^i$ (line 9); and take this solution as the diversity elite set (line 10).

Algorithm 5. Elite_Set_Construction

```
Require: $\mathcal{P}, \mathcal{V}$
Ensure: $\mathcal{C}, \mathcal{D}$
 1: for $i \leftarrow 1$ to $K$ do
 2:     Find the non-dominated solutions in $\Lambda_i \rightarrow U$;
 3:     if $|U| = 0$ then
 4:         Find out the nearest non-dominated solutions in the neighbourhood of
            $\Lambda_i$, i.e. $\bigcup_{j \in \mathcal{B}_i} \Lambda_j$;
 5:         Add the individuals associated with these solutions to $\mathcal{C}_i$;
 6:     else
 7:         $\mathcal{C}_i \leftarrow U$;
 8:     end if
 9:     $\bar{\mathbf{s}} \leftarrow \arg\min_{\mathbf{s} \in \Lambda_i} \langle \mathbf{s}, \mathbf{v}^i \rangle$;
10:     $\mathcal{D}_i \leftarrow \{\mathbf{p}\}$ where $\mathbf{p} \in \Omega_i$ and $\mathbf{F}(\mathbf{p}) = \bar{\mathbf{s}}$;
11: end for
12: return $\{\mathcal{C}_i, \mathcal{D}_i\}_{i=1}^{K}$
```

4 Experimental Study

We carried out controlled experiments to test the performance of IREA. We tested our algorithm on DTLZ1-DTLZ7 [4] with 3, 5, 8, 10, and 15 objectives. As suggested in [4], the number of decision variables is set to $m + r - 1$, where m is the number of objectives, $r = 5$ for DTLZ1 and $r = 10$ for DTLZ2, DTLZ3 and DTLZ4.

To assess the performance, we choose two widely used performance metrics: the inverse generational distance (IGD) [2] and Hypervolume (HV) [19]. They are the metric representatives to measure the convergence and diversity of the obtained solutions [7].

We choose four state-of-the-art multiobjective evolutionary algorithms, including PICEA-g [15], GrEA [16], HypE [1] and MOEA/D [18] for comparison. The parameters of the compared algorithms are set as reported by their authors, respectively. All the codes were written in Matlab. We obtained the Matlab codes of the compared algorithms from the authors' websites. For a fair comparison, each algorithm was executed independently 20 times on each test instance in our machine. To compute the performance metrics, the same Pareto optimal points for IGD and the reference points for HV were used in the compared algorithms.

All the compared algorithms terminate at 30,000 function evaluations. The specific parameter settings of our proposed RIEA are summarised as follows.

- The number of reference points: $100 \times m$.
- Population size: $N = 300$ for all test instances except $m = 15$ with $N = 900$.
- The number of reference vectors K is obtained by deciding H according to Eq. (3.1) such that it is close to $10m + 1$.
- Settings for reproduction operators: the mutation probability $p_m = 1/n$ and its distribution index is set to be 20;
- The DE parameters: $F_g = 0.5, CR_g = 0.5; F_l = 0.2, CR_l = 0.8$.
- The neighbourhood size: $T = 5$.
- The probability used to select parent solutions: $\delta = 0.9$.
- The minimum number of individuals in every subregion: $L = 5$.

4.1 Experimental Results

Table 1 shows the obtained results in terms of IGD, respectively. The best, mean and worst metric values are summarised in the tables. Moreover, the ranks of these algorithms on each problem are also presented. In the tables, best results are shaded in grey color.

From Table 1, we see clearly that RIEA outperforms GrEA and HypE on all test instances with all numbers of objectives in terms of median IGD value. Especially, the performance of HypE is the worst on these problems in terms of IGD; while GrEA is the second worst. For DTLZ1 and DTLZ3, RIEA performs better than PICEA-g in terms of the best and median values of the IGD metric for all considered numbers of objectives. For DTLZ2, RIEA obtains better median IGD values than the other three algorithm on 8–10 objectives. PICEA-g performs the best on DTLZ2 with 3- and 5-objectives in terms of IGD, but RIEA obtains the best IGD values on the problem with 3-objectives. On DTLZ4, we see that RIEA performs the best over all the compared algorithm. Moreover, it is observed that along with the increase of the number of objectives, the performance of RIEA becomes better. This clearly shows that the proposed fitness assignment scheme can indeed increase the comparability of the solutions in high-dimensional objective space.

Table 2 shows the comparison results on DTLZ1-DTLZ4 in terms of HV. From the table, it is observed that RIEA performs better along with the increase of the number of objectives. For DTLZ1, MOEA/D performs the best on 3- and 5-objectives, while RIEA performs the best on instances with ≥ 8 objectives. For DTLZ2 to DTLZ4, the best performance is obtained by GrEA and HypE for instances with small numbers (≤ 8) of objective; but RIEA achieves better performance on DTLZ2 with 10 and 15 objective.

Table 1. Statistical results (best/mean/worst) obtained by RIEA, MOEA/D, PICEA-G, GrEA and HypE in 20 independent runs on the DLTZ test suite in terms of the *IGD* metric.

	m	RIEA	MOEA/D	PICEA-G	GrEA	HypE		RIEA	MOEA/D	PICEA-G	GrEA	HypE
DTLZ1	3	**1.658E-03(1)**	2.869E-03(3)	3.529E-03(4)	2.76E-02(2)	1.82E+01(5)	DTLZ3	9.467E-03(2)	**3.277E-03(1)**	1.093E-01(4)	6.77E-02(3)	1.65E+02(5)
		2.567E-03(1)	2.904E-03(2)	2.067E-02(3)	3.34E-02(4)	1.97E+01(5)		**2.450E-02(1)**	4.635E-02(2)	2.335E-01(4)	7.69E-02(3)	1.70E+02(5)
		8.238E-03(2)	**2.942E-03(1)**	6.174E-02(3)	1.35E-01(4)	2.16E+01(5)		**6.114E-02(1)**	1.406E-01(2)	5.502E-01(4)	4.47E-01(3)	1.76E+02(5)
	5	**6.635E-03(1)**	8.596E-03(2)	1.243E-02(3)	7.37E-02(4)	1.80E+01(5)		**1.155E-02(1)**	**1.155E-02(1)**	5.789E-02(3)	5.33E-01(4)	1.83E+02(5)
		7.420E-03(1)	8.953E-03(2)	2.781E-02(3)	3.36E-01(4)	2.14E+01(5)		**3.112E-02(1)**	1.019E-01(2)	1.894E-01(3)	8.30E-01(4)	2.17E+02(5)
		8.741E-02(3)	**9.504E-03(1)**	8.106E-02(2)	4.94E-01(4)	2.36E+01(5)		**9.316E-02(1)**	4.355E-01(3)	3.935E-01(2)	1.12E+00(4)	2.28E+02(5)
	8	**1.932E-02(1)**	2.073E-02(2)	4.009E-02(3)	1.02E-01(4)	1.03E+01(5)		4.910E-02(2)	**4.331E-02(1)**	1.173E-01(3)	7.52E-01(4)	2.20E+02(5)
		2.411E-02(1)	5.742E-02(2)	5.628E-02(3)	1.20E-01(4)	2.27E+01(5)		**7.554E-02(1)**	6.094E-01(3)	2.988E-01(2)	1.02E+00(4)	2.70E+02(5)
		3.214E-02(1)	3.747E-01(3)	1.363E-01(2)	3.85E-01(4)	2.43E+01(5)		**1.270E-01(1)**	4.728E+00(4)	8.273E-01(2)	1.23E+00(3)	2.95E+02(5)
	10	**2.362E-02(1)**	2.498E-02(2)	4.431E-02(3)	1.18E-01(4)	1.43E+01(5)		4.898E-02(2)	**4.862E-02(1)**	1.224E-01(3)	8.66E-01(4)	1.72E+02(5)
		2.880E-02(1)	3.779E-02(2)	6.329E-02(3)	1.59E-01(4)	1.69E+01+(5)		**6.757E-02(1)**	4.128E-01(2)	3.737E-01(3)	1.15E+00(4)	2.89E+02(5)
		3.272E-02(1)	1.051E-01(2)	1.346E-01(3)	5.11E-01(4)	2.03E+01(5)		**1.492E-01(1)**	1.695E+00(4)	8.444E-01(2)	1.27E+00(3)	3.39E+02(5)
	15	**4.359E-02(1)**	5.702E-02(2)	8.240E-02(3)	8.06E-01(4)	1.80E+01(5)		1.024E-01(2)	**9.576E-02(1)**	1.732E-01(3)	9.39E+01(4)	2.36E+02(5)
		6.457E-02(1)	6.862E-02(2)	1.592E-01(3)	2.06E+00(4)	2.52E+01(5)		3.384E-01(2)	**2.902E-01(1)**	4.897E-01(3)	1.98E+02(4)	2.64E+02(5)
		7.977E-02(1)	1.077E-01(2)	4.672E-01(3)	6.31E+01(5)	2.59E+01(4)		**8.964E-01(1)**	9.217E-01(3)	9.605E-01(2)	3.24E+02(4)	3.45E+02(5)
DTLZ2	3	**2.531E-03(1)**	3.346E-03(3)	2.539E-03(2)	6.88E-02(5)	6.73E-02(4)	DTLZ4	4.233E-03(3)	3.266E-03(2)	**2.638E-03(1)**	6.87E-02(5)	6.66E-02(4)
		2.666E-03(2)	3.376E-03(3)	**2.617E-03(1)**	7.18E-02(5)	6.91E-02(4)		4.825E-03(2)	**4.692E-03(1)**	1.071E-02(3)	7.23E-02(5)	7.07E-02(4)
		2.828E-03(2)	3.418E-03(3)	**2.733E-03(1)**	7.44E-02(5)	7.10E-02(4)		**5.268E-03(1)**	2.935E-02(3)	2.938E-02(2)	9.40E-01(5)	5.27E-01(4)
	5	8.731E-03(2)	1.201E-02(3)	**7.951E-03(1)**	1.41E-01(4)	2.76E-01(5)		8.777E-03(2)	1.200E-02	**8.444E-03(1)**	1.42E-01(4)	2.60E-01(5)
		9.255E-03(2)	1.202E-02(3)	**8.236E-03(1)**	1.47E-0(4)	2.87E-01(5)		**9.338E-03(1)**	1.204E-02(2)	1.847E-02(3)	1.46E-01(4)	2.68E-01(5)
		9.715E-03(1)	1.205E-02(3)	1.003E-02(2)	1.56E-01(4)	2.92E-01(5)		**9.854E-03(1)**	1.212E-02(2)	2.778E-02(3)	1.61E-01(4)	5.30E-01(5)
	8	3.110E-02(2)	3.445E-02(3)	**2.094E-02(1)**	3.45E-01(4)	5.48E-01(5)		**2.274E-02(1)**	3.459E-02(3)	2.704E-02(2)	3.23E-01(4)	4.79E-01(5)
		2.574E-02(1)	3.769E-02(3)	3.600E-02(2)	3.73E-01(4)	6.03E-01(5)		**2.946E-02(1)**	3.715E-02(3)	2.991E-02(2)	3.31E-01(4)	4.96E-01(5)
		3.155E-02(1)	4.550E-02(2)	5.133E-02(3)	4.13E-01(4)	6.47E-01(5)		**3.369E-02(1)**	4.039E-02(3)	3.707E-02(2)	3.40E-01(4)	5.39E-01(5)
	10	**2.528E-02(1)**	4.477E-02(3)	3.643E-02(2)	4.11E-01(4)	6.78E-01(5)		**2.494E-02(1)**	4.440E-02(3)	3.734E-02(2)	4.19E-01(4)	6.76E-01(5)
		3.028E-02(1)	4.890E-02(2)	5.002E-02(3)	4.51E-01(4)	6.90E-01(5)		**2.764E-02(1)**	4.755E-02(3)	4.511E-02(2)	4.29E-01(4)	6.83E-01(5)
		3.345E-02(1)	5.646E-02 (2)	5.875E-02(3)	5.16E-01(4)	6.92E-01(5)		**3.364E-02(1)**	5.066E-02(2)	5.829E-02(3)	4.41E-01(4)	6.88E-01(5)
	15	**5.620E-02(1)**	9.018E-02 (3)	7.696E-02(2)	5.09E-01(4)	6.24E-01(5)		**5.449E-02(1)**	8.654E-02(3)	8.341E-02(2)	4.98E-01(4)	5.99E-01(5)
		5.915E-02(1)	9.541E-02(3)	9.159E-02(2)	5.29E-01(4)	8.64E-01(5)		**5.756E-02(1)**	9.560E-02(3)	8.993E-02(2)	5.03E-01(4)	6.10E-01(5)
		6.547E-02(1)	1.019E-01(3)	9.893E-02(2)	5.38E-01(4)	3.20E+00(5)		**6.046E-02(1)**	1.014E-01(3)	9.763E-02(2)	5.14E-01(4)	6.13E-01(5)

† (‡) indicates that the performance of the algorithm is significantly worse (better) than that of RIEA at a 0.05 confidence level according to the Wilcoxon's rank sum test.

Table 2. Statistical results (best/mean/worst) obtained by RIEA, MOEA/D, PICEA-G, GrEA and HypE in 20 independent runs on the DLTZ test suite in terms of the HV metric.

	m	RIEA	MOEA/D	PICEA-G	GrEA	HypE		RIEA	MOEA/D	PICEA-G	GrEA	HypE
DTLZ1	3	**0.9743**	0.9729	0.9707	0.9674	0.0000	DTLZ3	0.8718	**0.9310**	0.8315	0.9247	0.0000
		0.9690	**0.9694**	0.8559	0.9641	0.0000		0.6287	0.5660	0.0416	**0.9227**	0.0000
		0.9428	**0.9660**	0.3951	0.8280	0.0000		0.0463	0.0000	0.0000	**0.6212**	0.0000
	5	0.9977	**0.9978**	0.9805	0.9915	0.0000		0.9800	**0.9842**	0.3977	0.9630	0.0000
		0.9957	**0.9966**	0.7916	0.8445	0.0000		0.7458	0.4481	0.0199	**0.8081**	0.0000
		0.9938	**0.9946**	0.1648	0.5002	0.0000		0.0018	0.0000	0.0000	**0.5000**	0.0000
	8	**0.9995**	0.9965	0.9595	0.9991	0.0000		0.9294	0.9521	0.0000	**0.9535**	0.0000
		0.9959	0.8372	0.6577	**0.9980**	0.0000		0.5557	0.2421	0.0000	**0.7912**	0.0000
		0.9825	0.0000	0.0000	0.9027	0.0000		0.0312	0.0000	0.0000	**0.4986**	0.0000
	10	**0.9996**	0.9987	0.9184	0.9995	0.0000		**0.9859**	0.9546	0.0000	0.9622	0.0000
		0.9971	0.9370	0.7467	**0.9986**	0.0000		**0.7648**	0.2262	0.0000	0.7360	0.0000
		0.9901	0.2340	0.0680	0.5323	0.0000		0.4871	0.0000	0.0000	**0.5000**	0.0000
	15	**0.9987**	0.9812	0.9298	0.1725	0.0000		**0.9296**	0.8990	0.0000	0.0000	0.0000
		0.9810	0.9163	0.5061	0.0000	0.0000		0.3099	**0.4690**	0.0000+	0.0000	0.0000+
		0.9097	0.6327	0.0000	0.0000	0.0000		**0.2230**	0.0000	0.0000	0.0000	0.0000
DTLZ2	3	**0.9314**	0.9277	0.9292	0.9242	0.9257	DTLZ4	0.9294	**0.9297**	0.9279	0.9246	0.9264
		0.9254	0.9228	0.9256	0.9240	**0.9257**		0.9248	0.9175	0.8939	0.9241	**0.9262**
		0.9204	0.9162	0.9178	0.9237	**0.9255**		**0.9202**	0.7999	0.7965	0.5000	0.8005
	5	0.9882	0.9881	0.9886	**0.9904**	0.9879		0.9882	0.9881	0.9882	**0.9905**	0.9882
		0.9845	0.9857	0.9856	**0.9902**	0.9877		0.9863	0.9853	0.9682	**0.9904**	0.9880
		0.9793	0.9827	0.9824	**0.9901**	0.9875		0.9831	0.9828	0.9224	**0.9902**	0.9877
	8	0.9984	0.9624	0.9988	**1.0000**	0.9974		0.9980	0.9639	0.9975	**0.9991**	0.9980
		0.9936	0.9366	0.9886	**0.9997**	0.9966		0.9957	0.9428	0.9953	**0.9990**	0.9977
		0.9843	0.9184	0.9664	0.9893	**0.9958**		0.9927	0.9267	0.9911	**0.9990**	0.9976
	10	**0.9998**	0.9605	0.9938	0.9976	0.9990		**0.9999**	0.9671	0.9992	0.9997	0.9990
		0.9993	0.9393	0.9854	0.9964	0.9989		**0.9997**	0.9262	0.9873	0.9996	0.9989
		0.9983	0.9084	0.9690	0.9947	**0.9994**		0.9944	0.9035	0.9688	**0.9995**	0.9989
	15	**1.0000**	0.9069	0.9952	0.9995	0.9999		0.9954	0.9050	0.9974	**0.9996**	0.9991
		0.9997	0.8756	0.9767	0.9994	0.9970		**0.9998**	0.8616	0.9915	0.9995	0.9994
		0.9893	0.8170	0.9270	**0.9984**	0.0000		**0.9997**	0.8086	0.9843	0.9995	0.9991

5 Conclusion

This paper presented a multiobjective evolutionary algorithm, called RIEA, inspired by a set of randomly generated reference points, for many objective optimisation problems. A new fitness assignment scheme induced from these reference points is developed to integrate dominance and convergence-diversity information for effective ranking of solutions. A set of reference vectors is employed to divide the search space into subregions to facilitate diversity management. A restricted mating selection strategy which involves the construction of convergence elite set and diversity elite set at each subregion is proposed for selecting mating solutions to improve search efficiency. Empirical studies have shown that the subregion decomposition and the mating selection scheme based on the decomposition can indeed improve the search efficiency in terms of diversity and convergence.

References

1. Bader, J., Zitzler, E.: HypE: an algorithm for fast hypervolume-based many-objective optimization. Evol. Comput. **19**(1), 45–76 (2011)

2. Bosman, P.A., Thierens, D.: The balance between proximity and diversity in multiobjective evolutionary algorithms. IEEE Trans. Evol. Comput. **7**(2), 174–188 (2003)
3. Deb, K., Goyal, M.: A combined genetic adaptive search (GeneAS) for engineering design. Comput. Sci. Inf. **26**, 30–45 (1996)
4. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. Springer (2005)
5. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. IEEE Trans. Evol. Comput. **18**(4), 577–601 (2014)
6. Figueira, J., Liefooghe, A., Talbi, E.G., Wierzbicki, A.: A parallel multiple reference point approach for multobjective optimization. Eur. J. Oper. Res. **205**(2), 390–400 (2010)
7. Jiang, S., Ong, Y.S., Zhang, J., Feng, L.: Consistencies and contradictions of performance metrics in multiobjective optimization. IEEE Trans. Cybern. **44**(12), 2391–2404 (2014)
8. Li, B., Li, J., Tang, K., Yao, X.: Many-objective evolutionary algorithms: a survey. ACM Comput. Surv. (CSUR) **48**(1), 13 (2015)
9. Li, K., Deb, K., Zhang, Q., Kwong, S.: An evolutionary many-objective optimization algorithm based on dominance and decomposition. IEEE Trans. Evol. Alg. **19**(5), 694–716 (2015)
10. Li, K., Kwong, S., Zhang, Q., Deb, K.: Interrelationship-based selection for decomposition multiobjective optimization. IEEE Trans. Cybern. **45**(10), 2076–2088 (2015)
11. Liu, H.L., Gu, F., Zhang, Q.: Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. IEEE Trans. Evol. Comput. **18**(3), 450–455 (2014)
12. Moen, H.J., Hansen, N.B., Hovland, H., Tørresen, J.: Many-objective optimization using taxi-cab surface evolutionary algorithm. In: Evolutionary Multi-Criterion Optimization, pp. 128–142. Springer (2013)
13. Kata, P., Yao, X.: A new multi-objective evolutionary optimisation algorithm: the two-archive algorithm. In: 2006 International Conference on Computational Intelligence and Security, Vol. 1, pp. 286–291. IEEE (2006)
14. Wagner, T., Beume, N., Naujoks, B.: Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In: Proceedings of Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science, vol. 4403, Matsushima, Japan, pp. 742–756. Springer, Heidelberg (2007)
15. Wang, R., Purshouse, R.C., Fleming, P.J.: Preference-inspired coevolutionary algorithms for many-objective optimization. IEEE Trans. Evol. Comput. **17**(4), 474–494 (2013)
16. Yang, S., Li, M., Liu, X., Zheng, J.: A grid-based evolutionary algorithm for many-objective optimization. IEEE Trans. Evol. Comput. **17**(5), 721–736 (2013)
17. Yuan, Y., Xu, H., Wang, B., Yao, X.: A new dominance relation-based evolutionary algorithm for many-objective optimization. IEEE Trans. Evol. Comput. **20**(1), 16–37 (2016)
18. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
19. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans. Evol. Comput. **3**(4), 257–271 (1999)

Learning

Generation of Reducts and Threshold Functions Using Discernibility and Indiscernibility Matrices for Classification

Naohiro Ishii[1(✉)], Ippei Torii[1], Kazunori Iwata[2], Kazuya Odagiri[3], and Toyoshiro Nakashima[3]

[1] Aichi Institute of Technology, Toyota, Japan
{ishii,mac}@aitech.ac.jp
[2] Aichi University, Nagoya, Japan
kazunori@vega.aichi-u.ac.jp
[3] Sugiyama Jyogakuen University, Nagoya, Japan
{odagiri,nakasima}@sugiyama-u.ac.jp

Abstract. Dimension reduction of data is an important issue in the data processing and it is needed for the analysis of higher dimensional data in the application domain. Reduct in the rough set is a minimal subset of features, which has the same discernible power as the entire features in the higher dimensional scheme. In this paper, generations of reducts and threshold functions are developed for the classification system. The reduct followed by the nearest neighbor method or threshold functions is useful for the reduct classification system. For the classification, a nearest neighbor relation with minimal distance proposed here has a fundamental information for classification. Then, the nearest neighbor relation plays a fundamental role on the discernibility and in discernibility matrices, in which the indiscernibility matrix is proposed here to test the sufficient condition for reduct and threshold function. Then, generation methods for the reducts and threshold functions based on the nearest neighbor relation are proposed here using Boolean operations on the discernibility and the indiscernibility matrices.

Keywords: Reduct · Threshold function · Nearest neighbor relation · Discernibility matrix · Indiscernibility matrix

1 Introduction

Rough sets theory firstly introduced by Pawlak [1, 2] provides us a new approach to perform data analysis, practically. Up to now, rough set has been applied successfully and widely in machine learning and data mining. The need to manipulate higher dimensional data in the web and to support or process them gives rise to the question of how to represent the data in a lower-dimensional space to allow more space and time efficient computation. Thus, dimension reduction of data still remains as an important problem. An important task in rough set based data analysis is computation of the attributes or feature reducts for the classification [1, 2]. By Pawlak's [1, 2] rough set theory, a reduct is a minimal subset of features, which has the discernibility power as

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_13

using the entire features. Skowlon [3, 4] developed the reduct derivation by using the Boolean expression for the discernibility of data. In this paper, generations of reducts and threshold functions are developed for the classification system. The reduct followed by the nearest neighbor method or threshold functions is useful for the reduct classification system [7, 8]. So, a new concept for the efficient generation of reducts is expected from the point of view of the classification system. Nearest neighbor relation with minimal distance between different classes proposed here has a basic information for classification. For the classification of data, a nearest neighbor method [6–12] is simple and effective one. As a classification method, threshold function is well known [12]. Recent studies of threshold functions are of fundamental interest in circuit complexity, game theory and learning theory [11]. We have developed further analysis for the generation of reducts and threshold functions by using the nearest neighbor relations and the Boolean reasoning on the discernibility and indiscernibility matrices. We propose here new generation method for reducts and threshold functions based on the nearest neighbor relation with minimal distance using discernibility and the indiscernibility matrices, in which the indiscernibility matrix tests sufficient conditions for them. Thus, the generation methods based on the nearest neighbor relation are useful for the classified data with groups.

2 Boolean Reasoning of Reducts

Skowron proposed to represent a decision table in the form of the discernibility matrix [3, 4]. This representation has many advantages, in particular it enables simple computation of the core, reducts and other concepts [1–3]. The discernibility matrix is computed for pairs of instances and stores the different variables (attributes) between all possible pairs of instances that must remain discernible.

Definition 1. The discernibility matrix $M(T)$ is defined as follows. Let $T = \{U, A, C, D\}$ be a decision table, with $U = \{x_1, x_2, \ldots . x_n\}$, set of instances. A is a subset of C called condition, and D is a set of decision classes. By a discernibility matrix of T, denoted by $M(T)$, which is $n \times n$ matrix defined as

$$\begin{aligned} c_{ij} = \{a \in C : a(x_i) \neq a(x_j) \\ \wedge (d \in D, d(x_i) \neq d(x_j))\}\ i, j = 1, 2, \ldots . n, \end{aligned} \tag{1}$$

where U is the universe of discourse and C is a set of features or attributes.

Definition 2. A discernibility function f_A for A is a propositional formula of m Boolean variables, $a_1^*, \ldots, a_m^*$, corresponding to the attributes $a_1, \ldots, a_m$, defined by

$$f_A(a_1^*, \ldots, a_m^*) = \bigwedge_{1 \le j < i \le m} \ \bigvee_{c \in c_{ij}^*, c_{ij}^* \neq \phi} c_{ij}^*, \tag{2}$$

where $c_{ij}^* = \{a^* : a \in c_{ij}\}$ [3]. In the sequel, a_i is used instead of a_i^*, for simplicity. The symbol $\wedge$ shows Boolean product, while $\vee$ shows Boolean sum.

An example of decision table of the data set is shown in Table 1. The left side data in the column in Table 1 as shown in, $\{x_1, x_2, x_3, \ldots, x_7\}$ is a set of instances, while the data $\{a, b, c, d\}$ on the upper row, shows the set of attributes of the instance. In Table 2, the discernibility matrix of the decision table in Table 1 is shown. In case of instance x_1, the value of the attribute a, is $\mathrm{a}(x_1) = 1$. That of the attribute b, is $\mathrm{b}(x_1) = 0$. Since $\mathrm{a}(x_1) = 1$ and $\mathrm{a}(x_5) = 2$, $\mathrm{a}(x_1) \neq \mathrm{a}(x_5)$ holds.

Table 1. Decision table of data example (instances)

Attribute	a	b	c	d	class
x_1	1	0	2	1	+ 1
x_2	1	0	2	0	+1
x_3	2	2	0	0	-1
x_4	1	2	2	1	-1
x_5	2	1	0	1	-1
x_6	2	1	1	0	+1
x_7	2	1	2	1	-1

The discernibility function is represented by taking the combination of the disjunction expression of the discernibility matrix. In Table 2, the item (b, c, d) in the second row and the first column, implies $b + c + d$ in the Boolean sum expression, which shows the attribute b or c or d appear for the discrimination between instances x_1 and x_3 [3].

3 Generation of Reducts Based on Nearest Neighbor Relation and External Set

We can define a new concept, a nearest neighbor relation with minimal distance, δ. Instances with different classes are assumed to be measured in the metric distances for the nearest neighbor classification.

Definition 3. A nearest neighbor relation with minimal distance is a set of pair of instances, which are described in the equation

$$\{(X_i, X_j) : \beta(X_i) \neq \beta(X_j) \wedge |X_i - X_j| \leq \delta\}, \tag{3}$$

where $|X_i - X_j|$ shows the distance between X_i and X_j, and δ is the minimal distance. β is a decision function for the class. Then, X_i and X_j are called to be a nearest neighbor relation with minimal distance δ.

Table 2. Discernibility matrix of the decision table in Table 1

	x_1	x_2	x_3	x_4	x_5	x_6
x_2	—					
x_3	a,b,c,d	a,$\underline{\text{b,c}}$				
x_4	$\underline{\text{b}}$	$\underline{\text{b,d}}$	—			
x_5	a,b,c	a,b,c,d	—	—		
x_6	—	—	b,c	a,b,c,d	c,d	
x_7	a,b	a,b,d	—	—	—	c,d

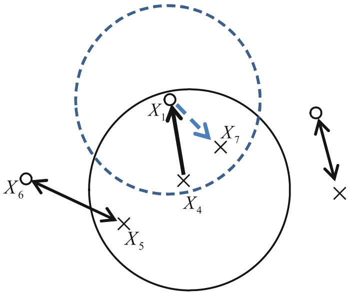


Fig. 1. Search of nearest neighbor relations

To find minimal nearest neighbor relation, the divide and conquer algorithm [5, 12] is applied to the array of data in Table 1 with the search of the nearest data so as to be classified to different classes.

In Table 1, $(x_6, x_7), (x_5, x_6), (x_1, x_7)$ and (x_3, x_6) are elements of the relation with a distance $\sqrt{2}$. Thus, a nearest neighbor relation with minimal distance $\sqrt{2}$ becomes

$$\{(x_6, x_7), (x_5, x_6), (x_1, x_7), (x_3, x_6)\} \tag{4}$$

Here, we want to introduce the nearest neighbor relation on the discernibility matrix. Assume that the set of elements of the nearest neighbor relation are $\{nn_{ij}\}$. Then, the following characteristics are shown.

Lemma 1. Respective Boolean term consisting of the set $\{nn_{ij}\}$ becomes a necessary condition to be reducts in the Boolean expression.

Lemma 2. Boolean product of respective terms corresponding to the set $\{nn_{ij}\}$ becomes a necessary condition to be reducts in the Boolean expression.

Lemma 3. Reducts in the Boolean expression are included in the Boolean term of Lemma 1 and the Boolean product in Lemma 2.

Figure 2 shows that nearest neighbor relation with classification is a necessary condition in the Boolean expression for reducts, but not sufficient condition. The distance δ of the nearest neighbor relation in the Eq. (3) is compared with the distance δ' of the relation in the following theorem.

Theorem 1. If the distance δ is greater than the δ', i.e., $\delta > \delta'$ in the Eq. (3), the Boolean expression of the case of δ' includes that of δ.

This is by the reason that the Boolean expression of the nearest neighbor relation is consists of the Boolean product of variables of the relation. The number of variables in the distance δ' are less than that of δ. Thus, the nearest neighbor relation with distance δ' includes the ellipse of δ in Fig. 2. Two sets of attributes (variables), [A] and [B] are defined as follows.

[A]: Set of elements in the discernibility matrix includes those of any respective element in $\{nn_{ij}\}$ and those of elements absorbed by $\{nn_{ij}\}$ in the Boolean expression

[B]: Set of elements in the discernibility matrix, which are not absorbed from those of any respective element in $\{nn_{ij}\}$.

The Boolean sum of attributes is absorbed in the Boolean sum element with the same fewer attributes in the set $\{nn_{ij}\}$. As an example, the Boolean sum of (a + b + c) in the set [A] is absorbed in the Boolean sum of (a + b) in the set $\{nn_{ij}\}$.

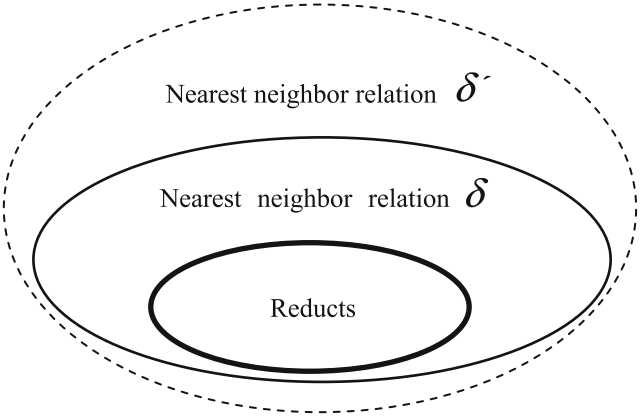

Nearest neighbor relation δ'
Nearest neighbor relation δ
Reducts

Fig. 2. Boolean condition of nearest neighbor relations and reducts

Lemma 4. Within set [B], the element with fewer attributes (variables) plays a role of the absorption for the element with larger attributes (variables).

Theorem 2. Reducts are derived from the nearest neighbor relation by the absorption of variables in set [B] terms in the Boolean expression.

In the relation (x_1, x_7) in Table 1, the x_1 in the class +1 is nearest to the is x_7 in the class −1. Similarly, (x_5, x_6) and (x_6, x_7) are nearest relations. Then, variables of the set [A] in these relations are shown in shaded cells in Table 2 of the discernible matrix. The Boolean product of these four terms becomes

$$(a+b)\cdot(b+c)\cdot(c+d) = b\cdot c + b\cdot d + a\cdot c, \tag{5}$$

which becomes a candidate of reducts. The third term in the Eq. (5) is absorbed by the product of variable {b} of the set [B] and the Eq. (5). The final reducts equation becomes

$$b\cdot c + b\cdot d \tag{6}$$

Thus, reducts $\{b, c\}$ and $\{b, d\}$ are obtained finally. To search final reducts in the Eq. (5), the following Boolean reasoning is considered at the set [A] of the nearest neighbor relations. At the step of the Boolean Eq. (4), some dominant Boolean variables are searched from the bottom up. From the Eq. (5), three Boolean minterms $b\cdot c$, $b\cdot d$ and $a\cdot c$ are derived from the nearest neighbor relations. Using the set [B], the minterm $a\cdot c$ is removed. Thus, the Eq. (6) is obtained.

4 Generation of Reducts Based on Nearest Neighbor Relation and Indiscernibility Matrix

In this section, we propose another generation method of reducts, which is based on nearest neighbor relation and indiscernibility matrix proposed here. The external set in the previous Sect. 3 is replaced to indiscernibility matrix.

Definition 4. Indiscernibility matrix is defined to be $IM(T)$, which is $n \times n$ matrix defined as

$$\begin{aligned} c_{ij} = \{&a \in C : a(x_i) = a(x_j) \\ &\wedge (d \in D, d(x_i) \neq d(x_j))\}\ i, j = 1, 2, \ldots n, \end{aligned} \tag{7}$$

where U is the universe of discourse, C is a set of features.

The difference between $M(T)$ in the Eq. (1) and $IM(T)$ in the Eq. (7), is shown in the following. Attributes $a(x_i) \neq a(x_j)$ holds in $M(T)$, while $a(x_i) = a(x_j)$ holds in $IM(T)$. Since the Boolean operation in the element of the indiscernible matrix is AND, the element value (a, c, d) between x_1 and x_4 shows Boolean product $a\cdot c\cdot d$. The Boolean product $a\cdot b\cdot c$ also implies $a\cdot c$ or $c\cdot d$ or $a\cdot d$.

In the Sect. 2, the nearest neighbor relation is derived from Table 1. In the relation (x_1, x_7) in Table 1, the x_1 in the class +1 is nearest to the is x_7 in the class −1. Similarly, (x_5, x_6) and (x_6, x_7) are nearest neighbor relations. Then, variables of the nearest neighbor relations are shown in shading in Table 2 of the discernible matrix. The Boolean product of these four terms becomes

$$(a+b)\cdot(b+c)\cdot(c+d) = b\cdot c + b\cdot d + a\cdot c \tag{8}$$

Table 3. Indiscernibility matrix in Table 1

	x_1	x_2	x_3	x_4	x_5	x_6
x_2	—					
x_3	—	d				
x_4	a,c,d	a,c	—			
x_5	d	—	—	—		
x_6	—	—	a,d	—	a,b	
x_7	c,d	c	—	—	—	a,b

In the three minterms in the above Eq. (8), the minterm $a \cdot c$ in Eq. (8) is checked in the indiscernibility matrix Table 3. The $(a \cdot c)$ is found between x_1 and x_4, also between x_2 and x_4. The minterm $a \cdot c$ cannot discriminate instances between x_1 and x_4, also between x_2 and x_4. Then, the minterm $a \cdot c$ is removed from the Eq. (8). Thus, the reducts

$$b \cdot c + b \cdot d \tag{9}$$

is obtained, which is the same derived in the Eq. (6). Thus, reducts $\{b, c\}$ and $\{b, d\}$ are obtained.

4.1 Relation Between Set [B] and Indiscernibility Matrix

For the one generation method of reducts, set [A] and set [B] are defined in Sect. 3 and the other generation method uses indiscernibility matrix in Sect. 4. The set [B] in Table 2, becomes $\{b, (b, d)\}$, which is remained after Boolean absorption by the set of nearest neighbor relations. In the set [B], the variable (b, d) is expressed in Boolean form as $(b + d)$. The variable b of Boolean product with $(b + d)$ becomes by Boolean absorption law,

$$b \cdot (b + d) = b + b \cdot d = b$$

Thus, the variable b represents the set [B]. The variable b multiplied by the Boolean form of the set [A] derived from the nearest neighbor relation becomes

$$b \cdot (a \cdot c + b \cdot c + b \cdot d) = b \cdot c + b \cdot d$$

The element variable (a, c, d) in indiscernibility table, which is placed in the same element in discernibility matrix, removes the Boolean minterm $a \cdot c$. Thus, the final Boolean reduct form becomes

$$b \cdot c + b \cdot d$$

Theorem 3. The Boolean absorption of the variables in the set [B] multiplied by the Boolean forms derived from the nearest neighbor relations generate reducts, which are also generated by directly removing Boolean variables in the indiscerniblity matrix from the Boolean forms of the nearest neighbor relations.

5 Generation of Threshold Functions Using Discernibility and Indiscernibility Matrices

The nearest neighbor relation is also applicable to the generation of threshold functions. The threshold function is a Boolean function based on the n-dimensional cube with 2^n vertices of n components of 1 or 0. The function f is characterized by the hyperplane $WX - \theta$ with the weight vector $W(= (w_1, w_2, \ldots, w_n))$ and threshold θ. The X is a vertex of the cube 2^n. In the following, the threshold function is assumed to be positive and canonical threshold function, in which the Boolean variables hold the partial order [12].

Definition 5. The nearest neighbor relation (X_i, X_j) on the threshold function is defined from Eq. (3),

$$\{(X_i, X_j) : \beta(X_i) \neq \beta(X_j) \wedge |X_i - X_j| \leq \delta(= 1)\}, \tag{10}$$

where $\delta = 1$ shows one bit difference between X_i and X_j in the Hamming distance (also in the Euclidean distance).

Definition 6. The boundary vector X is defined to be the vector which satisfies

$$|WX - \theta| \leq |WY - \theta| \text{ for the } X(\neq Y \in 2^n) \tag{11}$$

Theorem 4. The boundary vector X becomes an element of nearest neighbor relation in the threshold function.

This is proved, since the boundary vector X is the nearest data to the hyperplane, which divides the true data and the false data. The nearest neighbor relation is not necessarily boundary vector. Since the boundary vectors determine the hyperplane of the threshold function, the nearest neighbor relation also characterizes the threshold function. The data set is called to be admissible set of f, if the set realizes a threshold function f.

Theorem 5. The set of the element of the nearest neighbor relation is admissible set of the threshold function.

Theorem 6. The vectors X_i and X_j in the nearest neighbor relation (X_i, X_j) are the adjacent vectors, each of which belongs to different class through the hyperplane.

This theorem shows the line between adjacent vectors X_i and X_j with one bit difference is crossed by the hyperplane.

Theorem 7. The nearest neighbor relations $\{(X_i, X_j)\}$ in a threshold function f is unique in f.

This is proved by the contradiction. Assume a threshold function has two nearest neighbor relations NNR_A and NNR_B, in which $(X_{iA}, X_{jA}) \neq (X_{iB}, X_{jB})$ holds. First, since X_{iB} is not as true data in the NNR_A, it exists as a false data. Then (X_{iB}, X_{jB}) does not make nearest neighbor relation of f, which contradicts the assertion. Second, when the partial ordering $X_{iA} \prec X_{iB}$ holds [12], X_{iB} does not make nearest neighbor relation of f with X_{jB} in one bit distance. Thus, $\{(X_i, X_j)\}$ is unique in the given threshold function f. The data 3-dimensional cube is shown in Fig. 3, in which true data with the black circle belongs to +1 class, while false data with the white circle belongs to 0 class. In Fig. 3, a true valued data (101) has nearest neighbor relations as {(101), (001)} and {(101), (100)} as shown in shaded cells in Table 4. The Boolean reasoning in these relations becomes a Boolean product $x_1 \cdot x_3$ of the respective relation x_1 and x_3 in Table 4. The Boolean product, called minterm $x_1 \cdot x_3$ satisfies to be 1 for (101), while to be 0 for (001) and (100). Similarly, a true valued valued data (110) has nearest neighbor relations as {(110), (100)} and {(110), (010)}. From these relations, the Boolean miterm $x_2 \cdot x_3$ is generated. Finally, the minterm with one variable x_1 is generated from the relation {(111), (011)}.

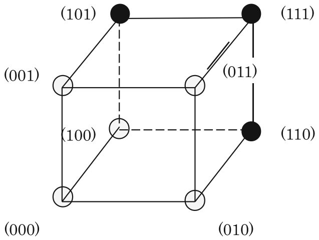


Fig. 3. Example of a threshold function in 3-dimensional cube

Table 4. Discernibility matrix of nearest neighbor data in Fig. 3

	● (101)	● (110)	● (111)
○ (001)	x_1	···	···
○ (100)	x_3	x_2	···
○ (010)	···	x_3	···
○ (011)	···	···	x_1

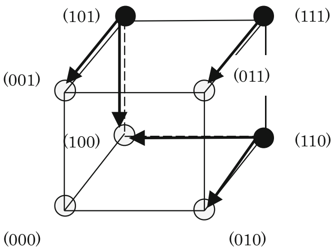


Fig. 4. Arrows show nearest neighbor relations in 3-dimensional cube

Similarly, the difference between (101) and (100) is x_3. For the Boolean realization of the difference of these variables is performed by the Boolean product $x_1 \cdot x_3$ also as shown in Fig. 5. The Boolean product, called minterm $x_1 \cdot x_3$ satisfies to be 1 for (101), while to be 0 for (001) and (100). Similarly, a true valued valued data (110) has nearest neighbor relations as {(110), (100)} and {(110), (010)}. From these relations, the Boolean miterm $x_2 \cdot x_3$ is generated. Finally, the minterm with one variable x_1 is generated from the relation {(111), (011)}. But, this minterm is removed from the indiscernibility matrix in Table 5.

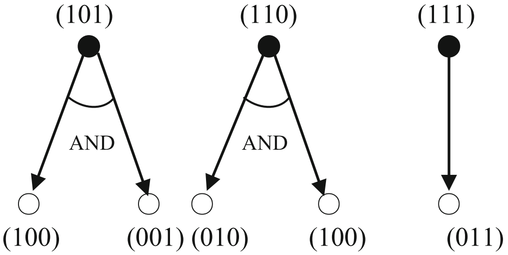


Fig. 5. Boolean operation for nearest neighbor relation

Table 5. Indiscernibility matrix of data in Fig. 3

	● (101)	● (110)	● (111)
○ (001)	$x_2 \cdot x_3$	$\cdots$	x_3
○ (100)	$x_1 \cdot x_2$	$x_1 \cdot x_3$	x_1
○ (010)	$\cdots$	$x_2 \cdot x_3$	x_2
○ (011)	x_3	x_3	$x_2 \cdot x_3$

In discernibility matrix for making threshold function for classification, it is necessary to perform AND operation of the terms in the column, while to perform OR operation among different columns. In Table 4, the difference of variables between ●(101) and ○(001) is x_1, in which $x_1 = 1$ in (101). The minterm x_1 in the relation {(111), (011)} in Table 4, which is shaded in the cell. Then, the minterm x_1 is removed from the Boolean sum from Table 4. Thus, the Boolean function obtained is

$$f = x_1 \cdot x_2 + x_1 \cdot x_3 \tag{12}$$

The Boolean function f becomes a threshold function, since a hyperplane exists to satisfy the Eq. (12).

6 Conclusion

Nearest neighbor relation developed here is the set of pair elements with minimal distance, which classify between different classes. Reduct is introduced as the minimal set for the data classification in the rough set theory. Threshold functions are of fundamental interest in circuit complexity, game theory and learning theory. This paper develops the role of the nearest neighbor relations for the classification of data, which is proposed here. Then, generation of the reducts and threshold functions based on the nearest neighbor relations using the discernibility and the indiscernibility matrices is developed for reduct classification system. The necessary condition for the generation of reducts and threshold functions is derived from the discernibility matrix with nearest neighbor relations, while the indiscernibility matrix is proposed to test the sufficient conditions for reducts and threshold functions.

References

1. Pawlak, Z.: Rough sets. Int. J. Comput. Inf. Sci. **11**, 341–356 (1982)
2. Pawlak, Z., Slowinski, R.: Rough set approach to multi-attribute decision analysis. Eur. J. Oper. Res. **72**, 443–459 (1994)
3. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Intelligent Decision Support- Handbook of Application and Advances of Rough Sets Theory, pp. 331–362. Kluwer Academic Publishers, Dordrecht (1992)
4. Skowron, A., Polkowski, L.: Decision algorithms, a survey of rough set theoretic methods. Fundamenta Informatica **30**(3-4), 345–358 (1997)
5. Meghabghab, G., Kandel, A.: Search Engines, Link Analysis, and User's Web Behavior. Springer, Heidelberg (2008)
6. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)
7. Ishii, N., Morioka, Y., Bao, Y., Tanaka, H.: Control of variables in reducts-kNN classification with confidence. In: KES 2011. LNCS, vol. 6884, pp. 98–107. Springer (2011)
8. Ishii, N., Torii, I., Bao, Y., Tanaka, H.: Modified reduct nearest neighbor classification. Proc. ACIS-ICIS, IEEE Comp. Soc. 310–315 (2012)

9. Ishii, N., Torii, I., Mukai, N., Iwata, K., Nakashima, T.: Classification on nonlinear mapping of reducts based on nearest neighbor relation. Proc. ACIS-ICIS IEEE Comp. Soc. 491–496 (2015)
10. Levitin, A.V.: Introduction to the Design and Analysis of Algorithms. Addison Wesley, Boston (2002)
11. De, A., Diakonikolas, I., Feldman, V., Servedio, R.A.: Nearly optimal solutions for the chow parameters problem and low-weight approximation of halfspaces. J. ACM **61**(2), 11:1–11:36 (2014)
12. Hu, S.T.: Threshold Logic. University of California Press, Berkeley (1965)

Adaptive Noise Cancelation Using Fuzzy Brain Emotional Learning Network

Qianqian Zhou[1], Chih-Min Lin[1,2], and Fei Chao[1](✉)

[1] Fujian Provincial Key Laboratory of the Brain-like Computing,
Cognitive Science Department, School of Informatics, Xiamen University,
Xiamen, Fujian, People's Republic of China
fchao@xmu.edu.cn

[2] Department of Electrical Engineering, Yuan Ze University,
Chung-Li 320, Tao-Yuan, Taiwan

Abstract. This paper proposes a fuzzy brain emotional learning network for adaptive noise cancelation. The proposed network is based on brain emotional learning algorithm which is developed according to the emotional learning process of mammalian and the fuzzy inference is added for better ability to handle uncertainties. Parameters in the network are modified online by the derived adaption laws. In addition, a stable convergence is guaranteed by utilizing the Lyapunov stability theorem. Finally, in order to demonstrate the performance of the proposed filter, it is applied in a signal processing application where different source signals and noise signals are used. A comparison between the proposed method, Least mean square algorithm and a fuzzy cerebellar model articulation controller filter shows that the proposed method can converge faster even when the source signal is corrupted severely.

Keywords: Emotional learning · Fuzzy inference · Adaptive noise cancelation

1 Introduction

Adaptive noise cancelation (ANC) has been applied in widespread fields including control, image processing and communications [1,2]. Among the various ANC systems, the adaptive linear filter is the most widely used one for its simple structure and low hardware implementation cost [3]. However, in most situations where nonlinear phenomena appear, the linear filter cannot achieve satisfactory performance [4]. Therefore, nonlinear adaptive filters have been developed, such as Kalman filter and the Volterra filter [5,6]. Meanwhile, neural network and fuzzy inference system have been popular among researchers in recent two decades, many successful filters have been developed based on them. For example, neural networks have been applied as channel equalization and noise cancelation and they have been shown to achieve more satisfactory performance

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_14

than non-neural-network-based adaptive filters [7]. However, the learning speed of neural networks is usually very slow since all the weights are modified during each training epoch [8,9].

In [10,11], a cerebellar model articulation controller (CMAC) which imitates cerebellums of human is developed. A CMAC usually converges faster than other methods for its great generalization ability and fast learning speed [12]. Filters based on CMAC are developed, [8] shows it can achieve better performance than linear filters. However, a main disadvantage of CMAC is that the output weights are singleton constants and it hasn't considered the emotion in the brain. In 1992, LeDoux found the connection between a stimulus and the corresponding emotional reaction in a part of brain called amygdala [13,14]. Therefore, in recent years, people are trying to add emotions into networks because emotions are necessary elements in intelligent control [15]. Basically, brain emotional learning (BEL) is an algorithm which is based on the computational model of emotional processing [16]. There is an orbitofrontal cortex and an amygdala in the brain, the former is a sensory neural network which has self-learning ability and the latter is an emotional neural network which undergoes stimulation by external factors and has an indirect impact on the sensory neural network. The output of a BEL controller is combined with these two networks which effect each other. Thus, it can perform well when disturbance exists and reduce the tracking error effectively.

In this paper, a fuzzy brain emotional learning network (FBELN) is developed for higher speed to converge and better filtering ability even the signal-to-noise ratio (SNR) is poor. The fuzzy inference system is added to provide the ability of nonlinear curve fitting for the network. The parameters of FBELN are all modified online by the derived adaptation laws and the stability is guaranteed by applying Lyapunov theorem. Finally, in order to verify the effectiveness of the proposed FBELN, different source signals and different intensity of noise signals are used in filtering process. Analysis and comparison between the new approach, Least mean square (LMS) and FCMAC for the noise cancelation will be also presented.

The rest of this paper is organized as follows. Section 2 describes the structure of FBELN network. The learning algorithms of parameters with convergence analysis are introduced in Sect. 3. Stimulation results are presented in Sect. 4 and the conclusion is finally given in Sect. 5.

2 Structure of FBELN

Brain emotional learning is an algorithm deriving from the emotional processing in the brain. There are four main components in the algorithm as shown in Fig. 1 [13]. The sensory input is distributed through orbitofrontal cortex and amygdala after passing through the thalamus and sensory cortex. The Thalamus passes the maximum sensory input among all the inputs to amygdala and the sensory cortex is stimulated as a computational delay [15]. The orbitofrontal cortex is a sensory neural network with self-learning ability and the amygdala is an emotional neural network which undergoes stimulation by external factors and has an indirect impact on the sensory neural network [17,18]. Finally, the output of the model is generated from the difference between the amygdala and the orbitofrontal cortex.

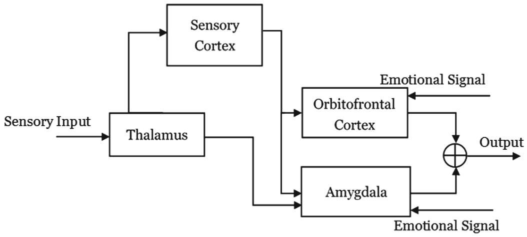


Fig. 1. Block diagram of simplified brain emotional learning system

In this paper, a novel form of BEL with a fuzzy inference system added is proposed. The fuzzy inference system has the ability to mimic the reasoning process of a human and it provides the ability of nonlinear curve fitting for the network [19]. The structure of FBELN is shown in Fig. 2. In the following, the proposed network is described in detail.

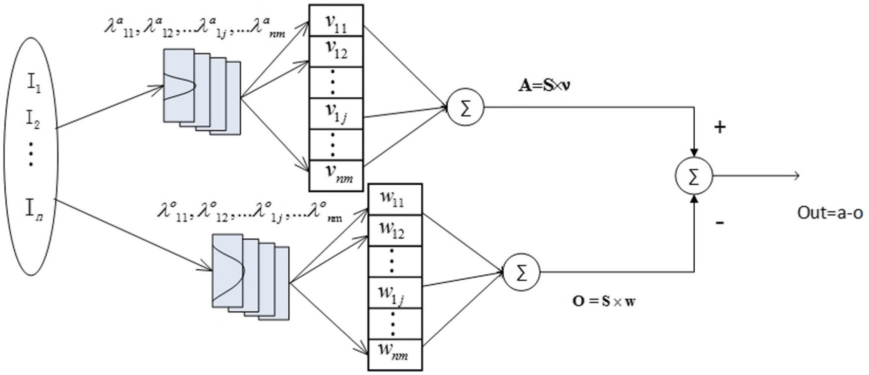

$\lambda^a_{11}, \lambda^a_{12}, \ldots \lambda^a_{1j}, \ldots \lambda^a_{nm}$
I_1
I_2
I_n
v_{11}
v_{12}
v_{1j}
v_{nm}
Σ
A=S×v
+
Σ
-
Out=a-o
$\lambda^o_{11}, \lambda^o_{12}, \ldots \lambda^o_{1j}, \ldots \lambda^o_{nm}$
w_{11}
w_{12}
w_{1j}
w_{nm}
Σ
O = S × w

Fig. 2. Structure of FBELN

The fuzzy inference system for amygdala is:

R^l: If I_1 is λ^a_{1j} and I_2 is $\lambda^a_{2j}, \ldots, I_n$ is λ^a_{nj}, then

$$\mathbf{A} = \mathbf{S} \times \mathbf{v}_{ij} \quad for \quad i = 1, 2, \ldots, n; j = 1, 2, \ldots, m \tag{1}$$

and the fuzzy inference system for orbitofrontal cortex is:

R^l: If I_1 is λ^o_{1j} and I_2 is $\lambda^o_{2j}, \ldots, I_n$ is λ^o_{nj}, then

$$\mathbf{O} = \mathbf{S} \times \mathbf{w}_{ij} \quad for \quad i = 1, 2, \ldots, n; j = 1, 2, \ldots, m \tag{2}$$

where $\mathbf{v}_{ij}$ and $\mathbf{w}_{ij}$ are the weights of the amygdala and the orbitofrontal cortex, $\mathbf{A}$ and $\mathbf{O}$ are the outputs of them and $\mathbf{S}$ represents the sensory input. The signal propagation of each layer is described as:

Layer 1: In this layer, the signals are transmitted to the next layer with no change. The output of this layer can be described as:

$$y_i{}^{(1)} = I_i \quad i = 1, 2, \ldots, n \tag{3}$$

where i is the dimension of the input vector.

Layer 2: The fuzzy inference process is adopted in this layer where each node serves as a membership function. The Gaussian function is adopted here as membership function which can be represented as:

$$\lambda_{ij}^{a} = \exp(-\frac{(y_i{}^{(1)} - m_{ij}^{a})^2}{\sigma_{ij}^{a}{}^2}) \quad i = 1, \ldots n; j = 1, \ldots m \tag{4}$$

$$\lambda_{ij}^{o} = \exp(-\frac{(y_i{}^{(1)} - m_{ij}^{o})^2}{\sigma_{ij}^{o}{}^2}) \quad i = 1, \ldots n; j = 1, \ldots m\ Fig : 5 \tag{5}$$

where m_{ij}^{a}, m_{ij}^{o} and σ_{ij}^{a}, σ_{ij}^{o} are the mean and standard deviation of the Gaussian functions during the fuzzification process in amygdala and orbitofrontal cortex, respectively.

Layer 3: The outputs for the sensory network and the emotional network are the product of weights and the output of fuzzification process in layer 2, and is expressed as:

$$\mathbf{A} = \sum_{i=1}^{n} \sum_{j=1}^{m} s_{ij} \times v_{ij} \tag{6}$$

$$\mathbf{O} = \sum_{i=1}^{n} \sum_{j=1}^{m} s_{ij} \times w_{ij} \tag{7}$$

where s_{ij} is the Gaussian function of prefrontal system input, A is the amygdala output and v_{ij} is the amygdala weight, O and w_{ij} are the orbitofrontal cortex output and the orbitofrontal cortex weight, respectively.

Layer 4: The output of FBELN is the difference between the amygdala and the orbitofrontal cortex which can be expressed as:

$$Out_{FBELN} = A - O \tag{8}$$

3 Learning Algorithm of the Adaptive FBELN Filter

3.1 Learning Algorithm

The update algorithms of weights v_{ij} and w_{ij} are given by:

$$\Delta v_{ij} = \alpha[s_{ij} \times (max[0, ES - A])] \tag{9}$$

$$\Delta w_{ij} = \beta[s_{ij} \times (Out_{FBELN} - ES)] \tag{10}$$

where α and β are the learning rates of Δv_{ij} and Δw_{ij}, respectively, ES is the emotional signal and which can be given by:

$$ES = b \times I_i + c \times Out_{FBELN} \tag{11}$$

where b and c are the gains, I_i and Out_{FBELN} are the input and output of FBELN. The emotional signal reflects the degree of satisfaction with the performance of the system.

The parameters in the Gaussian function are updated by back-propagation (BP) algorithm. The adaptive regulation is described as follows, the loss function is defined as:

$$E(k) = \frac{1}{2}(d(k) - y(k))^2 = \frac{1}{2}e^2(k) \tag{12}$$

where $e(k)$ is the error signal, indicating the difference between the desired response $d(k)$ and the filter's output $y(k)$. Based on the loss function, the parameters learning algorithm using gradient descent method in the sensory network can be described as:

$$\begin{aligned}\Delta m_{ij}^a =& -\eta_{m^a}\frac{\partial E}{\partial m_{ij}^a} = -\eta_{m^a}\frac{\partial E}{\partial \lambda_{ij}^a}\cdot\frac{\partial \lambda_{ij}^a}{\partial m_{ij}^a} = -\eta_{m^a}\frac{1}{2}\frac{\partial(d(k)-y(k))^2}{\partial \lambda_{ij}^a}\cdot\frac{\partial \lambda_{ij}^a}{\partial m_{ij}^a}\\ =&\eta_{m^a}(d(k)-y(k))\cdot v_{ij}\cdot\exp(-\frac{(x-m_{ij}^a)^2}{2{\sigma_{ij}^a}^2})\cdot\frac{x-m_{ij}^a}{{\sigma_{ij}^a}^2}\end{aligned} \tag{13}$$

where η_{m^a} is the learning rate. The mean of Gaussian function in the sensory network can be updated according to

$$m_{ij}^a(k+1) = m_{ij}^a(k) + \Delta m_{ij}^a \tag{14}$$

The variance of the Gaussian function can be modified according to a similar regulation which can be expressed as:

$$\begin{aligned}\Delta \sigma_{ij}^a =& -\eta_{\sigma^a}\frac{\partial E}{\partial \sigma_{ij}^a} = -\eta_{\sigma^a}\frac{\partial E}{\partial \lambda_{ij}^a}\cdot\frac{\partial \lambda_{ij}^a}{\partial \sigma_{ij}^a} = -\eta_{\sigma^a}\frac{1}{2}\frac{\partial(d(k)-y(k))^2}{\partial \lambda_{ij}^a}\cdot\frac{\partial \lambda_{ij}^a}{\partial \sigma_{ij}^a}\\ =&\eta_{\sigma^a}(d(k)-y(k))\cdot v_{ij}\cdot\exp(-\frac{(x-m_{ij}^a)^2}{2{\sigma_{ij}^a}^2})\cdot\frac{(x-m_{ij}^a)^2}{{\sigma_{ij}^a}^3}\end{aligned} \tag{15}$$

$$\sigma_{ij}^a(k+1) = \sigma_{ij}^a(k) + \Delta \sigma_{ij}^a \tag{16}$$

where η_{σ^a} is the learning rate. The parameter updating regulation for $m_{(ij)}^o$ and σ_{ij}^o in the emotional network can be derived by the same algorithm.

3.2 Convergence Analysis

In the parameter learning laws in (13) and (15), a proper learning rate η is necessary. For a small value of η, the learning speed can be very slow and if η is too large, the convergence of the system may not be guaranteed. Therefore, in order to train FBELN effectively, an appropriate learning rate should be determined. In the following, the range of learning rate is derived by using Lyapunov stability theorem.

First, the Lyapunov function is defined as:

$$V(k) = \frac{1}{2}e^2(k) \tag{17}$$

Then, the change of $V(k)$ can be described as:

$$\varDelta V(k) = V(k+1) - V(k) = \frac{1}{2}[e^2(k+1) - e^2(k)] \tag{18}$$

the error can be expressed as:

$$e(k+1) = e(k) + \varDelta e(k) = e(k) + [\frac{\partial e(k)}{\partial X}]\varDelta X \tag{19}$$

where X denotes the parameters in the Gaussian function which can be m_{ij} or σ_{ij} and $\varDelta X$ is the change of X. Using (8), it is obtained

$$\frac{\partial e(k)}{\partial X} = \frac{\partial e(k)}{\partial \lambda_{ij}} \cdot \frac{\partial \lambda_{ij}}{\partial X} = -v_{ij} \cdot P(X) \tag{20}$$

where $P(x)$ represents the derivative of the Gaussian function with respect to X. Thus, using (13) and (15) we can get

$$e(k+1) = e(k) - \eta v_{ij}^2 P^2(X) e(k) = e(k)[1 - \eta v_{ij}^2 P^2(X)] \tag{21}$$

Then, $\varDelta V(k)$ can be represented as

$$\varDelta V(k) = \frac{1}{2}e^2(k)[-2\eta v_{ij}^2 P^2(X) + (v_{ij}^2 P^2(X))^2] \tag{22}$$

If η is chosen as

$$0 < \eta < \frac{2}{v_{ij}^2 P^2(X)} \tag{23}$$

$\varDelta V(k)$ in (18) will be smaller than 0. Thus, the convergence is guaranteed by the Lyapunov stability theorem. The error between the desired output and the output of FBELN filter will converge to 0 as $k \rightarrow \infty$.

4 Simulation

The proposed FBELN filter is applied as an adaptive noise canceler in two examples, where the source signal is sinusoidal signal and a real human's speech signal, respectively. In the following, the simulations of different examples will be discussed. Furthermore, an adaptive fuzzy cerebellar model articulation controller (FCMAC) filter is added for comparison [20]. In these simulations, the values of the weights in the FBLEN are generated randomly. The block diagram of an ANC is shown in Fig. 3 [7,21].

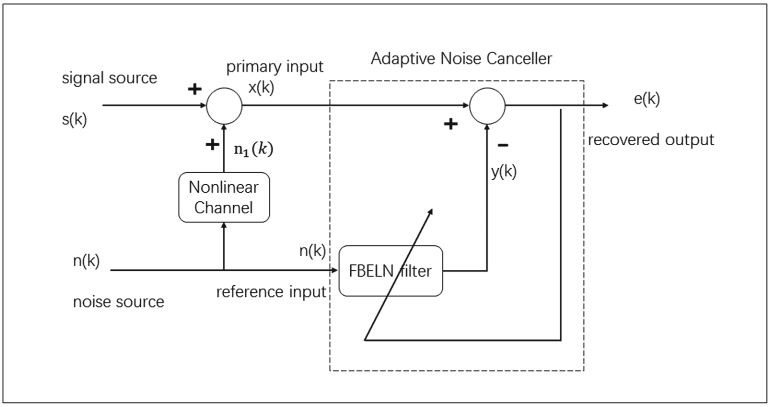


Fig. 3. Block diagram of FBELN adaptive filter

The primary input to the filter $x(k)$ is a combination of the signal source $s(k)$ and a corrupting noise $n_1(k)$ which is generated from the noise source $n(k)$. The ANC is adapted to reconstruct the desired signal $s(k)$ from the corrupted signal $x(k)$. The received signal can be expressed as:

$$x(k) = s(k) + n_1(k) \tag{24}$$

The noise signal $n(k)$ is used as the reference input of the filter. The filter can be regarded as a function which describes the relation between the noise source $n(k)$ and the corrupting signal $n_1(k)$, and then the output of the filter $y(k)$ is subtracted from $x(k)$ to estimate the desired signal $s(k)$. If the filter can mimic the corrupting signal $n_1(k)$ well enough, then the recovered signal $e(k)$ will be as close to $s(k)$ as possible. Assume the source signal $s(k)$, the primary noise $n_1(k)$, the reference input $n(k)$, and the primary noise estimate $y(k)$ are statistically stationary and have zero mean. The recovered output can be derived as

$$e(k) = x(k) - y(k) = s(k) + n_1(k) - y(k) \tag{25}$$

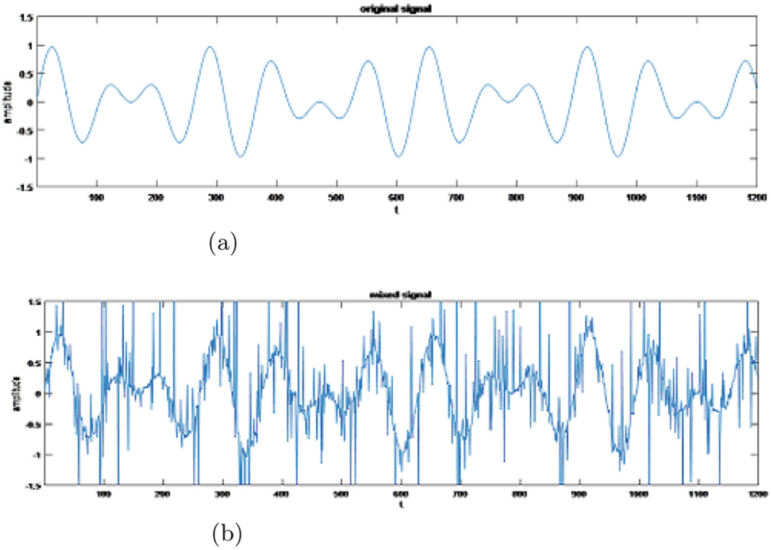
Taking the square and expectation of both sides yields (assume $s(k)$ is independent from $n_1(k)$ and $y(k)$)

$$E[e^2(k)] = E[s^2(k)] + E[(n_1(k) - y(k))^2] \tag{26}$$

In ANC applications, the purpose is to minimize $E[(n_1(k) - y(k))^2]$, which is equivalent to minimize $E[e^2(k)]$ from (26). When $E[(n_1(k) - y(k))^2]$ approximates to zero, the recovered signal $e(k)$ is in fact the desired signal $s(k)$.

Example 1: In this case, consider the source signal $s(k) = sin(0.06k)cos(0.01k)$ shown in Fig. 4(a), and the noise source is a white noise signal. Assume the noise source $n(k)$ passes through a nonlinear channel with a nonlinear function and then generate the corrupting noise $n_1(k)$ which can be described as follows [7]:

$$n_1(k) = 0.06(n(k))^3 \tag{27}$$

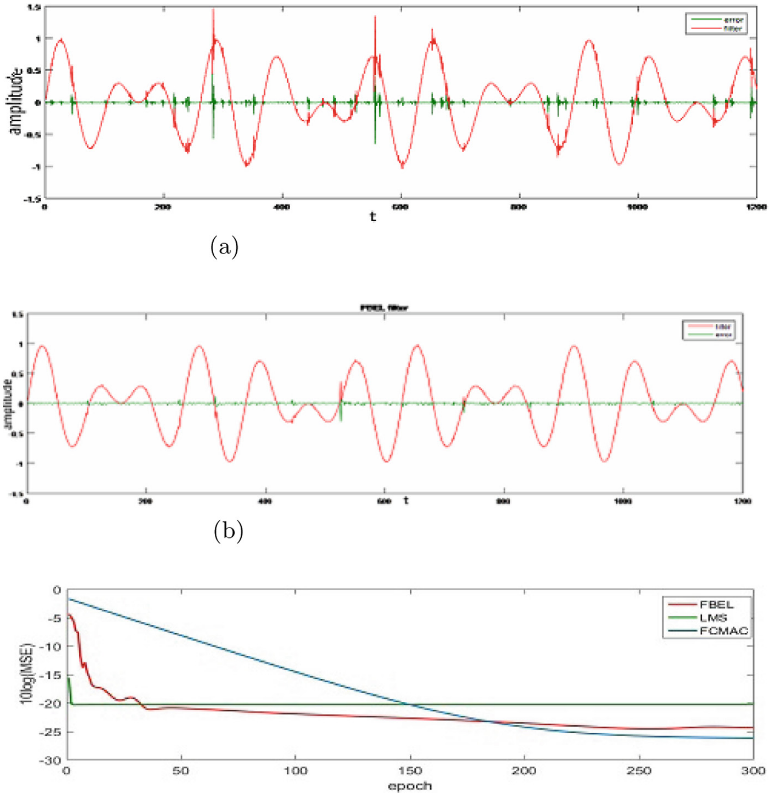


(a)

(b)

Fig. 4. (a) Original signal (b) corrupted signal with 0.6 dBW of Gaussain white noise

The mixed signal is shown in Fig. 4(b). The input of FBELN filter is the source noise $n(k)$ with 1200 training samples in total and the parameters of FBELN are characterized as follows. The structure of the FBELN is chosen as $n = 1$ and $m = 8$ The initial values of the parameters in the Gaussian function are set to be $\mathbf{m} = [1.9\ 2.2\ 2.5\ 2.8\ 3.1\ 3.4\ 3.7\ 4.0]$, $\boldsymbol{\sigma} = [8\ 8\ 8\ 8\ 8\ 8\ 8\ 8]$ and the learning rate of them are chosen as $\eta_m = 0.5$, $\eta_\sigma = 0.08$, respectively. The gains in the emotional signal are set to be $b = c = 0.8$ and the weights of the two networks in FBELN are randomly generated. The simulation results of FCMAC filter and FBELN filter with 0.6 dBW of Gaussain white noise added is shown in Fig. 5(a) and (b), respectively. Figure 5(c) shows the MSE (mean square error) of the three filters during convergence, the FBELN filter has lower MSE than the other filter. Therefore, it is indicated that the proposed method can achieve better performance than LMS filter and the FCMAC filter for noise cancelation. This is due to the special structure of FBELN with two networks infecting each other. The amygdala can only learn and the orbitofrontal cortex can both inhibit or excite depending on the previous knowledge it learned.



(a)

(b)

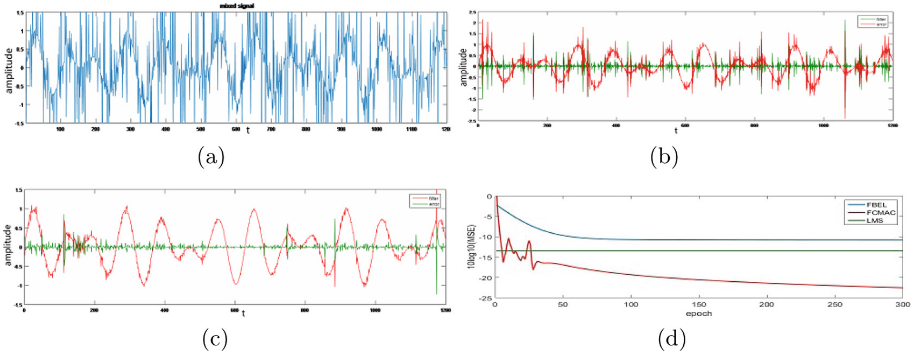
(c)

Fig. 5. (a) Recovered signal and error signal using FCMAC filter with 0.6 dBW of Gaussian white noise (b) recovered signal and error signal using FBELN filter with 0.6 dBW of Gaussian white noise (c) MSE of FCMAC (blue line), LMS filter (green line) and FBELN filter (red line)

To make the performance comparison more precise, simulations with different intensity of noises are added. Figures 6 and 7 show the results with 0.8 dBW and 1.0 dBW of Gaussian white noise, respectively.

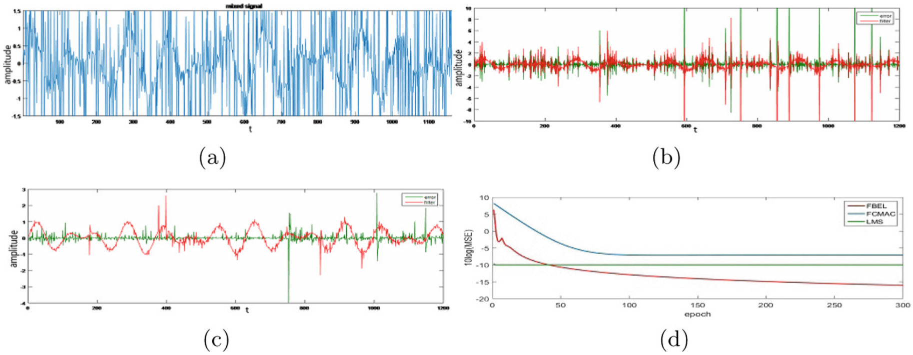
It is shown in Figs. 6 and 7, when the intensity of noise is larger, it is more difficult to reconstruct the source signal from the corrupted signal. The FCMAC filter cannot perform well to separate the source signal, When the noise is added to 1.0 dBW, the LMS filter and the FCMAC filter cannot converge to a low MSE even it trains for many epoches while the proposed filter can still converge to a small value of MSE when the noise is large.

Example 2: In the previous example, the source signal used is a simple sinusoidal signal. However, the source signal is usually very complex in real application. To demonstrate the performance of FBELN filter in real application, a real human's speech signal is used as source signal which is shown in Fig. 8(a). Consider the



(a) (b) (c) (d)

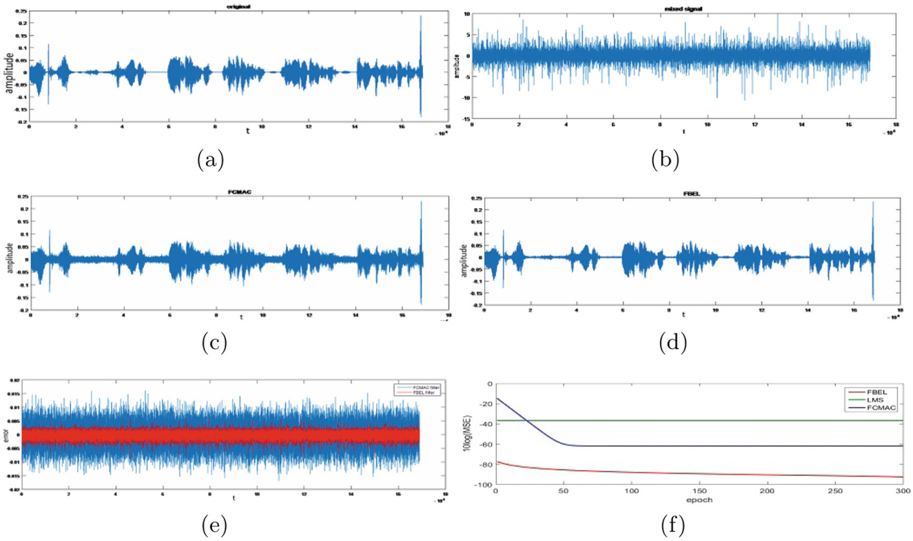
Fig. 6. (a) Corrupted signal with 0.8 dBW of Gaussian white noise (b) recovered signal (red line) and error signal (green line) using FCMAC filter with 0.8 dBW of Gaussian white noise (c) recovered signal (red line) and error signal (green line) using FBELN filter with 0.8 dBW of Gaussian white noise (d) MSE of FCMAC filter (blue line), LMS filter (green line) and FBELN filter (red line)



(a) (b) (c) (d)

Fig. 7. (a) Corrupted signal with 1.0 dBW of Gaussian white noise (b) recovered signal (red line) and error signal (green line) using FCMAC filter with 1.0 dBW of Gaussian white noise (c) recovered signal (red line) and error signal (green line) using FBELN filter with 1.0 dBW of Gaussian white noise (d) MSE of FCMAC filter (blue line), LMS filter (green line) and FBELN filter (red line)

channel function in (27) and the intensity of noise is 0.6 dBW, the corrupted signal is shown in Fig. 8(b). The parameter $\mathbf{m}$ in the Gaussian function is set to be the same as in Example 1 while $\boldsymbol{\sigma}$ is set to be [10 10 10 10 10 10 10 10 10] and the learning rate of them are chosen as $\eta_m = 0.05$, $\eta_\sigma = 0.08$, respectively. The gains in the emotional signal are set to be $b = c = 0.5$ and the weights of the two networks in FBELN are randomly generated. After training for 300 epoches, the simulation results are shown in Fig. 8(c), (f). Apparently, the error signal between the source signal and the recovered signal of FBELN filter is much smaller than FCMAC filter and the value of MSE is much smaller than



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 8. (a) Source signal (b) corrupted signal (c) recovered signal using FCMAC filter (d) recovered signal using FBELN filter (e) error signal of FCMAC filter (blue line) and FBELN filter (red line) (f) MSE of FCMAC (blue line), LMS filter (green line) and FBELN filter (red line)

that of LMS filter and FCMAC filter after convergence. Thus, the proposed filter can achieve better performance when a real speech signal is adapted as source signal.

5 Conclusion

In this paper, an FBELN is proposed for noise cancelation. The fuzzy inference can improve the ability to handle nonlinear problems and the two networks in brain emotional learning effecting each other can increase the robustness of the whole network. The simulation results show that the performance of the new network is better than LMS filter and FCMAC filter. The conditions of the adaptive learning-rates of FBELN are derived based on the Lyapunov function, so the convergence of the filters output error can be guaranteed. For future study, we may aim to include deep learning in our research to solve more complex problem.

References

1. Widrow, B., Stearns, S.D.: Adaptive Signal Processing. China Machine Press, Beijing (2008)

2. Solo, V., Kong, X.: Adaptive signal processing algorithms: stability and performance. Electron. Lett. **25**(6), 414–415 (1995)
3. Manimozhi, M., Snigdha, G., Nagalakshmi, S., Saravana Kumar, R.: State estimation and sensor bias detection using adaptive linear kalman filter. Int. Rev. Model. Simul. **6**(3), 1005–1010 (2013)
4. Pitas, I., Venetsanopoulos, A.N.: Nonlinear Digital Filters. Kluwer Academic Publishers, Dordrecht (1990)
5. Daum, F.: Nonlinear filters: beyond the kalman filter. Aerosp. Electron. Syst. Mag. IEEE **20**(8), 57–69 (2005)
6. Welch, G., Bishop, G.: An Introduction to the Kalman Filter. University of North Carolina at Chapel Hill, Chapel Hill (1995)
7. Lin, C.T., Juang, C.F.: An adaptive neural fuzzy filter and its applications. IEEE Trans. Syst. Man Cybern. Part B **27**(4), 635–656 (1997)
8. Lin, C.M., Chen, L.Y., Yeung, D.S.: Adaptive filter design using recurrent cerebellar model articulation controller. IEEE Trans. Neural Netw. **21**(7), 1149–1157 (2010)
9. Lin, C.M., Li, H.Y.: A novel adaptive wavelet fuzzy cerebellar model articulation control system design for voice coil motors. IEEE Trans. Ind. Electron. **59**(4), 2024–2033 (2012)
10. Lee, C.H., Chang, F.Y., Lin, C.M.: An efficient interval type-2 fuzzy CMAC for chaos time-series prediction and synchronization. IEEE Trans. Cybern. **44**(3), 329–341 (2014)
11. Lin, C.M., Hou, Y.L., Chen, T.Y., Chen, K.H.: Breast nodules computer-aided diagnostic system design using fuzzy cerebellar model neural networks. IEEE Trans. Fuzzy Syst. **22**(3), 693–699 (2014)
12. Lin, C.M., Lin, M.H., Yeh, R.G.: Synchronization of unified chaotic system via adaptive wavelet cerebellar model articulation controller. Neural Comput. Appl. **23**(3), 965–973 (2013)
13. Lucas, C., Shahmirzadi, D., Sheikholeslami, N.: Introducing BELBIC: brain emotional learning based intelligent controller. Intell. Autom. Soft Comput. **10**(1), 11–21 (2004)
14. Balkenius, C., Jan, M.: Emotional learning: a computational model of the amygdala. Cybern. Syst. **32**(6), 611–636 (2010)
15. Dehkordi, B.M., Parsapoor, A., Moallem, M., Lucas, C.: Sensorless speed control of switched reluctance motor using brain emotional learning based intelligent controller. Energy Convers. Manag. **52**(1), 85–96 (2011)
16. Sharbafi, M.A., Lucas, C., Daneshvar, R.: Motion control of omni-directional three-wheel robots by brain-emotional-learning-based intelligent controller. IEEE Trans. Syst. Man Cybern. Part C **40**(6), 630–638 (2010)
17. Lin, C.M., Chung, C.C.: Fuzzy brain emotional learning control system design for nonlinear systems. Int. J. Fuzzy Syst. **17**(2), 117–128 (2015)
18. Lotfi, E., Akbarzadeh-T, M.R.: Supervised brain emotional learning. In: International Joint Conference on Neural Networks, pp. 1–6 (2012)
19. Jouffe, L.: Fuzzy inference system learning by reinforcement methods. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **28**(3), 338–355 (1998)
20. Lin, C., Li, H.: Dynamic petri fuzzy cerebellar model articulation controller design for a magnetic levitation system and a two-axis linear piezoelectric ceramic motor drive system. IEEE Trans. Control Syst. Technol. **23**(2), 693–699 (2015)
21. Widrow, B., Glover, J.R., Mccool, J.M., Kaunitz, J., Williams, C.S., Hearn, R.H., Zeidler, J.R., Dong, J.R.E., Goodlin, R.C.: Adaptive noise cancelling: principles and applications. Proc. IEEE **63**(12), 1692–1716 (1975)

Artificial Neural Network Analysis of Volatile Organic Compounds for the Detection of Lung Cancer

John B. Butcher[1](✉), Abigail V. Rutter[2], Adam J. Wootton[1,3], Charles R. Day[1], and Josep Sulé-Suso[2,4]

[1] School of Computing and Mathematics, Keele University,
Staffordshire ST5 5BG, UK
j.b.butcher@keele.ac.uk

[2] Institute for Science & Technology in Medicine, Guy Hilton Research Centre,
Keele University, Staffordshire ST4 7QB, UK

[3] Foundation Year Centre, Keele University, Staffordshire ST5 5BG, UK

[4] Oncology Department, Royal Stoke University Hospital,
University Hospitals of North Midlands,
Newcastle Road, Staffordshire ST4 6QG, UK

Abstract. Lung cancer is a widespread disease and it is well understood that systematic, non-invasive and early detection of this progressive and life-threatening disorder is of vital importance for patient outcomes. In this work we present a convergence of familiar and less familiar artificial neural network techniques to help address this task. Our preliminary results demonstrate that improved, automated, early diagnosis of lung cancer based on the classification of volatile organic compounds detected in the exhaled gases of patients seems possible. Under strictly controlled conditions, using Selected Ion Flow Tube Mass Spectrometry (SIFT-MS), the naturally occurring concentrations of a range of volatile organic compounds in the exhaled gases of 20 lung cancer patients and 20 healthy individuals provided the dataset that has been analysed. We investigated the performance of several artificial neural network architectures, each with complementary pattern recognition properties, from the domains of supervised, unsupervised and recurrent neural networks. The neural networks were trained on a subset of the data, with their performance evaluated using unseen test data and classification accuracies ranging from 56% to 74% were obtained. In addition, there is promise that the topological ordering properties of the unsupervised networks' clusters will be able to provide further diagnostic insights, for example into patients who may have been heavy smokers but so far have not presented with any lung cancer. With the collection of data from a larger number of subjects across a long time period there is promise that an automated assistive tool in the diagnosis of lung cancer via breath analysis could soon be possible.

Keywords: Lung cancer diagnosis · Volatile organic compounds · SIFT · Artificial neural network analysis

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_15

1 Introduction

Lung cancer is associated with a poor prognosis and survival rates of less than 20% at 5 years [1]. One reason for the poor prognosis is that more than 50% of patients have advanced disease at the time of diagnosis [2]. Therefore, urgent work is needed to be able to quickly and cheaply diagnose this disease at an early stage. Importantly any new diagnostic tools should ideally be non-invasive.

There has been some interest over the last few years in the study of trace volatile organic compounds (VOCs) in exhaled breath. Initially, studies were carried out using gas chromatography. However, this is a laborious procedure and cannot easily provide information on absolute concentrations of these compounds. Over the last few years, new techniques have been developed to study trace gases in breath. These include the Selected Ion Flow Tube Mass Spectrometry (SIFT-MS), Proton Transfer Reaction Time-of-Flight Mass Spectrometry, Laser Spectroscopy and Quantum Cascade Laser-based gas sensors amongst others [3]. However, while several studies have linked some VOCs present in exhaled gases of patients with lung cancer (e.g. [4]), this methodology has not yet been applied in clinical practice.

Artificial neural networks are one of the most widely used computational intelligence approaches when tasked with the analysis and investigation of complex and noisy data with the added complication of potential non-linear interactions between subsets of features within the dataset. In this work, neural networks from three broad domains of machine learning have been deployed: the supervised backpropagation multilayer perceptron (MLP); Kohonen's unsupervised self-organising map (SOM) to carry out some autonomous clustering of the data and expose interrelationships that might be present in the SIFT-MS data; and from the domain of recurrent neural networks, more usually applied to time-series data processing, a 'clamped' variant of the echo state network (clamped-ESN) architecture.

The rest of the paper is organized as follows: Sect. 2 outlines the methodology used to collect the data from the patients and control group, in addition to the data pre-processing approaches and artificial neural network architectures, including their configurations, that were used; Sect. 3 presents the results of applying the ANNs to this dataset; Sect. 4 discusses these results in further detail; and finally, Sect. 5 concludes the paper and outlines future work to be conducted.

2 Methods

Patients with the histological diagnosis of lung cancer treated at the Oncology Department, Royal Stoke University Hospital were included in this study. Local ethical and R&D approval was obtained. Twenty patients with the histological diagnosis of lung cancer and 20 control cases with either basal cell carcinoma (BCC) or squamous cell carcinoma (SCC) of the skin, but with no diagnosis of lung cancer, attending the Oncology department for treatment were also included. The two types of skin tumors don't usually spread and tend to remain localized to the skin. Furthermore, they also present in an elderly population, as is the case for lung cancer, making them a good case control for this study. Data on smoking patterns, alcohol consumption, and the presence of chest infection were also recorded. In order to avoid contamination by other metabolites in their breath, both patients and control subjects with diabetes, renal failure or other metabolic diseases were excluded from the study. Patients (prior to starting treatment) and the control subjects all gave a morning breath sample. They did not eat food or drink alcohol nor smoke for the previous 12 h and were not allowed to clean their teeth. They rinsed their mouth with water prior to giving a breath sample. They were asked to take a deep inhalation and exhale into a specially designed bag. As soon as the patient had finished the exhalation, the bag was sealed and taken for analysis to the laboratory facilities at the Institute of Science and Technology in Medicine, Guy Hilton Research Centre. The bags were kept at 37 °C and the breath gas from the bag was passed directly into the SIFT-MS instrument. Using this method, partial pressures of the trace gases down to about 10 parts per billion can be measured. Furthermore, the time response of the instrument is 20 ms, allowing the time profiles of the trace gas concentration in the breath to be obtained during a normal breathing cycle (see [5] for more details of the SIFT-MS). However, breath can contain 850 different VOCs, with the precise significance and origin of many of them only poorly understood [6]. Thus, not only is more knowledge needed on the chemical pathways followed by VOCs between being released and finally expelled through breath [7], but also, improved data analysis methods and tools such as artificial neural networks (ANNs) to identify important biomarker(s) in this myriad of VOCs.

The analytical mass spectrometer in the SIFT-MS instrument is scanned over a pre-determined mass range of ions (10 to 180 m/z) to determine which breath metabolites are present and at what concentration. These data can readily be obtained within 1 min.

Prior to presenting any data to the neural networks, 15 VOCs were selected from the SIFT-MS data as potential indicators of lung cancer. In no particular order these were acetone, acetaldehyde, ethanol, pentanol, hexanol, butyric acid, pentene, putresciene, terpenes, xylene, propanol, acetic acid, benzene, toluene and butanol. The data was then normalized between the range −1 and +1. The relatively modest number of subjects, just forty in total, recorded in the dataset meant that measures to try to avoid overfitting any very small training datasets had to be taken. Accordingly, the available dataset was subjected to a randomized, five-fold cross-validation decomposition that yielded five pairs of training and testing datasets. Each fold delivered 32 subjects and 8 subjects to be used as disjoint sets of training and testing data, respectively.

Artificial neural networks are one of the most widely used computational intelligence approaches for the analysis of complex and noisy data. One of the most established supervised neural network architectures is the multi-layer perceptron (MLP) using the backpropagation of error learning algorithm [8]. The MLP's supervised training process involves evaluating a network's performance by comparing its actual output to the target output associated with an input pattern in order to calculate the error of the network. The error is then backpropagated through the network to change the weights on each connection in order to progressively reduce the network's overall error rate. MLPs have been successfully used in many studies across a wide range of domains for static pattern recognition, hence were considered to be a suitable architecture in this study. Examples in the medical domain include the detection of breast cancer [9], lung and oral cancer [10], early stage lung cancer using chest computed tomography images [11]. More recently, Adetiba et al. [12] demonstrated that an MLP could achieve 96% accuracy, outperforming a support vector machine and a Naïve Bayes classifier, when classifying VOCs from the Catalogue of Somatic Mutations in Cancer (COSMIC) database which contains lung cancer biomarker genes.

For this study, MLPs with 15 input units and a single output unit were used with high valued activations (i.e. activations > 0.5) at the output unit interpreted as cancer patients and low valued activations (i.e. activations < 0.5) at the output unit assigned to the class of non-cancer patients. The very small SIFT dataset size and the attendant risk of overfitting due to MLP over-parameterization meant that only modestly sized MLP hidden layers were ranged over (i.e. between 2 to 10 hidden units) in order to find the best performing hidden layer size: which was found to be around 6 hidden units. The MLPs were all trained with backpropagation learning rate and momentum values of 0.2 and 0.5, respectively. To avoid overfitting during training MLP performance on the training and the test data was assessed at 200 epoch intervals and training regimes lasting around 4,000 epochs were found to work best overall.

Echo state networks (ESNs) are a recurrent neural network (RNN) and have traditionally been applied to time-series data processing [13]. ESNs are part of the reservoir computing (RC) family of RNNs and rather than modifying all weighted connections in the network, only the ESN's reservoir to output layer weights are modified during the training process. This reduces the complexity of training, making RC approaches well suited to time-series data analysis where excellent performance has been obtained [14–17]. However, *clamped-ESNs* have also recently been shown to be well-suited to static pattern recognition tasks (i.e. involving non-time-series data, e.g. [18, 19]). For a clamped-ESN approach, each input pattern is clamped to the input units for a succession of presentations until the output node(s) settle to a stable state. Unlike conventional static pattern recognizers (such as MLPs that produce exactly the same response when presented with a static input pattern regardless of how many times the pattern is presented and regardless of what might have been presented in the recent past) the recurrent clamped-ESN, after its initial response, follows a trajectory towards an attractor in the high dimensional reservoir space that might not have been reached by a single static pattern presentation. This clamped-recurrent processing allows the weights connecting the ESN's reservoir units to the output units to better classify an input pattern [18, 19].

The ESNs used in this work consisted of 15 input units, 55 reservoir units and one output unit. The reservoir neurons had a 90% connectivity factor, spectral radius of 0.181, leak rate of 0.3820 and a *tanh* activation function, while the input scaling was set to 1. Each data sample was presented to the network 24 times before the output activation was harvested. The network was trained to give an output of +1 when presented with data from a patient with cancer, and −1 when presented with data from a patient without cancer. 20 different ESNs were trained on each set of training samples and tested on the corresponding set of testing samples, giving a total of 100 different trained ESNs. This was done to account for the variability of an ESN's weights that are generated randomly at initialization. After the testing data were presented, a final output value greater than zero was taken as a positive (i.e. a patient with cancer), while any output value less than zero was taken as a negative (i.e. a patient without cancer).

3 Classification Results

The lung cancer patients and controls both consisted of 12 males and 8 females. Regarding age, the control cases included a more elderly population. 55% of patients presented with metastatic disease (stage IV) while the remaining patients presented with locally advanced disease. Finally, there were more smokers and ex-smokers in the lung cancer patients group.

The performance of each ANN architecture was evaluated over each of the 5 folds used for cross validation. In addition, an overall average performance across all of the folds was determined and all of the results are shown in Table 1, below.

Table 1. The test performance of each architecture for each fold of cross validation and the overall average performance (standard deviation shown in brackets)

	Cross validation performance (% correct) for each fold					Overall (% correct)
Architecture	1	2	3	4	5	
MLP	75 (0)	88 (0)	75 (0)	69 (6.9282)	63 (14.4337)	73.9 (3.202)
Clamped-ESN	40 (13.204)	73.125 (4.5793)	51.25 (10.653)	50.625 (2.7951)	65 (6.5394)	56 (14.3768)

As shown in Table 1, the MLP offers the best performance with 73.9%. Table 2 shows sensitivity analysis of the performance of the two architectures detailing the true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) averaged across all 5 folds. The Matthews correlation coefficient is also shown where a score of +1 indicates perfect classification, a score of 0 random classification, and −1 a complete disagreement between the ANN output and the ground truth.

Table 2. Sensitivity analysis of the two architectures showing the true positives (TP), false positives (FP), true negatives (TN), false negatives (FN) scores, as well as their Mathews correlation coefficient (MCC) (standard deviation shown in brackets).

Architecture	TP	FP	TN	FN	MCC
MLP	3.05 (1.8235)	1.15 (0.8587)	2.85 (1.4534)	0.95 (1.3038)	0.6397
Clamped-ESN	2.22 (0.2727)	1.74 (0.4491)	2.26 (0.4491)	1.78 (0.2727)	0.2018

Inspection of Table 2 shows that the MLP has more true positives and true negatives, and low false positives and false negatives when compared to the clamped-ESN. A higher MCC score further corroborates the superior performance of the MLP.

4 Discussion

One well known risk factor associated with lung cancer is smoking. In this study, 4 patients were still active smokers over period when the breath sample was taken and 16 were ex-smokers (stopped smoking at least 1 month before obtaining a breath sample). For the 20 control cases, 10 subjects were ex-smokers and the remaining 10 had not smoked before.

Early investigations using the unsupervised clustering properties of SOM neural networks suggest that one of the source of error in the, already very good classification scores, is the extent to which a non-lung-cancer patient has previously smoked. An important VOC constituent of cigarette smoke [20] and possible marker for active/recent smoking is acetonitrile [21] and so far, this has not been used as one of the inputs to the classification process reported above.

Even if the inclusion of additional VOCs such as acetonitrile do nothing to improve the overall classification accuracies, the two-dimensional topological ordering of clusters of patients on the trained SOM's detector grid provides an excellent two-dimensional data visualization tool that will be useful in a clinical setting. This applies in particular when difficult decisions need to be made regarding high-dimensional breath-samples that might be at the margins of the classifications otherwise determined by the MLPs and clamped-ESNs, making the SOM a well-suited architecture when a breath sample falls between two or more classes.

The inferior performance provided by the clamped-ESN is surprising given previous performance on other static datasets. Further investigations are required to ascertain why this is the case, although the small size of the dataset analysed in this study could be a factor.

5 Conclusion

The early detection of lung cancer is of vital importance in the treatment and prognosis of the disease. Here we have presented preliminary results detailing the automated analysis of VOCs found in exhaled breath samples using several artificial neural networks. Despite the small size of the dataset (n = 40), MLPs and clamped-ESNs were able to offer very good test classification performance: the MLP achieving an overall test average of 74%. Future work will involve further analysis of the characteristics of the whole VOC dataset and the collection of more SIFT data from a larger, longitudinal study, with the overall aim of creating an automated tool to assist in the early detection of lung cancer. The use of other architectures that have shown promise in this and wider domains (e.g. deep belief networks [22]) could also be the focus of future work.

Smoking is also a major risk factor for most chronic lung and pleural diseases. Confirming a patient's smoking status has many treatment implications, for example, in cases where a potential lung transplant is required it is prudent that patients should have stopped smoking for at least 6 months. Smoking status is presently assessed by measuring levels of cotinine, the main breakdown product of nicotine, in the patient's urine. Therefore, SIFT-MS in combination with the supervised and unsupervised neural network techniques used here hold the prospect of cheaper, on-line, identification of patient compliance and smoking abstinence regimes as well being able to make accurate lung cancer determinations.

Acknowledgments. We gratefully acknowledge funding from the Slater & Gordon Health Projects & Research Fund/14/15 Round 1/A34489.

References

1. Lewis, D.R., Chen, H.S., Feurer, E.J., et al.: SEER Cancer Statistics Review, 1975–2008. MD National Cancer Institute, Bethesda (2010)
2. Comella, P., Frasci, G., Panza, N., et al.: Randomised trial comparing cisplatin, gemcitabine and vinorelbine with either cisplatin and gemcitabine or cisplatin and vinorelbine in advanced non small cell lung cancer: interim analysis of a Phase III trial of the Southern Italy Cooperative Oncology Group. J. Clin. Oncol. **18**, 1451–1457 (2000)
3. Amann, A., Smith, D. (eds.): Breath analysis for clinical diagnosis and therapeutic monitoring. World Scientific Publishing Co., Singapore (2005)
4. Buszewski, B., Ulanowska, A., Kowalkowski, T., Cieliski, K.: Investigation of lung cancer biomarkers by hyphenated separation techniques and chemometrics. Clin. Chem. Lab. Med. **50**, 573–581 (2012)
5. Rutter, A.V., Chippendale, T.W.E., Yang, Y., Španěl, P., Smith, D., Sulé-Suso, J.: Quantification by SIFT-MS of acetaldehyde released by lung cells in a 3D model. Anal. **138**, 91–95 (2013)
6. Amann, A., de Lacy Costello, B., Miekisch, W., Schubert, J., Buszewski, B., Pleil, J., Risby, T.: The human volatilome: volatile organic compounds (VOCs) in exhaled breath, skin emanations, urine, feces and saliva. J. Breath Res. **8**(3), 034001 (2014)
7. Hakim, M., Broza, Y.Y., Barash, O., et al.: Volatile organic compounds of lung cancer and possible biochemical pathways. Chem Rev. **112**, 5949–5966 (2012)
8. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Internal Representations by Error Propagation Parallel Distributed Processing, Explorations in the Microstructure of Cognition, vol. 1. MIT Press, Cambridge (1986)
9. Delen, D., Walker, G., Kadam, A.: Predicting breast cancer survivability: a comparison of three data mining methods. Artif. Intell. Med. **34**(2), 113–127 (2005)
10. Choudhury, T., Kumar, V., Nigam, D., Mandal, B.: Intelligent classification of lung & oral cancer through diverse data mining algorithms. In: 2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE), pp. 133–138. IEEE (2016)
11. Emaminejad, N., Qian, W., Guan, Y., Tan, M., Qiu, Y., Liu, H., Zheng, B.: Fusion of quantitative image and genomic biomarkers to improve prognosis assessment of early stage lung cancer patients. IEEE Trans. Biomed. Eng. **63**(5), 1034–1043 (2016)
12. Adetiba, E., Adebiyi, M.O., Thakur, S.: Breathogenomics: a computational architecture for screening, early diagnosis and genotyping of lung cancer. In: Rojas, I., Ortuño, F. (eds) Bioinformatics and Biomedical Engineering, IWBBIO 2017. Lecture Notes in Computer Science, vol. 10209. Springer, Cham (2017)
13. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. Bonn, Germany: German National Research Center for Information Technology GMD Technical report, vol. 148(34), p. 13 (2001)

14. Butcher, J.B., Day, C.R., Austin, J.C., Haycock, P.W., Verstraeten, D., Schrauwen, B.: Defect detection in reinforced concrete using random neural architectures. Comput. Aided Civ. Infrastruct. Eng. **29**(3), 191–207 (2013)
15. Verstraeten, D., Schrauwen, B., Stroobandt, D.: Reservoir-based techniques for speech recognition. In: IEEE International Joint Conference on Neural Networks (IJCNN 2006), pp. 1050–1053 (2006)
16. Butcher, J.B., Verstraeten, D., Schrauwen, B., Day, C.R., Haycock, P.W.: Reservoir computing and extreme learning machines for non-linear time-series data analysis. Neural Netw. **38**, 76–89 (2013)
17. Scardapane, S., Butcher, J.B., Bianchi, F.M., Malik, Z.K.: Advances in biologically inspired reservoir computing. Cogn. Comput. **9**(3), 295–296 (2017)
18. Wootton, A.J., Taylor, S.L., Day, C.R., Haycock, P.W.: Optimizing echo state networks for static pattern recognition. Cogn. Comput. **9**(3), 391–399 (2017)
19. Emmerich, C., Reinhart, R.F., Steil, J.J.: Recurrence enhances the spatial encoding of static inputs in reservoir networks. In: Diamantaras, K., Duch, W., Iliadis, L. (eds) Artificial Neural Networks - ICANN 2010, vol. 6353, pp. 148–153. Springer, Heidelberg (2010)
20. Campbell, J.K., Rhoades, J.W., Gross, A.L.: Acetonitrile as a constituent of cigarette smoke. Nature **198**, 991–992 (1963)
21. Kushch, I., et al.: Compounds enhanced in a mass spectrometric profile of smokers' exhaled breath versus non-smokers as determined in a pilot study using PTR-MS. J. Breath Res. **2**(2), 026002 (2008)
22. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**, 1527–1554 (2006)

Predicting the Occurrence of World News Events Using Recurrent Neural Networks and Auto-Regressive Moving Average Models

Emmanuel M. Smith[1](✉), Jim Smith[1], Phil Legg[1], and Simon Francis[2]

[1] University of the West of England, Bristol, UK
{sinclair.smith,james.smith,phil.legg}@uwe.ac.uk
[2] Montvieux Limited, Gloucestershire, UK
simon.francis@montvieux.com

Abstract. The ability to predict future states is fundamental for a wide variety of applications, from weather forecasting to stock market analysis. Understanding the related data attributes that can influence changes in time series is a challenging task that is critical for making accurate predictions. One particular application of key interest is understanding the factors that relate to the occurrence of global activities from online world news reports. Being able to understand why particular types of events may occur, such as violence and peace, could play a vital role in better protecting and understanding our global society. In this work, we explore the concept of predicting the occurrence of world news events, making use of Global Database of Events, Language and Tone online news aggregation source. We compare traditional Auto-Regressive Moving Average models with more recent deep learning strategies using Long Short-Term Memory Recurrent Neural Networks. Our results show that the latter are capable of achieving lower error rates. We also discuss how deep learning methods such as Recurrent Neural Networks have the potential for greater capability to incorporate complex associations of data attributes that may impact the occurrence of future events.

1 Introduction

The ability to predict the nature of upcoming events has a variety of potential applications for protecting and understanding our global society. In recent years, there has been much interest in attempting to predict the occurrence of such events [1,3,14,15,18–20]. One of the most challenging aspects is to characterise what previously-observed data may have a causal relationship on future activity. Currently, many researchers make use of the Conflict and Mediation Event Observations (CAMEO) [5] framework for coding event types, where events of different types and by various global actors (e.g., countries) are mapped as a time series. Existing statistical methods such as the Autoregressive Moving Average (ARMA) have been used previously in an attempt to predict the occurrence of different event types [18,20], as have Machine Learning (ML) algorithms such as Bayesian methods and random forest classifiers [1,14].

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_16

More recently, Recurrent Neural Networks (RNNs) have been used to improve prediction performance on several sequence-based learning problems [11]. In particular, Long Short-Term Memorys (LSTMs) have been used for time series predictions [2,4,13]. Hence, we are interested to study whether the reported benefits of using RNNs in other domains can be applied for predicting global event data. In this paper, we make the following contributions:

1. We present a comprehensive study that compares the use of RNNs with traditional ARMA models, for time series analysis.
2. We apply RNNs and models from the ARMA family to the problem of predicting the occurrence of world news events. We also investigate the suitability of each method for this task and discuss their potential extendability. We show how the RNN approach provides greater flexibility for incorporating additional information that would support its predictive capabilities.

2 Related Work

As global event data has become more accessible, many studies have attempted to make actionable predictions [1,8,14–16,18,19,21,22]. Much of this work has focused on statistical time series techniques, such as Autoregressive Fractionally Integrated Moving Average (ARFIMA) models. For example, Brandt et al. developed Markov switching and Bayesian vector autoregression models for predicting material conflict between Israel and Palestine at inter- and intra-state spacial resolutions [1]. Yonamine employed ARFIMA models to predict material conflict for Afghanistan districts [21]. Yuan forecasted the inter-country relationships of China, again using ARFIMA models [22].

Various ML methods have been applied. Perry [14] reported the use of naive Bayes and random forest classifiers on the Armed Conflict Location and Event Dataset (ACLED) event database [17]. Qiao et al. developed a Hidden Markov Model (HMM) for predicting social unrest in five separate south-east Asian countries using Global Database of Events, Location, and Tone (GDELT) data [16]. Phua et al. utilised decision trees to predict the Singapore stock market's Straits Times Index [15], again using (GDELT) data.

With the exception of HMMs, one limitation is that there is little consideration for the temporal dependence of future events. In other domains such as language modelling and video analysis, LSTMs have been shown to maintain temporality in sequential-based learning tasks [7,12]. Lipton et al. also used LSTMs to perform classification on multivariate time series data, obtained from patients' medical sensor readings [13]. They found that LSTMs perform better than their baseline models, and that heavy use of dropout allows for larger networks which achieve greater results. These clinical events are similar to global events, as they have associated time stamps, actors, and event types. With the recent successes of LSTMs, our study will address how such techniques can be utilised to better characterise and predict the occurrence of global events.

3 Auto-Regressive Moving Average

ARMA models are one of the most commonly used statistical techniques for modeling and predicting time series data [10] by parameterising and combining a number of independent components. The Auto-Regressive (AR) component also uses a weighted linear combination of the past values of the series, essentially performing a regression of the time series against itself. For a target variable x_t,

$$x_t = c + e_t + \sum_{i=1}^{p} \psi_i x_{t-i} \tag{1}$$

where x_t is the value of the time series at time t, c is some constant, e_t is the error at time t which is assumed to be white noise, p is the number of time lags to consider, and ψ are the parameters.

The Moving Average (MA) component also attempts to predict a target variable using a form of regression, although here it is based on the previous forecast errors. Whilst the errors can not be directly observed due to noise, the MA component can be inverted to be in the form of the AR component if the time series is assumed to be stationary. For a target variable x_t,

$$x_t = c + e_t + \sum_{i=1}^{q} \theta_i e_{t-i} \tag{2}$$

where q is the number of time lags, and θ are the parameters. If the assumption made for inverting from AR to MA holds for a given time series, the ARMA is simply the combination of the two components. For a target variable x_t,

$$x_t = c + e_t + \sum_{i=1}^{p} \psi_i x_{t-i} + \sum_{i=1}^{q} \theta_i e_{t-i} \tag{3}$$

One crucial assumption here is that the time series is stationary. Many prediction tasks deal with time series that do not meet this criterion. It is possible to transform a non-stationary time series into a stationary one by taking the difference between each variable and its predecessor. For example, the first order difference x_t^1 of the original variable x_t is $x_t^1 = x_t - x_{t-1}$, and consequently, x_t^2 is the second order difference. As alluded to, allowing the order to be fractional allows for finer granularity in this approach. For a fractional difference x_t^d,

$$x_t^d = \sum_{i=0}^{\infty} \frac{\prod_{j=0}^{i-1}(d-j)}{i!} x_{t-i} \tag{4}$$

where d is the fractional differencing coefficient. Combining the AR and MA components, and incorporating a fractional differencing component gives us the complete ARFIMA model.

4 Long Short-Term Memory

Traditional Artificial Neural Networks (ANNs) are not designed to incorporate temporal dependencies. RNNs extend this, such that each node incorporates a loop back connection which allows a hidden state to be passed through time. At each time step, the node receives its usual inputs, together with the hidden state obtained from the previous time step, allowing for a long term memory to be incorporated. It was later identified that RNNs suffer from the vanishing gradient problem [9]. LSTM, an extension to the RNN architecture, was developed to overcome this issue [7]. This is achieved through gating mechanisms and a cell activation state, in addition to the existing hidden state, since the network learns when to forget long-term information and when to incorporate new information. Separating the hidden state with the cell activation state also allows for the network to learn to control how much of the cell activation it outputs.

As can be seen in Fig. 1, a LSTM node takes, as input, a combination of an input vector, $\mathbf{x}$, and the previous hidden state, $\mathbf{h}$.[1] The LSTM first calculates a new candidate cell activation, $\tilde{\mathbf{c}}$, via a weighted sum of these inputs and a bias, b. The result of this calculation passes through a hyperbolic tangent activation function, as denoted by Eq. 5.

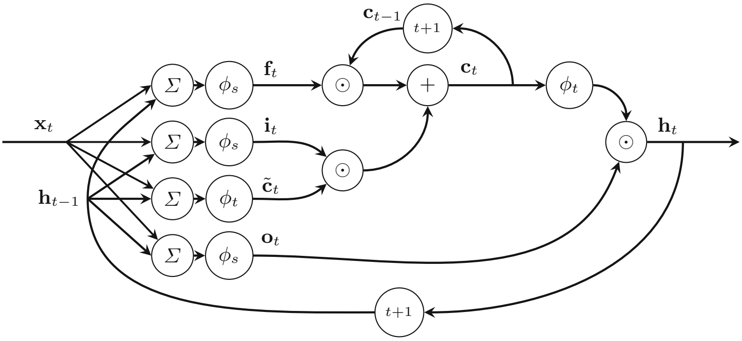


Fig. 1. Structure of a LSTM node, where each arrow represents the movement of vectors, and each circle denotes an operation performed on those vectors. Σ denotes a weighted summation, and $t+1$, denotes a delay of one time step. Input data flows through the node's calculations left-to-right, with the exception of the time delay that is retrieved from the previous time step.

Once the candidate cell activation is determined, the gates control how much of this activation we should keep and how much of cell activation from the previous time step we should forget. The gate that controls how much of the candidate cell activation we should keep is the input gate, $\mathbf{i}$, and the gate that controls how much we forget the past cell activation is the forget gate, $\mathbf{f}$.

[1] Usually the previous cell activation and hidden state are initially set to zero.

The final gate, the output gate, $\mathbf{o}$, is incorporated once the hidden state is calculated from the new cell activation. These values are calculated as follows:

$$\tilde{\mathbf{c}}_t = \phi_t(\mathbf{W}_{\tilde{c}}\mathbf{x}_t + \mathbf{U}_{\tilde{c}}\mathbf{h}_{t-1} + b_{\tilde{c}}) \tag{5}$$

$$\mathbf{f}_t = \phi_s(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + b_f) \tag{6}$$

$$\mathbf{i}_t = \phi_s(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + b_i) \tag{7}$$

$$\mathbf{o}_t = \phi_s(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + b_o) \tag{8}$$

where $\tilde{\mathbf{c}}_t$ is the candidate cell activation; $\{\mathbf{f}, \mathbf{i}, \mathbf{o}\}_t$ are the forget, input, and output gate vectors; $\phi_{\{s,t\}}$ are the sigmoid and hyperbolic tangent activation functions; $\{\mathbf{W}, \mathbf{U}\}_{\{\tilde{\mathbf{c}},\mathbf{f},\mathbf{i},\mathbf{o}\}}$ are weight matrices; $\mathbf{x}_t$ is the input vector; $\mathbf{h}_{t-1}$ is the hidden state vector at the previous time step; and $b_{\{\tilde{c},f,i,o\}}$ are biases.

To calculate the final hidden state vector $\mathbf{h}_t$, we must first compute the new cell activation $\mathbf{c}_t$, as a combination of the candidate cell activation, gated by $\mathbf{i}_t$, and the previous cell activation $\mathbf{c}_{t-1}$, gated by $\mathbf{f}_t$. The gating is performed using element-wise multiplication and the activations are then combined using addition. Finally, the hidden state is the cell activation, bounded by the hyperbolic tangent activation function, and gated by $\mathbf{o_t}$, denoted as follows:

$$\mathbf{c}_t = \mathbf{c}_{t-1} \odot \mathbf{f}_t + \tilde{\mathbf{c}}_t \odot \mathbf{i}_t \tag{9}$$

$$\mathbf{h}_t = \phi_t(\mathbf{c}_t) \odot \mathbf{o}_t \tag{10}$$

where $\odot$ is an operator for the element-wise multiplication of vectors.

5 Methodology

In this section, we present a comparative study of using LSTM and ARFIMA models for predicting the occurrence of material conflict events in Afghanistan. We make use of GDELT, which provides a real-time machine-coded data repository of global news event reports. GDELT provides event details such as the actors involved, the severity which ranges from verbal cooperation to material conflict, and a link to the source. Our study thus extends, via LSTMs, the work of Yonamine [21], who originally considered predicting the number of material conflict events that occur in Afghanistan using ARFIMA models.

5.1 Data Representation

GDELT is an event database which automatically collates global news, and identifies and encodes mentioned events along with their associated details. An event usually involves multiple actors which range from countries and known groups, to prominent figures. In an event, an identifiable action is performed at a particular time and location. GDELT was chosen for this research due to its tremendous temporal and geospatial scale, and for its release interval which allows for real-time, actionable predictions.

There are multiple methods of obtaining the GDELT data, however, for the purposes of this study, the dataset was obtained from their BigQuery API. Our query extracted all GDELT 1.0 events, where any of the actor codes contained the substring ‘AFG’ (Afghanistan), since the year 2000. Even though more historic data is available, we only include data from 2000 onwards in order to cover the time range covered by Yonamine’s study [21], namely 2001 to 2012.

The majority of the event features provided by GDELT are categorical, such as the actor, and thus do not easily lend themselves towards time series representations. This makes them unsuitable for use by both models in their standard form. Techniques such as one-hot encoding or other vector representations could be used, however, that is not the main focus of this investigation. After excluding non-numerical features, each event has a date, an event code, an associated latitude and longitude, a Goldstein scale [6], the number of articles mentioning the event, the number of mentions of the event within articles, the number of article sources, and the average article tone. As in Yonamine study, we focus on predicting solely the total count of material conflict events per month from April 2008 to April 2012, given a training set ranging from February 2001.

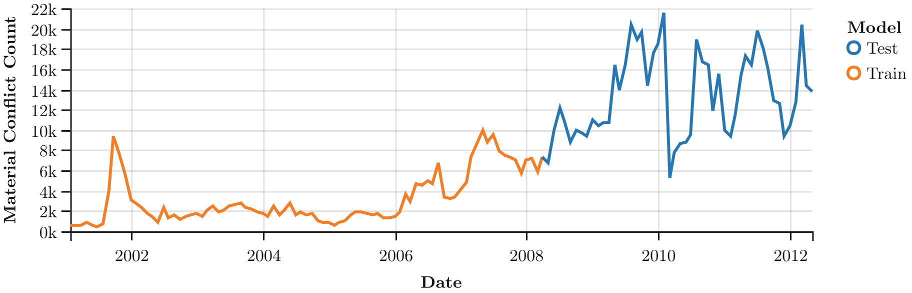


Fig. 2. The number of material conflict events from 2001 to 2012 as recorded by GDELT. The orange segment represents the portion of the time series explicitly used for training, and the blue shows the portion used for model evaluation.

To train both the ARFIMA and LSTM models on the material conflict count per month, the original dataset must first undergo some preprocessing. Firstly, the events must be aggregated temporally, in our case, to the month level. To do this, we filtered the dataset to the desired historical time range and calculated the total count of identified material conflict events for each month, giving us 136 unique data points, the result of which can be seen in Fig. 2. When temporally aggregating time series data much consideration is usually warranted, however, for the purpose of this study we focus on the monthly level, as historical global event data research has shown the monthly level to provide better results [21].

5.2 Model Architectures

For the ARFIMA model, we used the automatic parameter estimation functionality provided by the R forecast library, namely the 'arfima' function, which attempts to identify the ideal model parameters using statistical tests. When predicting future values, the function provides confidence intervals and a mean estimate, but for this study, we utilise the mean value.

For LSTM models, there are additional design decisions such as the network size, the activation functions, the initialisation scheme, and regularisation methods. Since we are training the model in a sequence to sequence fashion, the LSTMs were constructed to take as input, a vector consisting of the numerical event features, and predict a vector which consists of the same features but for the next time step. A visualisation of the process in provided in Fig. 3.

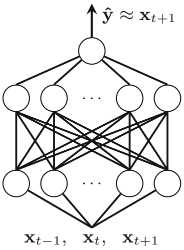

$\hat{\mathbf{y}} \approx \mathbf{x}_{t+1}$
···
···
$\mathbf{x}_{t-1}$, $\mathbf{x}_t$, $\mathbf{x}_{t+1}$

Fig. 3. A diagram of the LSTM model architecture used for this investigation. The network is comprised of two LSTM layers with a fully connected layer.

After some initial experimentation, our network setup contains two LSTM layers with 512 nodes each. We used the standard activation functions and initialisation scheme provided by the 'Keras' library, and added a 50% chance of dropout to each node as regularisation. Since ANNs tend to train more effectively with normalised inputs, the min-max normalisation method was applied to each feature, based on the minimum and maximum values observed in the original training portion of the dataset.

5.3 Training Procedures

To evaluate performance in real world contexts, we devised two distinct training procedures. The first involves training both LSTM and ARFIMA models on the original portion of the data, shown in orange in Figs. 2 and 4. When making predictions on the test component of the data we fixed the models' parameters, ensuring that they are not allowed to take into account new information present in the test set. This would be akin to scenarios where the actual target values remain unknown for data update intervals which extend past an actionable time window of prediction.

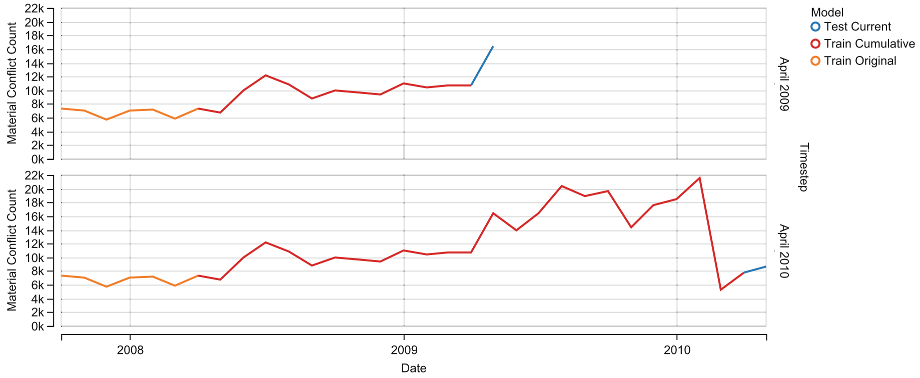


Fig. 4. A demonstration of how additional training data is incorporated into the models at two different time steps. With this ‘real-time’ setup, for each prediction, the models are allowed to update their parameters using all available historical observations. The red segment shows the additional training data provided to the model, and the blue section shows its relevant prediction target.

The second approach is to replicate a real-time predictive scenario, such as that provided by the GDELT data. Again each of the models is initially trained on the training set, but now for the prediction at every subsequent time step in the test set, the models are allowed to adjust their parameters given the additional information (i.e. Fig. 4).

For each of these training procedures and for all models, at each time step, they are presented with the event features for the current month and are tasked with predicting those values for the next month. This provides a solid basis for comparison how ARFIMA and LSTM models perform given different event data contraints. For this investigation, we refer to the first training procedure as the ‘fixed’ method and the second as the ‘real-time’ method.

6 Results

As a baseline each of the models were compared against a naive method, which is simply predicting that the next month will be the same as the last. For the sake of clarity, the naive method is omitted from the plots below, as this is simply a delayed version of the actual data.

Figure 5 shows the predicted number of material conflict events in Afghanistan. Each fixed model attempted to make a prediction for the number of material conflict events for the following month without updating its parameters. From Fig. 5 we can see that the ARFIMA model matches more closely with the shape of the actual series, however, there seems to be a time lag present in its predictions, similar to the naive approach. The fixed LSTM appears to be capturing a smooth version of the original dataset, however, it fails to properly account for the magnitude changes in the original series.

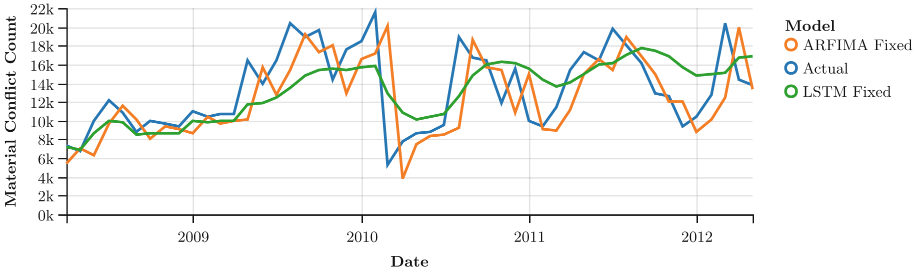


Fig. 5. Predictions made by ARFIMA and LSTM models on the Afghanistan time series with the fixed training procedure.

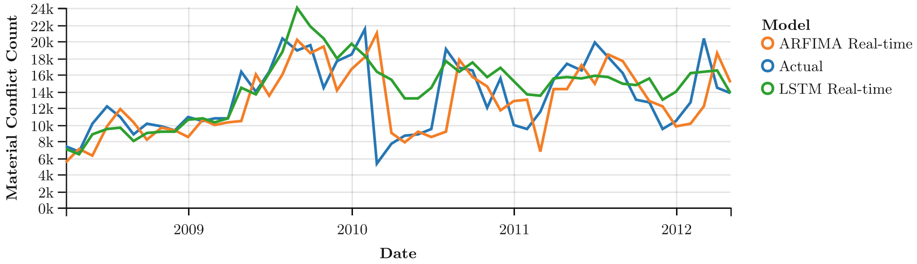


Fig. 6. The predictions made by ARFIMA and LSTM models on the Afghanistan time series using the 'real-time' training approach.

When allowed to take account of new data both models elicit different behaviours. Figure 6 shows the predictions made by both models when they are allowed to re-estimate their parameters based on the cumulative historic test data. The ARFIMA model no longer overshoots the large drop in early 2010, however, it now overestimates the drop present in early 2011. Overall the real-time ARFIMA model appears to have the same time lag behaviour as the fixed version, however, it appears to be making predictions which diverge further from the naive method.

The real-time LSTM model differs significantly from the behaviour of its fixed version. It no longer appears to be a smooth version of the original series, but rather, the model attempts to predict quite significant divergences from the naive method as can be seen in mid-2009. Additionally, the LSTM seems to identify the 2010 downward jump before it happens, however, doesn't account for the magnitude of the change. Allowing the models to use all available historical data seems to have allowed them to make closer predictions, however, it is difficult to tell without evaluating the errors across all points in the test set.

To evaluate these results we used three different commonly used metrics, the Mean Absolute Error (MAE), the Root Mean Squared Error (RMSE), and the Mean Absolute Percentage Error (MAPE) that make different assumptions about how to weight the extent to which a prediction is incorrect. Table 1 provides all three scores for all models tested.

Table 1. The error scores for each model and each of the training methods. The lowest error scores for each metric is highlighted in bold for visibility.

Model	Method	MAE (σ)	RMSE (σ)	MAPE (σ)
Naive	NA	2632.68 (±2809.32)	3829.55 (±2809.32)	22.94 (±42.57)
ARFIMA	Fixed	2713.66 (±2747.31)	3841.96 (±2747.31)	23.09 (±39.23)
LSTM	Fixed	2674.81 (±**1997.96**)	3326.65 (±**1997.96**)	21.70 (±**22.67**)
ARFIMA	Real-time	2610.70 (±2774.19)	3789.19 (±2774.19)	22.47 (±40.99)
LSTM	Real-time	**2305.31** (±2208.42)	**3177.11** (±2208.42)	**21.57** (±32.79)

As Fig. 2 shows, the magnitude and frequency of variations in the time series violate many of the assumptions of standard hypothesis testing, and in fact, the distribution of error values was typically trimodal, with large positive/negative errors at spikes in the real data. Therefore to compare between two approaches a and b we modelled the distribution of the *differences* between the normalised relative errors, i.e. of $e_{ab}^t = (pred_a^t - pred_b^t)/actual^t$. The null hypothesis is that there is no difference in predictive quality, so the e_{ab}^t are sampled from a distribution with mean 0. In most cases e_{ab} was unimodal but skewed, so we used the more conservative non-parametric Wilcoxon signed ranks test to check whether the observed differences are statistically significant.

Comparing to the naive baseline we found that the fixed ARFIMA performed significantly worse ($p < 0.01$) whereas the observed difference between the baseline and fixed LSTM was not significant. Both real-time models predicted significantly more accurately than the baseline ($p < 0.05$). Finally, we found that the real-time LSTM performed significantly better than both the real-time ARFIMA and the fixed LSTM ($p < 0.01$).

7 Discussion

The potential capabilities of LSTMs reach far beyond what we have observed. For one, LSTMs are able to not only predict all provided training features, they can be trained to predict entirely different features, such as the probability of specific types of events occurring.

There are a few limitations of the chosen methodology when it comes to taking advantage of event data. By treating the tasks as a time series problem, we are ignoring categorical features which contain useful event information such as geospatial relationships, e.g. the source and target actors, their ethnic groups, and religions. Using some method of representing these categorical features in a numerical vector space would allow LSTMs to incorporate this additional information. Also, we focus only on the subset of GDELT data that falls under the material conflict QuadClass. Removing the other categories prevents the network from identifying relationships between different types of events, or whether one type of event precedes another. For example, usually verbal conflict events would precede material conflict events. Additionally, the dataset is aggregated to the monthly level, which is common in event data analysis. However, LSTMs have the potential to account for the variance present in the weekly, daily, or even hourly levels. Given that ANNs perform better given more data, training on a fine-grained level of temporal aggregation might improve performance for the LSTM model even though the ARFIMA's would suffer.

Currently, when new data is encountered, the real-time LSTM trains an additional epoch using both the original training data and the new data observed since then. Whilst this provides a promising start, some issues for further investigation remain. Firstly, there is a trade-off between additional training time and the number of real-time training epochs to account for. Also, in a real context, including the entire history would cause the model to progressively become slower. Therefore, a fixed length historical window may likely be required.

One last point of note is that ANNs require some form of normalisation in order to be trained effectively, and LSTMs are no exception. We used a basic min-max normalisation scheme which defined upper and lower bounds for each feature based on the training set, and then future data was normalised accordingly. This choice of normalisation is not ideal as some of the features have large outliers which squash the majority of values to within a small range. Therefore, an improved normalisation scheme would likely provide better results.

8 Conclusion

We have presented a comparative study between LSTM and ARFIMA models for multivariate time series prediction. We show how LSTMs are capable of maintaining long-term memory states that can be informative for modelling changes in the underlying time series, using a case study on material conflict world news events. In both the LSTM and ARFIMA models, we observe a difficulty in identifying the true nature of the time series, however, we also find that the LSTM is more suitable for incorporating additional information that may help give a closer approximation to the underlying data. In our future work, we plan to investigate how more varied attributes can contribute towards the prediction accuracy for LSTMs. We also intend to explore the internal functions of the LSTM to diagnose uncertainties in how the network handles challenging information such as modelling real-world activities.

References

1. Brandt, P.T., Freeman, J.R., Schrodt, P.A.: Real time, time series forecasting of inter- and intra-state political conflict. Conflict Manag. Peace Sci. **28**(1), 41–64 (2011). doi:10.1177/0738894210388125
2. Busseti, E., Osband, I., Wong, S.: Deep learning for time series modeling. Technical report, Stanford (2012)
3. Cadena, J., Korkmaz, G., Kuhlman, C.J., Marathe, A., Ramakrishnan, N., Vullikanti, A.: Forecasting social unrest using activity cascades. PLOS ONE **10**(6) (2015). doi:10.1371/journal.pone.0128879

4. Esteban, C., Staeck, O., Yang, Y., Tresp, V.: Predicting clinical events by combining static and dynamic information using recurrent neural networks (2016). http://arxiv.org/abs/1602.02685
5. Gerner, D.J., Abu-jabr, R., Schrodt, P.A., Yilmaz, M.: Conflict and Mediation Event Observations (CAMEO): A New Event Data Framework for the Analysis of Foreign Policy Interactions. International Studies Association, New Orleans (2002). http://www.ku.edu/~keds
6. Goldstein, J.S.: A conflict-cooperation scale for WEIS events data. J. Conflict Resolut. **36**(2), 369–385 (1992). doi:10.1177/0022002792036002007
7. Greff, K., Kumar, R., Koutník, J., Steunebrink, B.R., Schmidhuber, U.: LSTM: a search space odyssey (2015). https://arxiv.org/pdf/1503.04069.pdf
8. Hammond, J., Weidmann, N.B.: Using machine-coded event data for the micro-level study of political violence. Res. Polit. **1**(2), 1–8 (2014). doi:10.1177/2053168014539924
9. Hochreiter, S., Bengio, Y., Frasconi, P., Urgen Schmidhuber, J.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies (2001)
10. Hyndman, R.J., Athanasopoulos, G.: Forecasting: Principles and Practice, online edn. OTexts (2013)
11. Karpathy, A.: The unreasonable effectiveness of recurrent neural networks (2015). http://karpathy.github.io/2015/05/21/rnn-effectiveness/
12. Karpathy, A., Johnson, J., Fei-Fei, L.: Visualizing and understanding recurrent networks (2015). http://arxiv.org/abs/1506.02078
13. Lipton, Z.C., Kale, D.C., Elkan, C., Wetzell, R.: Learning to diagnose with LSTM recurrent neural networks. In: International Conference on Learning Representations, vol. 4 (2015). http://arxiv.org/abs/1511.03677
14. Perry, C.: Machine learning and conflict prediction: a use case. Stability: Int. J. Secur. Dev. 1–18 (2013). http://dx.doi.org/10.5334/sta.cr
15. Phua, C., Feng, Y., Ji, J., Soh, T.: Visual and predictive analytics on Singapore news: experiments on GDELT, Wikipedia, and ^STI (2014). http://arxiv.org/abs/1404.1996, http://www.sas.com/singapore
16. Qiao, F., Li, P., Zhang, X., Ding, Z., Cheng, J., Wang, H.: Predicting social unrest events with hidden Markov models using GDELT (2013)
17. Raleigh, C., Hegre, H.: Introducing ACLED: an armed conflict location and event dataset. In: Disaggregating the Study of Civil War and Transnational Violence (2005)
18. Stoll, R.J., Subramanian, D.: Hubs, Authorities, and Networks: Predicting Conflict Using Events Data. International Studies Association (2006)
19. Weidmann, N.B., Ward, M.D.: Predicting conflict in space and time. J. Conflict Resolut. **54**(6), 883–901 (2010). doi:10.1177/0022002710371669
20. Yonamine, J.E.: A nuanced study of political conflict using the global datasets of events location and tone (GDELT) dataset. Ph.D. thesis, The Pennsylvania State University (2013)
21. Yonamine, J.E.: Predicting future levels of violence in afghanistan districts (2013). http://jayyonamine.com/wp-content/uploads/2013/03/Forecasting_Afghanistan.pdf
22. Yuan, Y.: Modeling inter-country connection from geotagged news reports: a time-series analysis (2016). http://arxiv.org/abs/1604.03647

A Comparison Study on Flush+Reload and Prime+Probe Attacks on AES Using Machine Learning Approaches

Zirak Allaf(✉), Mo Adda, and Alexander Gegov

University of Portsmouth, Portsmouth, Hampshire, UK
{zirak.allaf,mo.adda,alexander.gegov}@port.ac.uk

Abstract. AES, ElGamal are two examples of algorithms that have been developed in cryptography to protect data in a variety of domains including native and cloud systems, and mobile applications. There has been a good deal of research into the use of side channel attacks on these algorithms. This work has conducted an experiment to detect malicious loops inside Flush+Reload and Prime+Prob attack programs against AES through the exploitation of Hardware Performance Counters (HPC). This paper examines the accuracy and efficiency of three machine learning algorithms: Neural Network (NN); Decision Tree C4.5; and K Nearest Neighbours (KNN). The study also shows how Standard Performance Evaluation Corporation (SPEC) CPU2006 benchmarks impact predictions.

Keywords: Side-channel attack · Machine learning · Flush+Reload · Prime+Probe · AES

1 Introduction

Data sensitivity has assumed increasing importance, and this is particularly true in cloud computing. The primary use of cryptographic techniques on the Internet and in cloud systems is to protect such sensitive data as, among others, patient records, banking transactions and social web accounts and posts. There have been consequent attacks, designed to steal sensitive data, that target such critical cryptographic elements as secret keys, look-up tables and mathematical operations including square multiplication.

The use of machine learning has been studied in a variety of domains, with particular emphasis on anti-virus work to protect individual computers, Intrusion Detection Systems (IDS) to provide greater network security, and spam detection to improve security of information. Machine learning filters out the noise, enabling complex and noisy datasets to be categorised. We therefore proposed a study that will compare three popular machine learning methods (NN; C4.5; and KNN) to establish which will achieve the highest classification level in order to detect.

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_17

Flush+Reload (FR) and Prime+Probe (PP) attacks, making use of malicious activities performed during the attack stages. The attacks target Last Level Cache (LLC) and we will use machine learning approaches that rely on hardware features indicating the state of CPU.

The focus of this paper is on the ability of machine learning methods to detect loops that can be used for side channel attacks and other malicious attacks, relying on Model Specific Registers (MSR). In a multicore system, a large number of processes share a limited number of cores, which means that every process will have access to the components inside the core in which it runs, including MSR, whether or not accessed material is connected with the process in question. It follows that an increase in workload will affect detection models, so this paper addresses the question: Does an increase in workload negatively impact detection rates?

SPEC CPU2006 is used to create a range of scenarios featuring different int and floating workloads, which generate noise, to show what impact they have, for both FR and PP attacks, on the detection model. In addition, this paper compares for accuracy and efficiency (measured by how fast they detect an attack) of the machine learning methods: C4.5, KNN and NN.

The organisation of this paper is as follows. Section 2 examines previous research into attacks and their detection. Section 3 describes the theoretical concepts involved in the three algorithms used in our research. Our findings appear in Sect. 4; discussion and analysis of the results are in Sect. 4.3; and Sect. 5 contains the conclusion.

2 Background and Related Works

This section examines previously suggested techniques and methods for both attacks and the detection of attacks, for which a number of approaches have been used. This section also examines the most important of the key components involved in attacks and in defence against attacks.

2.1 Performance Monitor Unit (PMU)

Modern CPUs contain a PMU to make it possible for programmers to monitor the execution of a program or of a specific piece of code within the program [7]. Multicore processors contain one PMU in each core. A PMU comprises registers (called Performance Monitoring Counters (PMCs)) that maintain account of events of different types that occur within the CPU. There are more than 200 such events and they include: cache misses; elapsed and retired instructions; branch predictions; and hardware stalls. In a Sandy Bridge processor, the PMCs comprise two sets of counters: three fixed-function counters to count core cycles, reference cycles and core instructions; and four general-purpose counters capable of counting any events supported by the CPU model. The processor can also select four specific performance-related events to be monitored simultaneously, and can monitor more than four counters through multiplexing. There is, however, an overhead inherent in multiplexing mode due to switching between counters.

2.2 Side Channel Attacks

The first practical attack against the cryptographic algorithm DES occurred in 2003, and was proposed by Tsunoo et al. [18]. This kind of attack normally operates by stealing secret keys and depends on vulnerabilities in the cryptographic algorithms rather than a “brute force” attempt to apply all possible key combinations. CPU designers attempt to conceal the fundamental details of CPU components, but researchers with malicious intent have found vulnerabilities in CPU components and exploited them for side channel attacks. These vulnerabilities can be exploited through weaknesses in such OS features as memory deduplication and shared libraries [4,17]. Software companies, having found that attacks of this sort exploit features that exist to accelerate performance, have disabled such desirable features of their software as page sharing between two processes and memory deduplication.

CPU cycles were the original key factors in both attack and countermeasures (2004–2010). Attackers sought to monitor such processes as cache utilisation by the use of CPU cycles. Measuring their activities made it possible to use statistical methods to deduce cache lines recently used by the intended victim.

In the early days of side channel attacks, the most frequent target was L1 cache with low bandwidth in 2004 [1], 2005 [15], 2006 [14], 2009 [16], in 2012 [22]. In 2014, Yarom et al. [20] used LLC to obtain a fast attack to retrieve over %90 of key bits. Then Irazoqui et al. [9] proposed even faster attack and the whole key bits became recoverable in less than one minute in the cloud systems. Furthermore, Irazoqui et al. [8] made it possible through huge pages for attackers to obtain physical addresses. Where pages of 2 MB rather than 4 KB are enabled, and the data put into CPU caches, the attacker can deduce entire physical addresses.

2.3 Flush+Reload (FR)

This mechanism, first named time+evict and applied on L2 cache, was discovered in 2011 by [5]. Three years later, Yarom et al. [20] documented a Flush+Reload attack against RSA applied on inclusive LLC and capable of recovering 96.7% of the bits from secret keys between isolated processes. That same year, Irazoqui et al. [9] showed the ability to recover the secret key in less than one minute.

We applied a Flush+Reload attack against the OpenSSL implementation of AES from [9]. AES is a T-table based algorithm. This table[1] can be shared between unrelated processes on the same system, meaning that attacker and intended victim can share the same page. The attack requires page sharing and for static memory to be set by disabling Address Space Layout Randomisation (ASLR). LLC acts as a covert channel between the attacker and victim processes. The attacker uses the `cflush` instruction to observe the victim’s processes. The attacker begins by flushing one line in LLC and then scans the range of addresses in the T-table. Examining the location of the T-table in `libcrypto.so` file enables the attacker to find the table’s start and end addresses. A short access time means a hit; otherwise this is a miss.

[1] The shared library that OpenSSL produces during compilation is libcrypto.so.

In a shared library, the attacker's action in flushing cache line(s) removes all data in CPU caches, since attacker and victim are accessing the same shared file. The victim must therefore bring data back to the CPU caches. The result is local core events such as cache misses.

2.4 Prime+Probe (PP)

Cloud service providers disable memory deduplication to prevent FR attacks, but attackers use PP to carry out side channel attacks because PP requires neither page sharing nor ASLR to be enabled. Instead, the attacker evicts specific set(s) of data from cache memory and waits for a victim to evict its monitored set(s); when access time is compared, a fast access means that the cache line has been touched. The details can be found in [8].

2.5 Detection and Mitigation

A number of approaches have been used for the detection of side and covert channel attacks. Zhang et al. [21] proposed a statistical analysis of cache access-driven attacks mainly that would rely on CPU cycles to monitor accessed and non-accessed cache-based (miss/hit) attacks. Briongos et al. [2] suggested detection based on FR attack against AES by analysing the flush instruction of multiple cache lines that forms the core of the attack. Their model relied for the most part on the CPU cycle as a primary source for data collection.

In another approach, PMU registers were used to give greater granularity so that features supporting detection mechanisms could be extracted at higher resolution. Zhang et al. [21] proposed CloudRadar, where detection of signatures and anomalies detect existing and new forms of side channel attacks as well as such other cache attacks as denial of service attacks against CPU caches.

Kayaalp et al. [10] proposed Relaxed Inclusion Caches (RIC) to mitigate side channel attacks, while Vogl et al. [19] suggested PMC-based trapping as a way of monitoring programs at the instruction-level within VMs. The authors showed that it was possible to monitor a specific instruction in cloud systems.

More recently, Nomani et al. [13] proposed a detection and mitigation mechanism by injection (integrating or hooking) the OS system scheduler to monitor memory usage to detect the existence of malicious programs. The author primarily focused on integer and floating-point units and their effectiveness in measuring CPU such component usage as the CPU cache. They injected the OS scheduler to deploy the NN machine learning algorithm in user-space and collected data through the kernel from the PMC registers to predict malicious processes by identifying and separating two processes that are memory intensive.

3 Methodologies

In this work, we present three common supervised algorithms to classify Flush+Reload and Prime+Probe side channel attacks against AES, and then compare the results of each method used to determine which one most efficiently detects the attack.

3.1 Principle Component Analysis (PCA)

PCA is an unsupervised machine learning algorithm widely used in dimensional reduction to facilitate classification. It is a simple and widely used algorithm which finds the direction of spread of data with the greatest variance and then generates new coordinates.

3.2 Neural Network (NN)

NN is a supervised machine learning algorithm. It can build a predictive model by learning from historical data the patterns to throughput binary or multiclass classification.

The ability of NN to self-learn from examples allows researchers to train NN with features from CPU events from which it acquires the knowledge to classify CPU activities into malicious and non-malicious. Neural network architecture can generally be categorised into: single-layer feed-forward network; multi-layer feed-forward network; and recurrent network. A number of other types have emerged, however, including: perceptron; backpropagation; self-organising map; adaptive resonance theory; and radial basis function.

Ngiam et al. [12] showed the efficiency of the algorithm in dealing with low dimensional data sets. This can work more efficiently with PCA that reduces the dimension of our data. To accelerate the learning process, we use PCA to reduce the dimension and then pass it to the optimisation algorithm L-DFGS, which is efficient for small data sets.

In choosing between three activation functions, we have considered speed and accuracy. Because such attacks are fast, data can be retrieved in less than one minute. Recently, Kingme et al. [11] introduced Adaptive Moment Estimation (ADAM), an optimisation technique used in NN that is fast, computationally efficient and requires less memory than DFGS. It also deals efficiently with large data sets. The Quasi-Newton method, on the other hand, is computationally expensive and requires more memory to store the Hessian matrix, while LDFGS accelerates the speed [3] of deep network learning. They used the algorithm for large data sets and showed it to be faster than the SGD algorithm. Limited-memory DFGS does not store Hk and is therefore faster than DFGS. Faced with a large data set, as we mentioned, ADAM is faster.

3.3 K Nearest Neighbour KNN

KNN is a non-parametric classifier and a lazy learner classification method. Each tested data class is predicted by measuring the similarity of test data and training set records. Broadly, in KNN the classification process carried out for the test data relies on its neighbour, compared with K closest training example where the output is the class membership. Any sample of data points can be classified by a majority vote of its neighbour. K represent a very small integer, so if $\mathrm{k} = 1$, then k is assigned to the class of single nearest neighbour. Functions available for use in similarity calculations include Euclidean, Hamming, Manhattan and Minkowski.

3.4 C4.5

C4.5 is one of the oldest supervised machine learning algorithms. It is uniquely easy to read and understand. The goal is to build a model that predicts the value of a target variable by asking multiple linear questions one by one to create a boundary. Future data is classified using a very simple data structure which is called Tree.

It is a statistical classifier like other classifiers, and uses a set of data to train and build a decision tree model using the concept of information entropy. The trained data is split into n-dimensional vectors which represent features of the sample data and its class. C4.5 selects the most efficient features to divide its set of samples into subsets boosted in one class or the other using the following equation:

$$E(S) = \sum_{i=1}^{n} -P(C_i) * log_2 P(C_i) \tag{1}$$

$$G(S, F) = E(S) - \sum_{i=1}^{m} P(F_i) * E(S_{Fi}) \tag{2}$$

When, $S = \{s_1, s_2, \ldots s_n\}$ represents the set of samples. $E(S)$ is an informational entropy of S, $G(S, F)$ is a function to gain S after splitting feature F. $P(C_i)$ calculates the frequency of class C_i in S. The function $E(S_{Fi})$ generates a subset of S with items that have F_i value.

4 FR and PP Attack Detection

In this work, we demonstrate the hardware specifications used for data collection.

4.1 Hardware and Software Specifications

The experiment was conducted on HP Proliant DL360 G7 with Intel's Xeon X5650 2.66 GHz processor with 16 GB RAM running Ubuntu 14.04. The various tests used SPEC cpu2006.

4.2 Experiment

In this section, we conduct an experiment study by using data collected from the experiment. We create an agent process that encrypts fake data with the intention of simulating a victim, and used the custom Loadable Kernel Module (LKM) to access PMC registers with minimum overhead in order to gain high resolution data. Our data set consists of 7 features, three fixed function registers (core cycles, reference cycles and core instructions) and four more efficient programmable events. For this experiment, we selected the most efficient events having a positive impact on the classification of the selected methods by considering their relationship to attacks. We collected 100 samples, which is optimal, $S = \{S_1, S_2, \dots S_{100}\}$ and for each S_i we flattened all features into a single row (one big vector of mixed data). Each sample is so arranged that $S_1 = \{X_{(1,1)} \dots\dots X_{(1,n)}, X_{(2,1)} \dots\dots X_{(2,n)}, \dots\dots X_{(7,1)} \dots\dots X_{(7,n)}, y\}$. n is the number of encryption iterations executed by the victim to collect the specified event. In this experiment, we used 3000 iterations for each event. y is the binary class which represents attack or normal.

The data is collected under two scenarios, one for light and one for heavy workloads. In the first scenario, high resolution data was ensured by running only victim and attack programs. In the second scenario, we added noise by running additional applications from SPEC SPEC2006[2], two int applications (`bzip2` and `gcc`) and two floating applications (`bwaves` and `dealII`)

The dataset was split into training and testing sets to prepare for machine learning algorithms. Training sets contain 80 samples and 20 testing sets. To determine the influence of different data set splits under each method, we split the data sets randomly into 20-fold cross validations.

In this study, we show the impact of malicious loops running inside FR and PP attack programs on the victim's processes, which use a cryptographic algorithm to encrypt sensitive data. Our hope was to detect the attack in both light and heavy workloads. The attacker would try to interfere with the victim's processes and synchronise itself on the shared LLC by monitoring its cache memory activities and using statistics to deduce the cache lines most recently used by the victim. We also hoped to detect the attack in the shortest possible time less than 5 s; when the efficient attack [9] requires over 50 s to recover the whole key bits. This experiment can be applied in cloud systems, except for the additional overhead, which is produced by an additional translation layer. This definitely reduces the resolution rate detection, as the most recent detection work [2,6,21] shows the difference in accuracy rates between native and cloud systems.

The shared library co-allocates two unrelated processes on LLC on the same machine. Thus, we would detect malicious FR and PP attack activities when a malicious loop is run to synchronise with the victim process in order to give the attacker a chance of accessing the shared memory. The aim of our hypothesis was to evaluate the best classification method and the impact of SPEC in detecting such attacks with a high rate of accuracy even with the loading of the benchmark.

4.3 Result Analysis and Discussion

We are looking for an optimal classifier that works most accurately and efficiently among selected methods under both light and heavy workloads. Each of the three algorithms presented in the previous section was run on each of the 20-fold splits of the data set into training and testing sets. Based on previous studies and the results gained from our experiment, we compare the methods based on accuracy and efficiency because these two factors are important to victims when

[2] SPEC SPEC2006 is widely used to evaluate performance of computer systems https://www.spec.org/.

Table 1. Classification accuracy for the three methods C4.5, PCANN and KNN, against two attacks Flush+Reload (FR) and Prime+Probe (PP).

Classification accuracy on FR and PP							
SPEC	Benchmarks	C4.5		NN		KNN	
	Attack	FR	PP	FR	PP	FR	PP
	No SPEC	0.97	0.98	0.93	0.76	0.85	0.83
SPECint	bzip2	0.91	0.96	0.8	0.8	0.8	0.78
	GCC	0.87	0.94	0.77	0.79	0.84	0.8
SPECfp	bwaves	0.74	0.74	0.73	0.73	0.73	0.74
	dealII	0.75	0.7	0.7	0.64	0.63	0.7

dealing with sensitive data. For accuracy, the victim needs to correctly classify the attack, while for efficiency, the victim needs to detect the attacks quickly before the attacker retrieve the whole key-bits and disrupt the attack.

The results, presented in Table 1 and Fig. 1, show the accuracy of side channel attack classification including FR and PP techniques for all methods in three scenarios without SPEC ($\sim SPEC$), CPECint or SPECfp. The C4.5 algorithm performs with highest accuracy in all scenarios in detecting FR. Without SPEC the success rate is 0.97%. This falls in SPECint to 0.91% and 0.87% in `bzip2` and `gcc` respectively. There is a further decrease to 0.74% in SPECfp to 0.74% and 0.75% for `bwaves` and `dealII` respectively. However, in PP detecting it classifies better in SPEC, `bzip2` and `gcc`. Stays the same in `bwaves`, but it is worse in `dealII`. This is because, in a PP attack, the attacker uses more CPU components and this maximises the number of occurrences of specified events.

PCANN is good at detecting FR without SPECint and SPECfp, but performs poorly in detecting a PP attack even without benchmarks. KNN has a similar accuracy rate for FR and PP attacks, but drops down in a PP attack. The results from C4.5 are therefore seen to be more reliable and robust than from PCANN and KNN. This because C4.5 method deals with noisy data better than the rest of the methods due to fast data exploration and find the relation between the most significant variables. Turning to efficiency, Decision Tree is more inefficient than PCANN and KNN, but still detects the attack with reasonable efficiency.

However, running `bzip2`, `gcc` applications in order to load the SPEC benchmark showed C4.5 to have a higher level of accuracy than either NN or KNN. KNN efficiency was slightly lower than C4.5, but NN had the worst accuracy. When `bwaves` is loaded, they all have poor accuracy, because `bwaves` is in the float application group and floating operations make heavier use of CPU components than integer operations, resulting in a high number of cache misses and degrading the training of the classification models.

The results shown in Fig. 2 indicate that the size of the data set will be enough for the detection agent to be able to learn malicious activities in a very short time; the worst case is less than 1 s and compares well with recent and fast FR attack by [9] that needed over 50 s to retrieve the entire key bits. The agent can detect the attack early enough to prevent the attacker stealing the whole key and can perform the actions necessary to stop the attacker.

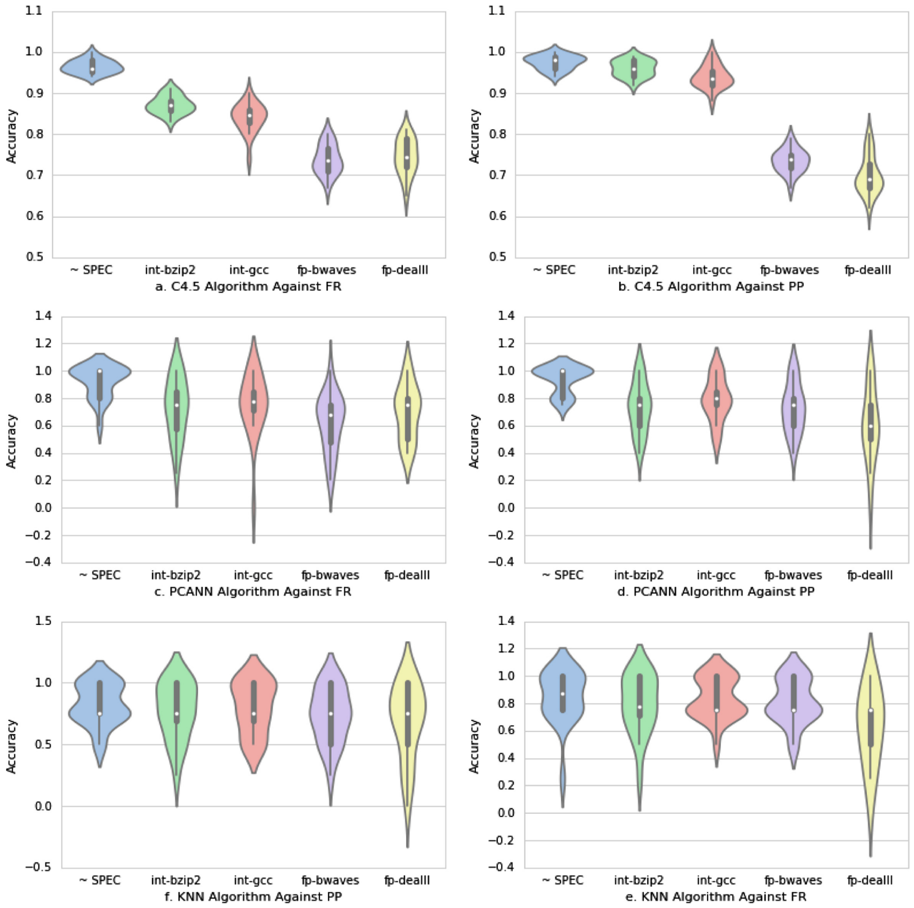


Fig. 1. Comparison accuracy rate of C4.5, PCANN and KNN. Each violin shape represents the sample distribution of 20 fold across validations of the experiments. The accuracy rate of detecting FR and PP attacks are compared under different workloads without SPEC ($\sim SPEC$), int-bzip2, int-gcc, fp-bwaves and fp-dealII. The inflated area indicates the density of the accuracy rate in 20 fold cross validation. Inside each violin, the accumulated black dots in the middle represent the accuracy of the tests and the white dot represents the median of the accuracy over all the tests.

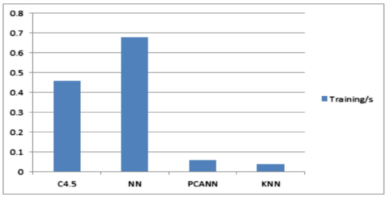


Fig. 2. Comparison of time execution for training in selected classifiers

It follows that detection of FR and PP attacks will be difficult in noisy environments, and especially so when intensive floating-point applications are running. This is because floating point applications make use of CPU caches and generate a large number of cache misses, while detection relies partly on CPU cache misses to detect FR and PP attacks.

These methods can be used by a host OS, in both native and cloud systems (though it must be noted that cloud systems are less accurate than native systems) to distribute a fake process running cryptographic algorithms such as AES to identify malicious activities and prevent them from stealing the whole of a secret key.

The results show that system activities in the background do not significantly impact the results, with all methods performing well, but intensive workloads introduce more noise into the system and have a negative impact on the accuracy. In particular, SPECfp benchmark made the result worse than SPECint benchmarks, because the floating operations cause high occurrence of CPU events. Loading SPEC benchmarks places stress on CPU components, and particularly on caches. A SPECfp benchmark interferes with the monitoring processes and introduces noise to the environment.

5 Conclusions

We investigated the use of three classification methods for detecting Flush+Reload and Prime+Probe attacks in two different scenarios, one with and one without SPEC CPU2006 benchmark workloads. We concluded that a heavy workload has a negative impact on the detection rate due to stress on the CPU components. Our findings indicated that the Decision Tree classifier is better than NN and KNN to detect Flush+Reload and Prime+Probe attacks.

References

1. Bernstein, D.J.: Cache-timing attacks on AES (2005)
2. Briongos, S., Malagón, P., Risco-Martín, J.L., Moya, J.M.: Modeling side-channel cache attacks on AES. In: Proceedings of the Summer Computer Simulation Conference, p. 37. Society for Computer Simulation International (2016)
3. Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Senior, A., Tucker, P., Yang, K., Le, Q.V., et al.: Large scale distributed deep networks. In: Advances in Neural Information Processing Systems, pp. 1223–1231 (2012)
4. Gruss, D., Bidner, D., Mangard, S.: Practical memory deduplication attacks in sandboxed Javascript. In: European Symposium on Research in Computer Security, pp. 108–122. Springer (2015)
5. Gullasch, D., Bangerter, E., Krenn, S.: Cache games-bringing access-based cache attacks on AES to practice. In: 2011 IEEE Symposium on Security and Privacy, pp. 490–505. IEEE (2011)
6. Gulmezoglu, B., Eisenbarth, T., Sunar, B.: Cache-based application detection in the cloud using machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pp. 288–300. ACM (2017)

7. Intel® 64 and IA-32 Architectures Software Developer's Manual. Volume 3A: System Programming Guide, Part 1(64) (2011)
8. Irazoqui, G., Eisenbarth, T., Sunar, B.: S$A: a shared cache attack that works across cores and defies VM sandboxing-and its application to AES. In: 2015 IEEE Symposium on Security and Privacy, pp. 591–604. IEEE (2015)
9. Irazoqui, G., Inci, M.S., Eisenbarth, T., Sunar, B.: Wait a minute! a fast, cross-VM attack on AES. In: International Workshop on Recent Advances in Intrusion Detection, pp. 299–319. Springer (2014)
10. Kayaalp, M., Khasawneh, K.N., Esfeden, H.A., Elwell, J., Abu-Ghazaleh, N., Ponomarev, D., Jaleel, A.: RIC: relaxed inclusion caches for mitigating LLC side-channel attacks. In: Proceedings of the 54th Annual Design Automation Conference 2017, p. 7. ACM (2017)
11. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
12. Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., Le, Q.V., Ng, A.Y.: On optimization methods for deep learning. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 265–272 (2011)
13. Nomani, J., Szefer, J.: Predicting program phases and defending against side-channel attacks using hardware performance counters. In: Proceedings of the Fourth Workshop on Hardware and Architectural Support for Security and Privacy, p. 9. ACM (2015)
14. Osvik, D.A., Shamir, A., Tromer, E.: Cache attacks and countermeasures: the case of AES. In: Topics in Cryptology–CT-RSA 2006, pp. 1–20. Springer (2006)
15. Percival, C.: Cache missing for fun and profit (2005)
16. Ristenpart, T., Tromer, E., Shacham, H., Savage, S.: Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 199–212. ACM (2009)
17. Suzaki, K., Iijima, K., Yagi, T., Artho, C.: Memory deduplication as a threat to the guest OS. In: Proceedings of the Fourth European Workshop on System Security, p. 1. ACM (2011)
18. Tsunoo, Y., Saito, T., Suzaki, T., Shigeri, M., Miyauchi, H.: Cryptanalysis of DES implemented on computers with cache. In: International Workshop on Cryptographic Hardware and Embedded Systems, pp. 62–76. Springer (2003)
19. Vogl, S., Eckert, C.: Using hardware performance events for instruction-level monitoring on the x86 architecture. In: Proceedings of the 2012 European Workshop on System Security EuroSec, vol. 12 (2012)
20. Yarom, Y., Falkner, K.: FLUSH+RELOAD: a high resolution, low noise, L3 cache side-channel attack. In: 23rd USENIX Security Symposium (USENIX Security 14), pp. 719–732 (2014)
21. Zhang, T., Zhang, Y., Lee, R.B.: Cloudradar: a real-time side-channel attack detection system in clouds. In: International Symposium on Research in Attacks, Intrusions, and Defenses, pp. 118–140. Springer (2016)
22. Zhang, Y., Juels, A., Reiter, M.K., Ristenpart, T.: Cross-VM side channels and their use to extract private keys. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, pp. 305–316. ACM (2012)

Classifying and Recommending Using Gradient Boosted Machines and Vector Space Models

Humphrey Sheil(✉) and Omer Rana

School of Computer Science and Informatics, Cardiff University, Cardiff, UK
sheilh@cardiff.ac.uk

Abstract. Deciphering user intent from website clickstreams and providing more relevant product recommendations to users remains an important challenge in Ecommerce. We outline our approach to the twin tasks of user classification and content ranking in an Ecommerce setting using an open dataset. Design and development lessons learned through the use of gradient boosted machines are described and initial findings reviewed. We describe a novel application of word embeddings to the dataset chosen to model item-item similarity. A roadmap is proposed outlining future planned work.

Keywords: Gradient boosted machine · Classification · Ranking · Recommender system · Vector space model · Ecommerce

1 Problem Domain: Overview

The primary method used to gather data in the Ecommerce domain is to log browser requests for web pages ordered temporally and grouped by a session ID. These logs are then used to train models which classify users by their intent (clicking, browsing, buying) and what items those users are most interested in. Our motivation is to predict the intent of web users using their individual and group prior behaviour and to select from a potentially large set of available content, the items of most interest to match with a specific user. Correctly identifying user intent and matching users to the most relevant content directly impacts retailer revenue and profit [9].

1.1 RecSys Challenge

This work focused on an open dataset from the ACM RecSys 2015 conference challenge [1]. The challenge ran for nine months, involved 850 teams from 49 countries, with a total of 5,437 solutions submitted. The winners of the challenge scored approximately 50% of the maximum score. A variety of linear and nonlinear classifiers were employed as ensembles and two of the top three accepted submissions [4] and [3] relied heavily on Gradient Boosted Machine (GBM) classifiers, with [3] employing both Neural Networks and GBM.

The challenge dataset is a snapshot of web user activity where users mostly browse and infrequently purchase items from a catalogue. The data is:

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_18

1. Reasonable in size - containing 34,154,697 events grouped into 9,249,729 sessions. The sessions comprise events over 52,739 items distributed over 338 categories, with 19,949 of the items purchased.
2. Imbalanced - buyer sessions represent just 5.51% (509,696) of the total.
3. Incomplete - for example 49.5% of the clicks do not contain a category ID for the item clicked.

The objective function to maximise is:

$$Score(Sl) = \sum_{\forall s \in Sl} \begin{cases} ifs \in S_b \rightarrow \frac{|S_b|}{|S|} + \frac{|A_s \cap B_s|}{|A_s \cup B_s|} \\ else \rightarrow -\frac{|S_b|}{|S|} \end{cases} \tag{1}$$

where:

Sl = sessions in submitted solution
S = All sessions in the test set
s = session in the test set
S_b = buy sessions in test set

A_s = predicted buy items in session s
B_s = actual bought items in session s

The score is maximised by correctly classifying true buyers while minimising the number of false buyers (i.e. clickers). This is followed by the recommendation or ranking task, where for each buyer the exact items purchased are predicted from the click set for that buyer (a buyer can purchase just one item clicked, all or some).

1.2 Wider Applicability

Classification and ranking in order to recommend are not specific to the Ecommerce domain. Multiple other domains such as security, finance and healthcare apply similar techniques to solve domain-specific problems. Our intent is to generalise our framework and approach to multiple domains. However, different domain problems will have substantially different objective functions - Table 3 shows that in the Ecommerce domain a high false negative classification score can still result in a respectable score - it is easy to imagine a problem in the finance, security or healthcare domains where better classification performance is required (but equally better distinguishing data must also be available).

2 Implementation

GBM was selected as the initial model for a number of reasons. The trained models are interpretable in terms of feature usage, gain and coverage. GBM enjoys a robust, fast and scalable implementation in [5]. GBM is also straightforward to train as it iteratively grows an ensemble of Classification and Regression Trees (CART) to learn an objective function with regularisation applied to promote generalisation on unseen data. New trees are iteratively added during training to better model the objective function and correct for the errors made by earlier trees. Figure 1 illustrates the data flow through the primary modules of the system.

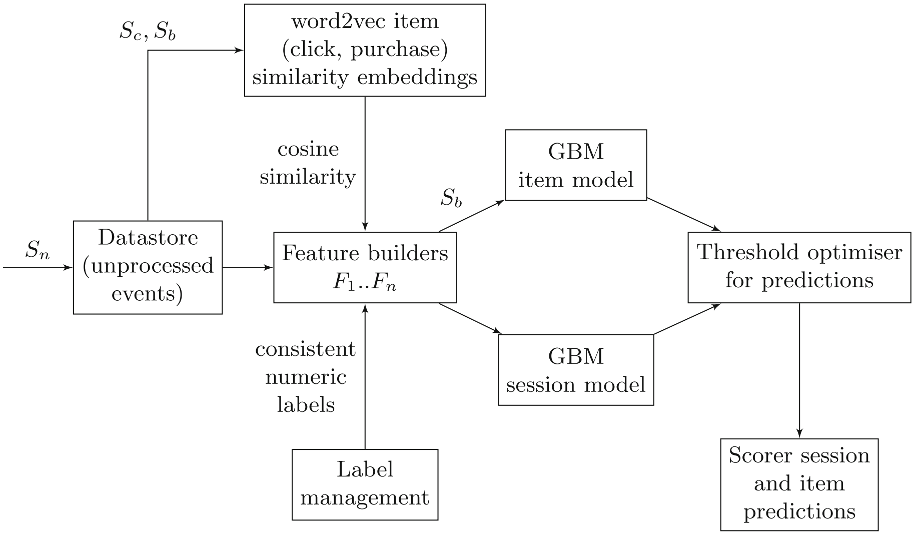

S_c, S_b
word2vec item (click, purchase) similarity embeddings
cosine similarity
S_b
GBM item model
S_n
Datastore (unprocessed events)
Feature builders $F_1..F_n$
Threshold optimiser for predictions
consistent numeric labels
GBM session model
Label management
Scorer session and item predictions

Fig. 1. Primary modules of the end-to-end system implementation. The system currently contains 10 feature builders calculating 68 features in total. Calculated features are saved in LIBSVM format (label:value) and consumed by GBM to build a forest of CART trees. word2vec receives all sessions (S_c and S_b) and is used to calculate two distinct similarity embeddings - modelling items that are frequently clicked together, and items that are frequently bought together. The item model is trained on buyer sessions S_b only, as only buyer sessions can contribute towards the item component of the target score.

2.1 Framework

During the implementation, specific functions and attributes to enable efficient and rapid progress were identified and coded into a re-usable machine learning management framework. The models covered here represent just the first iteration, and this framework will be used with future models. The main properties of the framework are:

1. *Reproducibility.* Threshold optimisation and hyperparameter values have a significant impact on the accuracy obtained and thus final score. The framework supports version control of code, data, logs and configuration relating to each experiment.
2. *Composition and Combinability.* Currently we combine homogenous models together to solve a target task, however the framework also admits heterogenous models.
3. *Rich data querying capabilities.* Spark SQL is used to enable rapid, iterative data analysis to test feature accuracy and to suggest new feature designs.
4. *Consistent Feature generation across models.* Feature re-use across models promotes code re-use across models and experiments.
5. *Labelling and aggregation.* The framework stores labels at session and item level through all segments of the transformation, training and scoring pipeline.

Table 1. The top ten session features after training for 7,500 rounds, ordered by most important features descending. The importance values here and in Table 2 are relative and are calculated as the number of occurrences for each feature in the model trees divided by the total feature occurrences in the model as a whole. Importance values are lower here than for the item model in Table 2 due to the number of features used in the session model. GBM models are interpretable, allowing feature importance scores to be easily calculated. Features are grouped into four types - Temporal, Counts, Similarity and Price. In our opinion, temporal features model user engagement, price features model item competitiveness (item prices rise and fall over time), count features model popularity statistics over the dataset and similarity features model user intent - casual vs focused browsing.

Description	Relative importance	Type
Max time spent on an item (millisecs)	0.049	T
Global buys (last item)/global clicks (last item)	0.048	C
Session duration (millisecs)	0.044	T
Global clicks (last item)	0.043	C
Min item price in session	0.041	P
Max value of (buys/clicks) in the session	0.04	P
Cross entropy of dwelltime across items	0.038	T
Max item price in session	0.038	P
Max click similarity in session	0.037	S
Click similarity standard deviation	0.037	S

2.2 Features

For the session model, 40+ features and a one-hot item vector for the top 5,000 most popular items were calculated from the dataset. For the item model, 20+ features were calculated. The features overlap significantly with other competition submissions [3,4]. The most common features used are well-understood information retrieval metrics - Tables 1 and 2 describe the top ten features for the session and item models, graded by their feature importance score.

2.3 Models and Training

The two terms of the objective function are independent, so a divide and conquer approach is a rational strategy. Two models were trained in parallel - the session predictor and the item predictor. During the training phase, the quality of models generated by GBM is sensitive to values chosen for some key values: the tree depth (max_depth), learning rate (eta) and breakpoint for new tree nodes (min_child_weight). We currently use sensible values for these parameters as suggested by [3], with a hyperparameter search planned in future work.

It quickly became apparent that the classification task is more difficult to learn than the recommending task - Area under the curve (AUC) is used to

Table 2. Top ten item features after training for 5,000 rounds, ordered by most important first. The click and buy similarity metrics carry significant weight in this model, resulting in a focused effort to improve them - beginning with a simple count-based Jaccard similarity, progressing to matrix factorisation using Alternating Least Squares, to the current best solution - using pair-wise cosine similarity on embeddings or vectors calculated for each item. The current embeddings are of length 300, with each vector co-ordinate representing a latent variable modelling the item set. Similarity-based features are strongly represented in the table, showing the effectiveness of the current similarity implementation.

Description	Relative importance	Type
Summed buy similarity	0.099	S
Max click similarity in session	0.097	S
Summed click similarity	0.097	S
Buy similarity standard deviation	0.096	S
Std deviation of buy similarity/click similarity × num clicks	0.091	S
Summed buy similarity/click similarity × num session clicks	0.083	S
Item dwelltime in this session	0.065	T
Global clicks for this item	0.064	C
Global item buys/global item clicks	0.063	C
(Global buys/global clicks) × num session clicks for this item	0.054	C

measure training progress on a validation set and the best session classification AUC is 0.851 vs 0.895 for the item prediction task. This is due to the imbalanced nature of the dataset and that some of the most common sessions comprise those with lengths between one and three, removing valuable context from some of the global features (for example cross-entropy, click similarity and buy similarity). We partly mitigated the class imbalance issue by down-sampling clickers by 50% and this resulted in a small score increase. Thus with appropriate hyperparameter selection, GBM appears to be reasonably resistant to over-fitting on the dominant class in an imbalanced setting.

2.4 Inference - Initial Results

The model confidence in predicting user behaviour and recommending items increases based on session length. Therefore the thresholds were selected at a session-length level, instead of using a "one fits all" value. Thresholds for both models were selected using grid search with a stepsize of 0.01 after training, using threshold start and end ranges known empirically to bracket the optimal thresholds. As Table 3 shows, it is necessary to reduce the probability thresholds for session selection to an average of just 0.09 (0.069 if clickers are not under-sampled), compared to an average of 0.47 for item selection. This low session threshold value demonstrates the difficulty of the session classification task. In general, it is important to use dynamic confidence thresholds predicated on session length to maximise both the session and item components of the overall score.

Table 3. Session and item thresholds by session length with scores for the current models, showing the increase in model predictive confidence as the number of events per session grows.

Session length	Session threshold	Item threshold	Session score	Item score
1	0.05	0.4	−3276	3344
2	0.06	0.6	−11224	21620
3	0.07	0.54	−6393	14421
4	0.07	0.51	−4328	11246
5	0.08	0.53	−2620	8222
6	0.09	0.49	−1701	6537
7	0.11	0.44	−1067	4942
8	0.11	0.43	−777	3993
9	0.1	0.44	−601	3125
10	0.08	0.44	−485	2530
11	0.1	0.47	−322	2076
12	0.1	0.42	−252	1702
13	0.08	0.46	−212	1335
14	0.14	0.45	−119	1104
15+	0.14	0.42	−554	6175
Totals			−33931	92373

The current implementation would have placed 6th or 7th (the conference did not contain a paper from the second placed team) on the competition leaderboard out of 850 and makes 99.4% of the GBM-only target score (58,442 vs 58,820 in [3]). The code for the original scoring methodology used in the competition is no longer available [2], however we reverse-engineered and validated the scoring methodology using three input sources - the solution file provided after the competition ended, the solution file provided by the authors of [4] and our own solution file.

2.5 Optimising Click and Buy Item Similarity Features

The optimal similarity measure discovered to date is unique in the competition we believe. Multiple similarity implementations were evaluated including Jaccard similarity and Alternating Least Squares (ALS) matrix factorisation - a staple technique in the recommender community. Currently, items are modelled as words, sessions as sentences and each "word" is transformed [8] into a distributed representation - the embedding or word vector. This feature is trained by maximising its log-likelihood on the training set:

$$J_{\text{NEG}} = \log Q_\theta(D=1|w_t, h) + k \mathop{\mathbb{E}}_{\tilde{w} \sim P_{\text{noise}}} [\log Q_\theta(D=0|\tilde{w}, h)]$$

where:
$Q_\theta(D = 1|w, h)$ = the binary logistic regression probability

h = the context (user session) D = corpus of all sessions
w = word (item ID)
θ = learned embedding vectors

A good vector space model will map semantically similar words close together and this feature exploits this property by calculating item-item similarity using pair-wise cosine similarity. Further experimentation for these features is planned, focusing on document ordering, parameters such as embedding length (currently 300), context (currently 15 words) and the best internal word2vec model to use - Skip Grams vs Continuous Bag Of Words (CBOW).

3 Conclusion and Future Work

The work to date has demonstrated that a homogenous GBM implementation can achieve an impressive score on the competition dataset - competing well with more advanced heterogenous [3] and proprietary [4] solutions. In all of our experiments, GBM functioned consistently well as a robust classifier, therefore we posit that the score achieved relies substantially on careful feature engineering rather than algorithm improvements or hyperparameter selection. Given the preponderance of click/user event datasets in the Ecommerce and recommender domains, we expect the work completed so far to generalise well to other datasets in the same domain. In the future, a more detailed investigation into the word embedding similarity features will be conducted. Hyperparameter search and data set balancing will be used to further improve the current score. Planning beyond this goal and evaluating the current implementation to higher-scoring competition submissions, it will be necessary to combine heterogenous models together to improve further. Using the framework constructed, we plan to implement and benchmark a model optimised for temporal series (potentially LSTM [6]) against the GBM model, and measure what benefits this model enjoys in modelling temporal data one session event at a time compared to GBM and in leveraging the distributed representations used to calculate the click and buy similarities.

References

1. Ben-Shimon, D., Tsikinovsky, A., Friedmann, M., Shapira, B., Rokach, L., Hoerle, J.: RecSys challenge 2015 and the YOOCHOOSE dataset. In: Proceedings of the 9th ACM Conference on Recommender Systems (RecSys 2015), pp. 357–358. ACM, New York (2015). https://doi.org/10.1145/2792838.2798723
2. Personal communication from David Ben-Shimon

3. Volkovs, M.: Two-stage approach to item recommendation from user sessions. In: Ben-Shimon, D., Friedmann, M., Rokach, L., Shapira, B. (eds.) Proceedings of the 2015 International ACM Recommender Systems Challenge (RecSys 2015 Challenge). ACM, New York (2015). Article 3, 4 p. http://dx.doi.org/10.1145/2813448.2813512
4. Romov, P., Sokolov, E.: RecSys challenge 2015: ensemble learning with categorical features. In: Ben-Shimon, D., Friedmann, M., Rokach, L., Shapira, B. (eds.) Proceedings of the 2015 International ACM Recommender Systems Challenge (RecSys 2015 Challenge). ACM, New York (2015). Article 1, 4 p. http://dx.doi.org/10.1145/2813448.2813510
5. Chen, T., Guestrin, C.: XGBoost: A Scalable Tree Boosting System (2016). http://arxiv.org/abs/1603.02754
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997). http://dx.doi.org/10.1162/neco.1997.9.8.1735
7. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. **29**(5), 1189–1232 (2001). JSTOR. www.jstor.org/stable/2699986
8. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space (2013). https://arxiv.org/abs/1301.3781
9. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2003). http://dx.doi.org/10.1109/MIC.2003.1167344

SemCluster: Unsupervised Automatic Keyphrase Extraction Using Affinity Propagation

Hassan H. Alrehamy and Coral Walker(✉)

School of Computer Science and Informatics, Cardiff University, Cardiff, UK
{alrehamy,huangy}@cardiff.ac.uk

Abstract. Keyphrases provide important semantic metadata for organizing and managing free-text documents. As data grow exponentially, there is a pressing demand for automatic and efficient keyphrase extraction methods. We introduce in this paper *SemCluster*, a clustering-based unsupervised keyphrase extraction method. By integrating an internal ontology (i.e., WordNet) with external knowledge sources, *SemCluster* identifies and extracts semantically important terms from a given document, clusters the terms, and, using the clustering results as heuristics, identifies the most representative phrases and singles them out as keyphrases. *SemCluster* is evaluated against two baseline unsupervised methods, TextRank and KeyCluster, over the Inspec dataset under an F1-measure metric. The evaluation results clearly show that *SemCluster* outperforms both methods.

Keywords: Keyphrase extraction · Clustering-based AKE · Unsupervised AKE

1 Introduction

Keyphrases provide a concise description of the essential meaning of a document and play a vital role in many information management tasks. Currently, only a small fraction of digital documents have assigned keyphrases. This is because keyphrase assignment is done manually in most cases, and such an undertaking is laborious and time-consuming. In the booming era of Big Data, automating the task of keyphrase extraction has gained significant attention and research effort. Automatic Keyphrase Extraction (AKE) is a natural language processing (NLP) task concerned with the automatic selection of representative phrases from the body of a document [1]. AKE approaches may be divided into two categories [2]: supervised and unsupervised. In a supervised approach, AKE is cast as a classification problem, and a classifier is employed to determine whether a candidate phrase in a document is a keyphrase. Although many current supervised AKE approaches deliver promising results, they require careful selection of manually annotated training datasets, which itself is tedious and may lead to inconsistency, especially in a heterogeneous processing environment that requires cross-domain tractability [15]. Unsupervised AKE is a much recent trend aimed at discovering the underlying structure of a document without the assistance of machine

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_19

learning, and thus reducing expensive human labour by using various extraction techniques. A typical unsupervised approach uses graph-based modelling [3–5], where the document is modelled as a graph, terms as nodes, and relations between nodes as weighted edges. A typical workflow is: (i) weight importance of each node based on its relations to other nodes, (ii) rank each node based on its weight, (iii) select nodes of top rank as candidates. However, an unsupervised approach may encounter the following challenges regarding keyphrase extraction: firstly, a top-ranking term of statistical and/or semantic importance is not guaranteed to be associated with keyphrases representative of the document theme [6]; secondly, a term representative of the document theme is not guaranteed to be a top-ranking term if it occurs infrequently in the document [2]. In the text segment in Fig. 1, "*Olympics*" appears frequently, so we are not surprised to see many AKEs select "*Olympics*", "*Olympic games*", and "*Olympic movement*" as keyphrases without considering that "*Olympic movement*" has no immediate relevance to the document theme (i.e., "*athelete news*") and is an unsuitable keyphrase. Including an irrelevant phrase as one of the keyphrases for a document can cause confusion in search queries. For instance, a miscellany of documents about "*nonprofit organization*" or "*regulations*" may be returned in response to a search query for documents about "*athlete news*". On the other hand, "*dash*", a term of significant semantic importance, is rarely identified as a keyphrase ("*100-m dash*") by many AKEs because of its infrequent occurrence in the document.

> Canadian **Ben Johnson** left the **Olympics** today "in a complete state of shock," accused of cheating with drugs in the worlds fastest **100-meter dash** and stripped of his **gold medal**. The prize went to American **Carl Lewis**. Many athletes accepted the accusation that Johnson used a muscle-building but dangerous and illegal anabolic steroid called **Stanozolol** as confirmation of what they said they know has been going on in track and field. Two tests of Johnsons urine sample proved positive and his denials of **drug use** were rejected today. "This is a blow for the **Olympic Games** and the Olympic movement," said International Olympic Committee President Juan Antonio Samaranch.

Fig. 1. A news article on "Ben Johnson" from the *DUC-2001 dataset.*

In this paper, we introduce *SemCluster* to address the above challenges. *SemCluster* is an unsupervised clustering-based method for effective keyphrase extraction in any domain. Motivated by [2,6], *SemCluster* performs clustering on the terms of a given document. It groups similar terms within the same cluster based on their lexical, semantic, and statistical information, and each resulting cluster may implicitly represent a topic of the document. Terms that are close to the centroids of specific clusters are selected as seeds to ensure finding candidate phrases in the document that semantically cover its main theme. Finally, *SemCluster* refines the candidate phrases and the resulting candidates are chosen as keyphrases.

The rest of the paper is organized as follows: a summary of related work is presented in Sect. 2, the *SemCluster* details are provided in Sect. 3. In Sect. 4, we undertake the evaluation and analysis of *SemCluster* performance, and, finally, we conclude the work and discuss future work in Sect. 5.

2 Related Work

A plethora of approaches for unsupervised keyphrase extraction have been devised, of which the simple ones rely fully on frequency statistics, such as TF-IDF, while others explore more sophisticated techniques. Steier and Belew [7] use mutual information statistics to discover bi-gram keyphrases in a document. However, the keyphrases it identifies can be of unpredictable length. Barker and Cornacchia [8] propose a simple system that selects only noun phrases as keyphrases without considering that some nouns carry minimal semantic significance. Litvak and Last [9] propose HITS, a graph-based ranking algorithm that extracts keyphrases from scientific documents and selects top-ranking vertices of the graph as keyphrases. Mihalcea and Tarau [5] present TextRank, a graph-based algorithm that uses the concept of Voting [10] between keywords and selects those with the most votes as keyphrases. Tsatsaronis et al. [11] present SemanticRank, a graph-based algorithm similar to TextRank, where edges are weighted based on their semantic relatedness, and nodes linked to heavily weighted edges are considered to be keyphrases. SemanticRank was tested using the *Inspec abstracts dataset*, and according to its authors has better performance than many other variations of PageRank and HITS. Bracewell et al. [12], on the other hand, propose a model that extracts noun phrases from a document, groups them into clusters in which two noun phrases are considered close if they share one or more noun terms, and, depending on their frequencies, ranks the clusters and selects the top ones in each clusters as keyphrases. Finally, KeyCluster is a similar clustering-based method proposed by Liu et al. [6]. It extracts single noun terms, groups them into clusters based on their semantic relatedness using Wikipedia and co-occurrence similarity measures, and selects phrases that contain one or more cluster centroids, and that follow a certain linguistic pattern, as keyphrases. KeyCluster outperforms many prominent AKE methods but suffers several practical drawbacks. Clustering-based methods, in general, cannot guarantee that all generated clusters are important in representing the document theme, and selecting the centroid of an unimportant cluster as a heuristic to identify and extract keyphrases leads to inappropriate keyphrases.

3 SemCluster Overview

With O as its ontology, *SemCluster* takes a free-text document D as input. Using O, *SemCluster* extracts from D keyphrases that are most representative of the theme of D. The *SemCluster* workflow is detailed in the following subsections.

3.1 Candidate Term Extraction and Disambiguation

The Term Extraction and Disambiguation stage is the first step of *SemCluster*. It performs text preprocessing on D involving tokenization, sentence boundary detection, part-of-speech (POS) tagging, and chunking. Penn Treebank notion is adopted for chunking and Part-Of-Speech (POS) tagging. The aim of chunking is to group words into chunks based on their discrete grammatical meanings. It is observed that the majority of manually-assigned keyphrases are typically embedded in noun phrases (NP chunks) [6,8,13]. Thus, *SemCluster* considers only NP chunks to find keyphrases, and detects and extracts terms in each NP chunk based on their POS annotations. We initially allow the selection of n-gram terms (where $0 < n \leq 5$) using the following POS patterns:

Table 1. Candidate term POS extraction patterns with examples from Fig. 1.

Pattern	Example
$\mathtt{N} = (\mathrm{NN}\mid\mathrm{NNS})$	dash/NN, prize/NN, drugs/NNS
$\mathtt{C} = (\mathrm{JJ}) * (\mathrm{NN}\mid\mathrm{NNS})+$	anabolic/JJ steroid/NN, gold/NN medal/NN
$E = (\mathrm{NNP}\mid\mathrm{NNPS}) * (\mathcal{S}) * (\mathrm{NNP}\mid\mathrm{NNPS})+$	Stanozolol/NNP, Ben/NNP Johnson/NNP, Olympic/NNP Games/NNPS

In Table 1, $\mathcal{N}$ denotes *Noun*, a singleton word tagged as (NN) or (NNS). $\mathcal{C}$ denotes *Compound Noun*, a sequence of words starting either with an adjective (JJ) or noun (NN and NNS). E denotes *Entity*, a sequence of words of proper nouns (NNP or NNPS) with at most one stop-word ($\mathcal{S}$), such as *the* and *of*, in the middle. Each term extracted using these patterns is mapped into *SemCluster*'s ontology O, and, depending on the mapping result, a term is regarded either as a *candidate term* or *miscellaneous*. When a term does not map to any entries in the ontology, it is decomposed into smaller constituents to be mapped again. The terms that fail to find matches even after being reduced to smaller constituents are discarded.

SemCluster's core ontology is WordNet [14], a widely used lexical database. WordNet has four lexical categories: nouns, verbs, adjectives and adverbs, and we use only the noun category. Nouns of equivalent meaning are grouped into synsets. Each synset consists of a list of synonyms and its defining gloss. Synsets are connected to other synsets by means of semantic relations. Synsets are organised into hyponym/hypernym (Is-A), and meronym/holonym (Part-Of) relationships, providing a hierarchical tree-like structure.

In practice, no knowledge base is comprehensive, and neither is WordNet. WordNet contains a *limited* number of English nouns collected over a decade ago and does not support newly emerged nouns. To compensate for this, we design a novel procedure to integrate external ontology-based knowledge bases, such as DBPedia,[1] Yago,[2] and other *ad hoc* ontologies. The workflow of the

[1] http://www.dpedia.org.

[2] http://yago-knowledge.org.

proposed procedure is as follows: for a required external knowledge base, its schema is modeled as an external ontology, and each entry in the knowledge base is assigned to one or more type classes. To perform a meaningful integration, the external ontology is horizontally aligned with *SemCluster*'s core ontology by mapping each type class to its exact or semantically equivalent synset in the core ontology. Such alignment is a one-to-one mapping where each class is mapped to exactly one synset. When *SemCluster* encounters a term that does not exist in the core ontology, the term is queried against the external knowledge base(s). If matches are found, each matching entry is retrieved from the knowledge base and considered as an external contextual meaning (or *sense*) for the given term. All external ontology classes associated with external senses are mapped into their corresponding WordNet synsets and are considered as hypernyms. The synset that corresponds to the deepest class in the external ontology's hierarchy is considered the direct hypernym of the external sense. With this construct, we allow *SemCluster* to dynamically generate appropriate senses for new terms that are absent in WordNet, and expand the set of synsets for an existing term. To illustrate, we consider integrating DBPedia and aligning all its schema classes[3] with their equivalent WordNet synsets. For instance, the class "*dbp:Athlete*" in DBPedia is directly mapped to "*wn:Athlete#n1*" in WordNet, while "*dbp:MusicFestival*" is mapped to its equivalent synset "*wn:Fete#n2*". Revisiting Fig. 1, we see that the entity "*Ben Johnson*" has no entries in WordNet but five entries in DBPedia. Accordingly, *SemCluster* generates five new senses for "*Ben Johnson*", each matching one entry in DBPedia. The third sense, ("*Ben Johnson (Sprinter)*"), is associated with four classes: "*owl:Thing*", "*dbo:Agent*", "*dbo:Person*", "*dbo:Athlete*". The deepest among the four classes, "*dpo:Athlete*", becomes the hypernym of the third sense and is referred to as "*wn:Athlete#n1*". After mapping each extracted term against the extended ontology $O^{\dagger}$, only a subset of the terms are regarded as candidate terms. We denote the set of the candidate terms by T_D. Due to pattern-based term extraction, especially when D contains informal text, T_D may harbour noisy terms that can adversely affect similarity computation and clustering performance. Noisy terms are nouns with no semantic value (e.g. "*one*", "*someone*"). To identify and remove noisy terms, *SemCluster* maps each term in T_D to a list of the most frequent noisy terms in the English language, and any term found in the list is removed from T_D.

Prior to semantic similarity computation, *SemCluster* must identify the contextual meaning (or *sense*) of each term in T_D. Word Sense Disambiguation (WSD) is an NLP task that concerns giving machines the ability to computationally determine which sense of a term is activated by its use in a particular context. WSD approaches are generally divided into three categories [16]: supervised, unsupervised, and knowledge-based. *SemCluster* employs the SenseRelate-TargetWord algorithm [17] for term sense disambiguation. SenseRelate-TargetWord is a WordNet-based algorithm implemented in WordNet::Similarity, a widely used package in computational linguistics.

[3] DBpedia schema is available at http://mappings.dbpedia.org/server/ontology/classes/.

SenseRelate-TargetWord takes one *target* candidate term as input and outputs a synset for that term based on information about the target candidate term and a few other candidate terms surrounding the target. The surrounding candidate terms are called the context window. Let t_i be a target candidate term, $t_i \in T_D$, let the size of the context window be N, and let the set of surrounding candidate terms in the context window be W, $W = \{w_1, w_2, \ldots, w_N\}$, where, if $|w| < N$, then $N = |W|$. Each t_i has one or more senses, and each sense is assigned to a different synset in WordNet, and thus we denote the senses of the term t_i by $Sense(t_i) = \{s_{i1}, s_{i2}, \ldots, s_{im}\}$, where m is the number of related synsets. Through corresponding synsets, we obtain not only the synonym list and defining gloss of the synset specifying s_{ij}, but also the synonym lists and glosses of other synsets that are related to the synset and its sense s_{ij} via the following set of semantic relations: {Hypernym, Hyponym, Meronym, Holonym}. The goal of the algorithm is to find the synset responsible for s_{ij} whose synonyms and gloss content maximises the string-based overlap score for each w_k in the context window.

3.2 Candidate Terms Similarity Computation and Clustering

After disambiguating all candidate terms in T_D, we assume each $t_i \in T_D$ is associated with the specific information: a POS tag, its position in the document, and a pointer pointing correctly to a synset s_i associated with t_i. Once the synsets of candidate terms are determined, *SemCluster* computes the pairwise semantic similarity between candidate terms based on their synset pointers. There exist many similarity measures between synsets and they may be devided into three categories [18]: path-length based, information-content based, and feature based. Unlike the other two, path-length measures offer greater flexibility to compute the similarity between synsets based on *SemCluster*'s integrated ontology. The WuPalmer measure [19] is a prominent path-length measure to compute semantic similarity between two synsets s_i, s_j by finding the shortest path between each synset and the deepest common parent synset (*Least Common Subsumer* (LCS)). The similarity $S(s_i, s_j)$ is quantified by counting the nodes in the shortest path relative to LCS depth in the ontology hierarchy. The measure is given as [19]:

$$S(s_i, s_j) = \frac{2d}{L_{s_i} + L_{s_j} + 2d} \tag{1}$$

where d is the depth of LCS from the root node, L_{s_i} is the path length from s_i to LCS, and L_{s_j} is the path length from s_j to LCS. We modify the WuPalmer algorithm to capture extra semantic similarity between s_i and s_j. Path length measures in general, and WuPalmer in particular, focus on measuring synset similarities by exploiting the explicit semantic relations existing between them. However, WordNet does not cover all possible relations that may exist between synsets. For example, there is no direct link between "*wn:Bush#n4*" and "*wn:President#n2*", although they are clearly related if they co-occur in a document. To capture explicit as well as implicit semantic similarities between s_i and s_j, we extend the WuPalmer measure as follows:

$$S(s_i, s_j) = \frac{2d + \mathit{Overlap}(C(s_i), C(s_j))}{L_{s_i} + L_{s_j} + 2d + \mathit{Overlap}(C(s_i), C(s_j))} \tag{2}$$

where $C(s_i)$ and $C(s_j)$ are functions that retrieve s_i and s_j information from WordNet in string format, and $\mathit{Overlap}(C(s_i), C(s_j))$ is a function that measures the string-based overlap between $C(s_i)$ and $C(s_j)$. Let $\mathit{Synonyms}(s_i)$ be a function that retrieves all the words in the synonym list of the synset s_i, $\mathit{Gloss}(s_i)$ be a function that retrieves the definition of s_i, $\mathit{Related}(s_i)$ be a function that retrieves the synonyms and definitions of all synsets connected directly to s_i via the relation set $\{Hypernym, Hyponym, Meronym, Holonym\}$, then $C(s_i)$ is defined as follows:

$$C(s_i) = \mathit{Synonyms}(s_i) \cup \mathit{Gloss}(s_i) \cup \mathit{Related}(s_i) \tag{3}$$

where $\cup$ is the string concatenation function. $\mathit{Overlap}(C(s_i), C(s_j))$ finds the maximum number of words shared in the output of $C(s_i)$ and $C(s_j)$ normalized by natural logarithm to prevent too great an effect of implicit semantic similarity on the WuPalmer explicit semantic similarity measurement. Thus, we define *Overlap* as follows:

$$\mathit{Overlap}(C(s_i), C(s_j)) = \log\left(C(s_i) \cap C(s_j) + 1\right) \tag{4}$$

The extended WuPalmer measure is used to compute the pairwise similarities between each pair of terms in T_D, and the result is a complete adjacency similarity matrix of size $|T_D| \times |T_D|$ denoted by A. Once we have produced A, we move on to the second phase of the step - terms clustering. There are many state-of-the-art clustering algorithms to cluster T_D efficiently. Affinity Propagation (AP) [20] is proposed as a powerful technique for exemplar learning by passing messages between nodes. It is reported to find clusters with much lower error compared to other algorithms and does not require specifying the number of desirable clusters in advance. Both merits are extremely important for *SemCluster* to support fully automated keyphrase extraction and hence AP is used as the clustering algorithm in *SemCluster*. The input to AP is the matrix A. The set T_D is modelled as a graph with nodes t_i, $t_i \in T_D$. An edge between t_i and t_j exists if $S(t_i, t_j) > 0$ and the weight of the edge is given by element $A[i][j]$. Initially, all the nodes are viewed as exemplars, and after a large number of real-valued information messages have been transmitted along the edges of the graph, a relevant set of exemplars and corresponding clusters is identified. In AP terms, the similarity $S(t_i, t_j)$ indicates how much t_j is suitable to be exemplar of t_i. In *SemCluster*, $S(t_i, t_j) = A[i][j]$, $i \neq j$. If there is no heuristic knowledge, self-similarities are called *preferences*, and are set as constant values. In *SemCluster*, the preference $S(t_i, t_i)$ is computed using the median. In AP two kinds of messages are exchanged between nodes: *responsibility* and *availability*. A responsibility message is sent from node t_i to candidate exemplar t_j, and reflects the accumulated evidence for how well-suited t_j is to serve as the exemplar for t_i. An availability message is sent from candidate exemplar t_j to t_i, and reflects the accumulated evidence for how well-suited it would be for t_i to choose t_j as its exemplar. At the beginning, all availabilities are initialised to zero, and during m iterations, both responsibility and availability messages are updated iteratively until they remain constant for a specific number of iterations, and then both responsibilities and availabilities are combined to discover exemplars [20]. Eventually, every term in T_D is annotated with its exemplar. The number of clusters and other clustering information are directly obtained by grouping terms based on their shared exemplars. At start-up, the input set T_D can be *redundant* to reflect not only the semantic and lexical information of each term t_i, but also the influence of its frequency information on the clustering results, such that, if the term t_i is highly frequent in the document, its frequency can be a reason to qualify as an exemplar on the condition that t_i is always allocated the same sense s_{ij} in all its occurrences in D. Typically, clustering-based AKE approaches use cluster centroids as seeds [6,12], and any phrase in D containing one or more centroids is chosen as a keyphrase. From our empirical observation, we suggest that direct selection of centroids resulting from AP or similar algorithms may lead to poor keyphrase extraction recall and/or precision, due to the following two reasons:

Theme-Independent Seed Selection. Clustering-based methods assign equal importance to all cluster centroids [2]. Thus, a phrase containing a centroid of an unimportant cluster is ranked exactly equivalent to a phrase containing a centroid of an extremely important cluster relative to the document theme [21]. Consequently, there is no guarantee that the extracted keyphrases are the best representative phrases. Our solution to this is to discard irrelevant or marginally related clusters and keep the most relevant ones. The solution is largely based on the observation that clusters that sufficiently cover the document theme tend to be semantically more related to each other than irrelevant or marginally related clusters. Regarding AP, the exemplar is the best representative of its clusters semantics. Therefore, we assess the average of semantic relatedness strength of each exemplar against all other exemplars, and any cluster whose exemplar exhibits *weak* semantic relatedness is removed. Let C_D be the set of clusters resulting from clustering T_D, $C_D = \{C_1, C_2, \ldots, C_N\}$ where $N = |C_D|$. For each cluster C_i, we compute its exemplar's average semantic relatedness, $Ave(\varepsilon_i)$, as follows:

$$Ave(\varepsilon_i) = \frac{\Sigma_{i \neq j} SR(\varepsilon_i, \varepsilon_j)}{N-1}, \quad N > 1 \tag{5}$$

In the above definition, $SR(\varepsilon_i, \varepsilon_j)$ measures the semantic relatedness between the exemplars of two clusters C_i, C_j. Each cluster C_i is ranked based on its exemplar average score, and it will be removed from C_D if its average score, $Ave(\varepsilon_i)$, is below the average of all clusters. $SR(\varepsilon_i, \varepsilon_j)$ is similar to $S(\varepsilon_i, \varepsilon_j)$ and measures the semantic relatedness between ε_i and ε_j. For instance, the terms "*drug*" and "*Olympics*" are not similar, but, because of their tendency to occur together ("*drug use*" appears frequently in "*Olympics*" themes), they are judged to be semantically related. To quantify such relatedness in an unsupervised cross-domain environment, we are expanding *SemCluster* to take advantage of *Wikipedia*, the largest and fastest growing knowledge base. There is a selection of approaches that measure semantic relatedness by exploiting Wikipedia. *Explicit Semantic Analysis* [22] is one of the most accurate Wikipedia-based measures that, to a great extent, comes close to the accuracy of a human [23] and, hence, is employed by *SemCluster* to compute semantic relatedness.

Search Space Restriction. The sole reliance on the clusters centroids may lead to restricting the search space of finding the best representative phrases in a given document and, consequently, result in degrading keyphrase extraction recall and/or precision. Suppose we have a valid keyphrase containing a term t_j that is semantically close to a centroid term ε_i. The phrase will not be extracted simply because t_j is not a centroid. This may explain why spectral clustering outperforms AP clustering in KeyCluster experiments - the former allows multiple terms close to a cluster centroid to be chosen as seeds and accordingly extends the keyphrase search space. Taking advantage of this observation, *SemCluster* expands the selection of seeds from AP clustering in a fashion similar to that of spectral clustering. Let C'_D be the final set of clusters resulting from clustering T_D using AP after centroid relatedness average ranking, where $C'_D \subseteq C_D$, $C'_D = \{C_1, C_2, \ldots, C_k\}$. For each cluster C_i, $i \leq k$, we select its exemplar ε_i as a seed. We regard each member t_j in C_i ($t_j \neq \varepsilon_i$) as an additional seed if $S(\varepsilon_i, t_j) \geq \tau$, where $S(\varepsilon_i, t_j)$ is the computed score stored in A from a previous step (see Sect. 3.1) and τ is a predefined distance threshold specifying how semantically close t_j should be to the centroid ε_i in order to qualify as a seed. We repeat this procedure for all the clusters in C'_D to obtain a set of appropriate seeds from the extended search space.

3.3 Candidate Phrase Extraction and Keyphrase Selection

After selection of the seeds, each chunk NP_i in D is scanned, and *SemCluster* extracts any sequence of words inside NP_i that satisfies two conditions: (i) containing a seed, (ii) matching an *Extraction POS Pattern* by the following rules. (i) If NP_i contains a seed extracted using an E-pattern, the seed is regarded as a candidate phrase. (ii) If NP_i contains a seed extracted using a $\mathcal{C}$-pattern, two cases are considered: if the seed starts with (JJ), the sequence matching pattern $(\mathcal{C}) *$ (NN|NNS)+ is extracted from NP_i; if the seed starts with (NN), the sequence matching pattern (JJ) $* (\mathcal{C})+$ is extracted from NP_i. (iii) Finally, if NP_i contains a seed extracted using a $\mathcal{N}$-pattern, the sequence matching pattern (JJ) $* (\mathcal{N})+$ is extracted from NP_i. Each extracted sequence using these POS patterns is called a candidate phrase. The final step in the *SemCluster* processing approach is to refine the set of extracted candidate phrases. The refining step starts by pruning redundant candidate phrases. Two or more candidate phrases may be semantically equivalent but exist in different forms. They may be synonymous phrases, e.g. both "*Olympics*" and "*Olympic Games*" belong to the same synset in WordNet; or adjective-synonymous phrases, e.g., the Wikipedia article referring to "*Bernard Madoff*" contains phrases that share an important seed "*fraud*" such as "*financial fraud*", "*gigantic fraud*", "*massive fraud*". In this case, we keep the first occurring candidate phrase and remove the others. There is also the case of subphrases, as in the example of "*Johnson*" and "*Ben Johnson*". Both phrases contain "*Johnson*", so we keep the longer phrase, which is more specific, and discard the short one.

4 Evaluation and Results

To evaluate *SemCluster*, we use an aligned ontology resulting from integrating WordNet with the DBPedia schema.[4] The external knowledge sources that we exploit for semantics-related computations are DBPedia lookup-server[5] for extending the core ontology semantic coverage, and EasyESA[6] for ESA-based semantic relatedness measurement. *SemCluster* performs text preprocessing using OpenNLP[7] with models trained on huge collections in multiple domains. We use WordNet[8] with some slight modifications to accommodate our needs, including re-indexing synsets for faster access, modifying each synset's gloss to keep only nouns and adjectives, and lemmatizing all noun tokens.

The dataset used in *SemCluster* evaluation is the *Inspec dataset*,[9] which is frequently used in the evaluation of AKE methods [5,6,13]. This is a collection of abstracts of scientific papers from the *Inspec Database*, consisting of 2000 abstracts. Each abstract is represented by three files: *.abstr*, *.contr* and *.uncontr*. The file *.abstr* contains the actual text; *.contr* contains keyphrases restricted to a specific dictionary; and *.uncontr* contains keyphrases assigned by human experts. In Hulth's experiment [13], the proposed method was supervised, and the dataset was spilt into three partitions: 1000 abstracts for training, 500 for validation, and 500 for testing. TextRank [5] and KeyCluster [6] are unsupervised methods, and thus only the test partition was used in their evaluations. Since *SemCluster* is also unsupervised, we adopt a similar approach to [5,6], and use only the test partition to provide a precise comparison with the mentioned AKE methods. In the dataset, phrases that are not in the abstract, if regarded by the human expert as suitable, are stored in *.uncontr* as keyphrases. In our evaluation, we consider only keyphrases that actually occur in the abstract. An output keyphrase is considered to be valid if it is identical to, semantically equivalent to, or is a subphrase of, a manually assigned keyphrase in *.uncontr*.

We use the *Inspec dataset* as a benchmark to compare *SemCluster*'s performance with two AKE methods: TextRank [5], a baseline unsupervised method, and KeyCluster [6], a current state-of-the-art unsupervised clustering-based method. KeyCluster is implemented using three different algorithms: Hierarchal

[4] The alignment results are available at http://www.pdlab.io/semcluster/inspec.rar.

[5] The lookup code is available at https://github.com/dbpedia/lookup.

[6] ESA code is available at http://treo.deri.ie/easyesa/.

[7] http://opennlp.apache.org.

[8] Wordnet v.3.1 is available at https://wordnet.princeton.edu/wordnet/download.

[9] Hulth's dataset copy is available at http://pdllab.io/semcluster/inspec.rar.

Clustering (HC), Spectral Clustering (SC), and Affinity Propagation (AP). Due to the poor performance of HC reported in [6], we evaluate KeyCluster based on SC and AP, not HC. During method comparisons, only the best results under the best possible settings, if any, for a given approach are considered. The setting for KeyCluster-SC is that m, the predefined number of clusters, is $m = \frac{2}{3}n$, where $n = |D|$. For KeyCluster-AP, the maximum number of iterations is set to 1000, the damping factor is set to 0.9, and the clustering preference is computed using *max*. *SemCluster* performs AKE in fully automatic mode. However, it requires tuning a set of parameters, which are: WSD context window size N (see Sect. 3.1), τ distance threshold (see Sect. 3.2), and AP algorithm default parameters (i.e., the maximum number of iterations M, damping factor λ, convit, and preference). From empirical observation, *SemCluster* performs the best possible WSD when $N = 10$. When $N < 5$, the performance of term sense disambiguation degrades; when $N > 10$, there is no obvious influence on the results. The default tuning of AP parameters is: $M = 1000, \lambda = 0.9$, and $convit = 50$. The custom tuning of these AP parameters produces no recognizable changes to the clustering results since the input similarities are always positive and in the range of $[0, 1]$. In *SemCluster*, the AP clustering preference is set to *median* to ensure that *SemCluster* performs clustering with higher granularity (i.e., a larger number of clusters) so that unimportant terms can be allocated to unimportant clusters and hence removed from the clustering results using ESA-based averaging.

As discussed in Sect. 3.2, τ, the distance threshold, has a direct impact on the performance of *SemCluster*. When $\tau = 1$, only the centroids of the clusters are chosen as seeds to identify and extract keyphrases; when $\tau = 0$, all the terms in T_D (except those belonging to the irrelevant clusters that have been discarded) are selected as seeds; hence most NP chunks in D are chosen as keyphrases. Considering the semantic similarity score of 0.5 as the least for which two terms can be judged semantically close, τ takes any value in the range $0.5 < \tau < 1$. Learning the optimal τ setting is a hyperparameter optimization problem that can be readily solved either by manual search or by utilizing optimization search algorithm [24]. In this experiment, we use manual search for τ estimation, and we randomly select 50 documents from Hulth's dataset to perform AKE using *SemCluster* with $\tau = 0.6, 0.7, 0.8$, and 0.9. *SemCluster* achieves the best possible performance on the selected samples when $\tau = 0.7$. The metric used for all *SemCluster* evaluations is `Precision/Recall/F1-measure`, which are defined as follows:

$$P = \frac{k_{correct}}{k_{extract}}, \qquad R = \frac{k_{correct}}{k_{standard}}, \qquad f = \frac{2 * PR}{P + R} \tag{6}$$

where $k_{correct}$ is the number of correct keyphrases extracted by *SemCluster*, $k_{extract}$ is the total number of the keyphrases extracted, and $k_{standard}$ is the total number of keyphrases assigned by human experts. Using the test partition of Hulth's dataset, *SemCluster* extracts 6974 keyphrases from 500 abstracts, among which 2737 phrases belong to the set of "gold-standard" keyphrases (i.e., $k_{standard}$). The results of the F1-measure and comparison with other methods are presented in Table 2.

Table 2. Comparison of AKE results for *SemCluster*, TextRank, and KeyCluster (SC/AP)

Method	Precision	Recall	F1-measure
TextRank	0.312	0.431	0.362
KeyCluster-SC	0.350	0.660	0.457
KeyCluster-AP	0.330	0.697	0.448
SemCluster	**0.392**	**0.721**	**0.507**

The results clearly show that *SemCluster* outperforms the other methods on both the recall and precision of keyphrases. Compared to KeyCluster-SC, the second best in performance, *SemCluster* achieves an improvement of 5% in F1-measure. Both *SemCluster* and KeyCluster-AP utilize the same clustering algorithm, but the former outperforms the latter with a 6% improvement in F1-measure. We observe that initial n-gram term selection and pruning unimportant clusters are the main contributors in boosting *SemCluster* precision. Likewise, the WordNet-based semantic representation of documents and expansion of seeds have a positive impact on keyphrase recall. According to the best of our knowledge, *SemCluster*'s F1-measure score of 0.507 using Hulth's dataset is the highest among current state-of-the-art unsupervised clustering-based methods.

SemCluster loads Wordnet directly into its memory (occupying about 22 MB) to give fast access to the semantics of any term in D. *SemCluster* accesses external knowledge bases only when a given term is not found in WordNet, which, compared with KeyCluster's crawling of about 50 million Wikipedia articles for every item to construct its conceptual vector, results in a significant improvement in performance.

5 Conclusion and Future Work

In this paper, we have introduced *SemCluster*, a clustering-based unsupervised keyphrase extraction method. By integrating an internal ontology (i.e., WordNet) with external knowledge sources, *SemCluster* identifies and extracts semantically important terms from a given document, clusters the extracted terms, identifies the most representive phrases and select among them the keyphrases via post-processing of the clustering results. The evaluation results verify the findings of Liu et al. [6] that unsupervised clustering-based AKE methods are effective and robust.

SemCluster is a part of a larger project specializing in big data processing called the Personal Data Lake (PDL) [15]. Though *SemCluster* is a general tool for extracting keyphrases from unstructured text, PDL is the main motivation behind it. PDL requires a domain-agnostic AKE tool to automatically process documents ingested from heterogenous data providers, and extract their keyphrases with the best possible precision and computational efficiency, and thus generates semantic metadata for free-text data and allows them to interrelate on a micro-semantic level.

Although SemCluster exhibits good performance, there is still room for improvement. In an experiment on a collection of informal texts collected online from social networks and news websites, in which we replaced WuPalmer [19] with Jiang-Conrath [25] and used Babelfy [26] for term sense disambiguation, we observed an approximate 7% improvement in the F1-measure. However, computational efficiency decreased because Babelfy is an on line service. Nevertheless, this suggests a potential improvement to *SemCluster*, particularly by improving its semantic similarity measurement and word sense disambiguation.

Clustering, like all other clustering-based approaches, is at the heart of *SemCluster*. Improving the clustering performance improves the overall performance of *SemCluster*. Recently we became aware of a relatively new approach, an extension of Affinity Propagation, called AP clustering with seeds [27], which is reported to outperform the original AP algorithm. Testing this extension, or some modified version, on *SemCluster* over a variety of testing datasets is also one of our immediate undertakings in the future.

References

1. Turney, P.D.: Learning algorithms for keyphrase extraction. Inf. Retr. **2**(4), 303–336 (2000)
2. Hasan, K.S., Ng, V.: Automatic keyphrase extraction: a survey of the state of the art. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 1262–1273 (2014)
3. Washio, T., Motoda, H.: State of the art of graph-based data mining. ACM SIGKDD Explor. Newsl. **5**(1), 59–68 (2003)
4. Sonowane, S.S., Kulkarni, P.A.: Graph based representation and analysis of text document: a survey of techniques. Int. J. Comput. Appl. **96**, 1–8 (2014)
5. Mihalcea, R., Tarau, P.: TextRank: bringing order into texts. Association for Computational Linguistics (2004)
6. Liu, Z., Li, P., Zheng, Y., Sun, M.: Clustering to find exemplar terms for keyphrase extraction. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pp. 257–266 (2009)
7. Steier, A.M., Belew, R.K.: Exporting phrases: a statistical analysis of topical language. In: Second Symposium on Document Analysis and Information Retrieval, pp. 179–190 (1993)
8. Barker, K., Cornacchia, N.: Using noun phrase heads to extract document keyphrases. In: Conference of the Canadian Society for Computational Studies of Intelligence, pp. 40–52. Springer (2000)
9. Litvak, M., Last, M.: Graph-based keyword extraction for single-document summarization. In: Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization, pp. 17–24 (2008)
10. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the web. Technical Report, Stanford InfoLab (1999)
11. Tsatsaronis, G., Varlamis, I., Nrvg, K.: SemanticRank: ranking keywords and sentences using semantic graphs. In: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 1074–1082 (1999)

12. Bracewell, D.B., Ren, F., Kuriowa, S.: Multilingual single document keyword extraction for information retrieval. In: Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering, pp. 517–522 (2005)
13. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, pp. 216–223. Association for Computational Linguistics (2003)
14. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.J.: Introduction to WordNet: an on-line lexical database. Int. J. Lexicogr. **3**(4), 235–244 (1990)
15. Alrehamy, H., Walker, C.: Personal data lake with data gravity pull. In: Proceedings of the 2015 IEEE Fifth International Conference on Big Data and Cloud Computing, pp. 160–167 (2015)
16. Navigli, R.: Word sense disambiguation: a survey. ACM Comput. Surv. **41**(2) (2009)
17. Patwardhan, S., Banerjee, S., Pedersen, T.: SenseRelate::TargetWord: a generalized framework for word sense disambiguation. In Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions, pp. 73–76. Association for Computational Linguistics (2005)
18. Meng, L., Huang, R., Gu, J.: A review of semantic similarity measures in WordNet. Int. J. Hybrid Inf. Technol. **6**(1), 1–12 (2013)
19. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, pp. 133–138. Association for Computational Linguistics (1994)
20. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science **315**(5814), 972–976 (2007)
21. Liu, F., Pennell, D., Liu, F., Liu, Y.: Unsupervised approaches for automatic keyword extraction using meeting transcripts. In: Proceedings of Human Language Technologies, pp. 620–628. Association for Computational Linguistics (2009)
22. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 1606–1611 (2007)
23. Witten, I., Milne, D.: An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In: Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy, pp. 25–30. AAAI Press, Chicago (2008)
24. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**(2), 281–305 (2012)
25. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint http://arxiv.org/abs/cmp-lg/9709008 (1997)
26. Moro, A., Cecconi, F., Navigli, R.: Multilingual word sense disambiguation and entity linking for everybody. In: Proceedings of the 2014 International Conference on Posters and Demonstrations, pp. 25–28. CEUR-WS.org (2014)
27. Guan, R., Shi, X., Marchese, M., Yang, C., Liang, Y.: Text clustering with seeds affinity propagation. IEEE Trans. Knowl. Data Eng. **23**(4), 627–637 (2011)

Control and Human-Machine Systems

Towards Low-Cost P300-Based BCI Using Emotiv Epoc Headset

Xiangqian Liu[1,2], Fei Chao[1,2], Min Jiang[1,2], Changle Zhou[1,2],
Weifeng Ren[1,2], and Minghui Shi[1,2](✉)

[1] Department of Cognitive Science,
School of Information Science and Engineering,
Xiamen University, Xiamen 361005, China
smh@xmu.edu.com

[2] Fujian Key Laboratory of Brain-inspired Computing Technique and
Applications, Xiamen University, Xiamen 361005, China

Abstract. P300-based brain-computer interface (BCI) has been widely studied over two decades. However, there are several factors that hamper P300-based BCI to be used in daily life. EEG acquisition devices are often too much expensive for an average customer. Although the Emotiv Epoc headset is a kind of low-cost device for recording brain signals and has been adopted to develop some BCI systems, due to the limited number of electrodes, the Emotiv Epoc headset cannot cover the regions of scalp that are convenient for detecting P300, so the effectiveness of the Emotiv Epoc headset used in the P300-based BCI has been doubted by many researchers. This paper aims to examine the performance of Emotiv Epoc headset used in the P300-based BCI system. Six participants participated in the experiment and two paradigms were compared. The results demonstrated that P300 could be effectively detected from the brain signals recorded by the Emotiv Epoc headset, showing the promising future to develop low-cost P300-based BCI systems.

Keywords: Brain computer interface · Emotiv Epoc headset · P300

1 Introduction

A brain-computer interface (BCI) can be defined as a system that can allow users to communicate with the surrounding environment directly by their brain without using their muscles [1]. In a BCI system, some kind of particular patterns in brain activities are induced by external stimuli or internal mental activities. On the other hand, by identifying the pattern in the recorded brain signals, a BCI system can infer the corresponding external stimuli or internal mental activities that induce the pattern. In this way, the users can communicate with the surrounding environment by means of BCI systems.

To meet various applications, diverse BCI systems have been developed in recent years, among which the most common one is the electroencephalogram (EEG)-based BCI system [2]. EEG is a general term for various electrical activities induced by cortical neurons. It can be recorded through a set of electrodes placed on the scalp, and

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_20

has the advantage of safety, convenience, and high temporal resolution compared with other brain imaging methods such as fMRI, MEG, ECoG, etc.

P300 is one kind of typical event-related potential (ERP), discovered by Sutton et al. in 1965 [3]. P300 appears in the brain signals of a subject around 300 ms after the occurrence of an odd-ball event (small probability event) attended by the subject. During the running period of a P300-based BCI system, a series of events (e.g. character flashes in a spelling system) are presented in the user interface. The user pays attention to one of the events while ignoring others. The events attended by the user are expected to induce the P300 pattern in the user's brain signals. Therefore, by recognizing P300, which acts as the characteristic of the brain signals, the BCI system can identify which event is attended by the user. Furthermore, if the events are deliberately associated with some kinds of commands or intentions, the user can utilize the P300-based BCI system to execute some commands or express intentions just by paying attention to a particular odd-ball event that is associated with the user's desire commands or intentions.

Currently P300-based BCI systems have been applied in many fields, such as traffic light recognition [4], cursor control [5], robot manipulation [6], or lie detection [7]. There are still several challenges to be resolved such as the high cost of EEG acquisition device. Fortunately, some consumer-grade EEG headsets have emerged on the market recently, such as ThinkGear (NeuroSky, Inc), Enobio (Starlab, Inc), BR8+ (Brain Rhythm, Inc), Emotiv Epoc (Emotiv systems, Inc), etc.

Currently the effectiveness of the Emotiv Epoc adopted in P300-based BCI systems is doubted by many researchers, since the Emotiv Epoc headset has limited number of electrodes, and cannot cover some scalp parts, such as Fz, Cz, and Pz in the International 10–20 system (Fig. 1) [8], that are convenient to detect P300.

This paper aims to examine the performance of Emotiv Epoc headset adopted in P300-based BCI systems. The remainder of this paper is organized as follows. Section 2 presents the experiment process, then the results are discussed in Sect. 3, and finally Sect. 4 makes a conclusion.

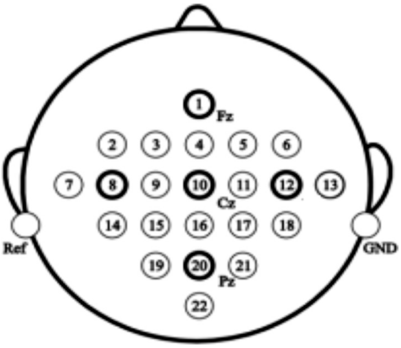


Fig. 1. International 10–20 system.

2 Methodology

2.1 Participants

Six postgraduate students (4 males and 2 females, 23–25 years old) at Xiamen University participated in the experiment. They were able-bodied, with normal or corrected-to-normal vision, and with no history of neurological diseases. Two of them had BCI experience, but not with a P300-based BCI. Each participant read and signed an informed consent prior to the experiment.

2.2 EEG Equipment

In this study, an Emotiv Epoc headset was used to record the brain signals of the users. The Emotiv Epoc headset is wireless, and mainly includes 14 channels (plus two references, CMS/DRL), each based on a saline sensor. Figure 2 depicts these channels in the standard 10–20 system. In our experiment, all of the channels were used, and the electrode impedance was kept below 10 kΩ. The sampling rate is 128 Hz.

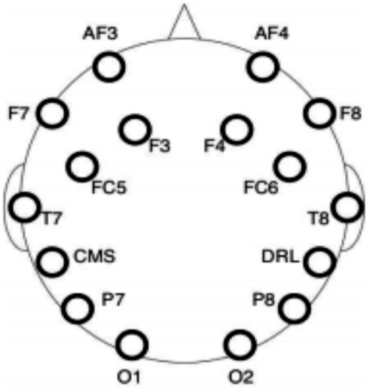


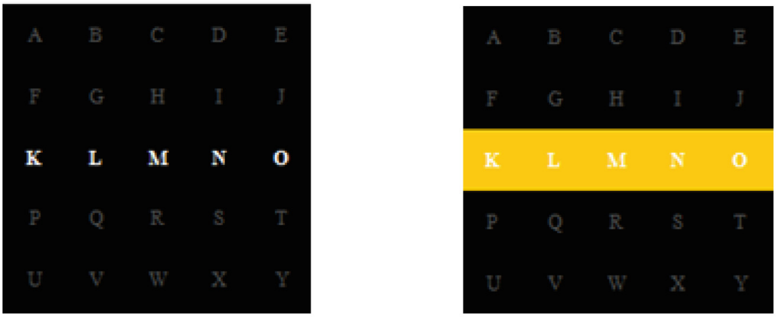
Fig. 2. Distribution of Emotiv Epoc electrodes.

2.3 Stimuli and Procedure

During the experiment, participants were comfortably seated in a quiet room in front of a computer screen. They were required to remain relaxed and avoid unnecessary movements. Figure 3 shows the user interface presented to the participants, which was a 5 × 5 matrix with 25 grey English letters against a black background. Two distinct paradigms of stimulus presentation were studied in the experiment, both of which were based on Farwell and Donchin's Row/Column speller [9].

Two paradigms were conducted and compared. The rows or columns of letters are intensified randomly one by one, either by changing only the foreground color (Paradigm I, see Fig. 3(a)), or by changing both the foreground color and the background color (Paradigm II, see Fig. 3(b)).

All participants completed two experimental sessions in two days, each for one paradigm. We use the term “subtrial” that refers to as one row/column intensification. Each subtrial lasted for 250 ms—the intensification itself was displayed for 150 ms, followed by a 100 ms blank before the next intensification began. One trial contained 10 different subtrials, i.e., each row and each column of the matrix was intensified once in a trial. Fifteen trials form a run. During each run, the participant’s task was to focus on a particular letter, and silently counted each time it was intensified. There was a 1.5-s interval between two consecutive runs, and each session was consisted of 18 runs.



(a) ParadigmI (b) ParadigmII

Fig. 3. User interfaces of two respective paradigms.

2.4 Data Analysis and Classification

The EEG data recorded from one participant was excluded from the further analysis because the classifier cannot detect a reliable P300 response in his EEG. Thus, the final sample consisted of 5 participants (3 males and 2 females).

In this study, MATLAB (version 8.0.0, R2012b) and EEGLAB (version 12.0.0b) were used to analyze the recorded brain signals.

The second-order Butterworth filter was first used to band-pass filter the raw EEG between 0.1 Hz and 30 Hz. Then the filtered EEG was downsampled by a factor of 4 in order to reduce the computational complexity. After that, the 1200-ms segments from the beginning point of each subtrial were extracted from the downsampled data. To ensure a reliable artifact rejection, all the segments containing amplitudes exceeding ±70 µV were removed. The remaining segments were averaged according to the type of subtrials to form the feature vectors for each run.

Support vector machine (SVM) was used as the classification algorithm. SVM seeks to find a classification hyperplane that separates two classes optimally; details can be seen in [10]. Previous studies have demonstrated that SVM could provide good performance in P300 recognition [11]. The SVM classifier was trained in the two paradigms, using the data obtained from the first four runs of each session. The data of the remaining runs were used as the testing data.

3 Results and Discussions

We defined classification accuracy as the percentage of feature vectors (from the testing data) that are correctly classified by SVM.

In addition, to examine the amplitudes of the evoked P300, P300 amplitude was calculated by the peak-to-peak method [12], i.e., P300 amplitude was calculated by subtracting the negative peak from the positive peak. The positive peak is the maximum average positive amplitude over one 100 ms segment contained in the 250–500 ms period after the time when the odd-ball event is onset, while the negative peak is the maximum average negative amplitude over one 100 ms segment contained in the 700 ms following the time corresponding to the positive peak.

Table 1 shows the classification accuracy and the P300 amplitudes for the two paradigms.

Table 1. Individual classification accuracy and P300 amplitude.

Participants	Accuracy (%)		P300 amplitude (μV)	
	Paradigm I	Paradigm II	Paradigm I	Paradigm II
S1	74.29	78.57	4.76	5.02
S2	78.57	80.71	4.04	4.72
S3	72.86	69.29	3.91	4.33
S4	76.43	79.29	4.38	4.89
S5	75.71	77.14	5.26	5.72
Mean	75.57	77.01	4.47	4.94

The mean accuracy of Paradigm I and Paradigm II reached 75.57% and 77.01% respectively. It can be observed that an enhancement of P300 amplitude in Paradigm II compared with that in Paradigm I. This fact may be the reason for the higher classification accuracy of Paradigm II, and is in line with the viewpoint that the more visually salient stimulus pattern may give rise to higher amplitude of P300 and increase the classification accuracy of P300-based BCI systems [13].

4 Conclusion

Aiming to develop low-cost P300-based BCI systems, this paper investigated the performance of the Emotiv Epoc headset adopted in the P300-based BCI systems. The experiment was conducted under two paradigms with six healthy participants. Results showed that the Emotiv Epoc headset can be used to develop low-cost P300-based BCI although it has limited number of electrodes. Our further research will design improved paradigms and develop novel P300-based BCI applications using the Emotiv Epoc headset.

Acknowledgment. The project was supported by the Natural Science Foundation of Fujian Province of China (Nos. 2017J01128 and 2017J01129) and by the National Natural Science Foundation of China (Nos. 61673322 and 61673328).

References

1. Wolpaw, J.R., McFarland, D.J.: Brain-computer interfaces for communication and control. Clin. Neurophysiol. **113**(6), 761–791 (2002)
2. Leuthardt, E.C., Schalk, G., Wolpaw, J.R., Ojemann, J.G.: A brain-computer interface using electrocorticographic signals in humans. J. Neural Eng. **1**(2), 63–71 (2004)
3. Sutton, S., Braren, M., Zubin, J., John, E.R.: Evoked-potential correlates of stimulus uncertainty. Science **150**(3700), 1187–1188 (1965)
4. Bayliss, J.D., Ballard, D.H.: Single trial P3 epoch recognition in a virtual environment. Neurocomputing **32**(1), 637–642 (2000)
5. Wolpaw, J.R., McFarland, D.J., Neat, G.W., Fomeris, C.A.: An EEG-based brain-computer interface for cursor control. Electroencephalogr. Clin. Neurophysiol. **78**(3), 252–259 (1991)
6. Bell, C.J., Shenoy, P., Chalodhom, R., Rao, R.P.: Control of a humanoid robot by a noninvasive brain-computer interface in humans. J. Neural Eng. **5**(2), 214–220 (2008)
7. Farwell, L.A., Smith, S.S.: Using brain MERMER testing to detect knowledge despite efforts to conceal. J. Forensic Sci. **46**(1), 135–143 (2001)
8. Sur, S., Sinha, V.K.: Event-related potential: an overview. Ind. Psychiatry J. **18**(1), 70–73 (2009)
9. Farwell, L.A., Donchin, E.: Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. Electroencephalogr. Clin. Neurophysiol. **70**(6), 510–523 (1988)
10. Liu, H., Zhou, W.D., Huang, A.H.: P300 EEG recognition based on SVM approach. Chin. J. Biomed. Eng. **18**(1), 35–39 (2009)
11. Kaper, M., Meinicke, P., Grossekathoefer, U., Linger, T.: BCI competition 2003–data set IIb: support vector machines for the P300 speller paradigm. IEEE Trans. Biomed. Eng. **51**(6), 1073–1076 (2004)
12. Soskins, M., Rosenfeld, J.P., Niendam, T.: Peak-to-peak measurement of P300 recorded at 0.3 Hz high pass filter settings in intraindividual diagnosis: complex vs simple paradigms. Int. J. Psychophysiol. **40**(2), 173–180 (2001)
13. Zhongwei, M.A., Gao, S.: P300-based brain-computer interface: effect of stimulus intensity on performance. J. Tsinghua Univ. **48**(3), 415–418 (2008)

Emotion Detection in E-learning Using Expectation-Maximization Deep Spatial-Temporal Inference Network

Jiangqin Xu[1], Zhongqiang Huang[2], Minghui Shi[2], and Min Jiang[2](✉)

[1] College of Foreign Languages and Cultures, Xiamen University,
Xiamen, Fujian, China
xujiangqin@xmu.edu.cn

[2] Fujian Province Key Laboratory for Brain-inspired Computing Technique and Applications, School of Information and Engineering, Xiamen University,
Xiamen, Fujian, China
minjiang@xmu.edu.cn

Abstract. It is very useful for the E-learning systems to detect the students emotional state accurately, and this can remind the teacher in time to change the teaching rhythm or content to meet the student's emotional changes for making the teaching effect optimization. In this paper, we propose an emotion detection method based on a deep learning approach, Expectation-maximization Deep Spatial-Temporal Inference Network (EM-DeSTIN). This method takes the student's facial expression as input and combine with Support Vector Machine (SVM) to implement emotion classification and identification. Experimental results show that the proposed method improves the performance of detecting emotion in a noisy environment compared with other methods.

Keywords: E-learning · Emotion detection · Deep learning

1 Introduction

In recent years, with the Massive Open Online Courses (MOOC) and other new teaching methods emerging [1], the use of artificial intelligence technology to improve the quality of teaching attracts a considerable number of researchers interest [2,3]. In these studies, it is particularly important to detect learners emotions because the emotions have a crucial effect on the effect of learning [4,5]. It is also understood that emotions can heavily influence the knowledge and overall goals of the students [6]. Therefore, for E-learning, one of the prerequisites for effective teaching activities is that teachers are able to respond quickly to student mood changes. However, how to accurately and quickly understand the emotional interaction of students in a complex learning environment is still a very challenging task.

In many attempts, the methods which used pattern recognition techniques to detect students' emotional changes are considered an effective way [7]. For

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_21

example, in [8] the authors propose an AdaBoost based method for detecting the state of human eyes. The AutoTutor [9] research project wants to detect and utilize emotions of learners to augment the learning and teaching process. In [10], a neural network architecture is constructed to be able to handle the fusion of facial features, prosody and lexical content in speech for detecting emotion. In [11], the authors proposed a method based on convolutional neural network to recognize facial expression.

"Good features" is a vital part for an emotion-recognizing system and Deep Learning [12] has proved to be a very effective feature extraction method. Images obtained in E-learning systems often contain noise, and the EM-DeSTIN method [13] is very effective when dealing with noisy images, so the major contribution of this research is that we employ a deep learning technique to develop an emotion recognition approach which can be used to improve the quality of E-learning system.

The rest of this paper is organized as follows. In Sect. 2, we will briefly introduce Deep Spatial-Temporal Inference Network (DeSTIN). In Sect. 3, we will describe the detail of EM-DeSTIN. In Sect. 4, experimental results will be presented. Finally, Sect. 5 gives a brief conclusion.

2 Preliminaries and Related Research

2.1 Deep Spatial-Temporal Inference Network

Deep Spatial-Temporal Inference Network (DeSTIN) [14] originated as a mathematical model of the human visual and auditory cortex. The strengths of DeSTIN are its ability to learn from a small number of training examples and to handle temporal data.

The structure of DeSTIN is pyramidal-shaped. In this structure, as shown in Fig. 1, the two adjacent levels have the following relationship. At the $n-1$-th level, every $2 * 2$ nodes forms a group, and the outputs of the nodes are fed to a node on the n-th level. Similarly, on the n-th level, every $2 * 2$ nodes will form a group, and these nodes are connected to a node on the $n+1$-th level. Each level of DeSTIN contains a square grid of nodes and every node in the intermediate layers (hidden layers) contains a number of centroids[1]. In the lowest level of DeSTIN, raw data feeds into the corresponding nodes directly. We can consider that DeSTIN abstracts the raw input data to a different extent in the different levels.

The centroids associated with a node depict each possible "state" s for the node. At the beginning, the estimated centroids are initialized to random values. The original DeSTIN takes a well-known online algorithm, Winner Take All (WTA) approach, to cluster the input (called observation) at a given time, based on the centroids for that node at that time. This means that an observation o coming from the lower level is assigned to a single estimated centroid χ based on

[1] How many centroids in a node depends on a balance between resource limitation and representational capacity.

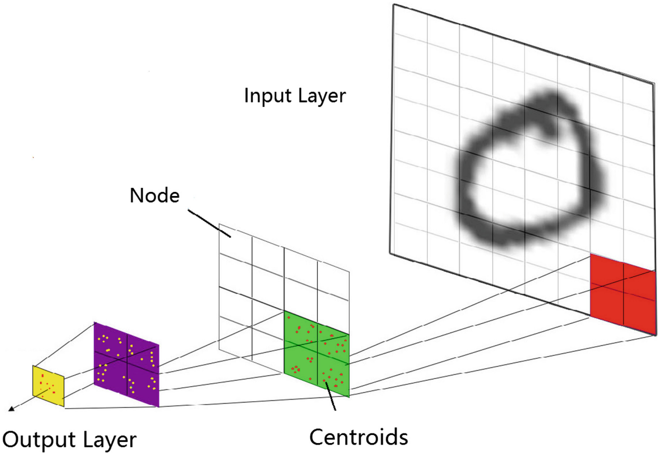


Fig. 1. The construction of DeSTIN

the minimum distance. Given observation o, a belief state s, and belief state of a higher level node a[2], DeSTIN uses the following formula to update the belief state or advice of a node from s to s':

$$b(s'|a) \propto P(o|s')\{\sum_{s\in S} P(s'|s,a)b(s)\} \tag{1}$$

Accurately speaking,

$$b'\left(s'|a\right) = \frac{\mathbf{Pr}\left(o|s'\right)\sum_{s'\in S}\mathbf{Pr}\left(s'|s,a\right)b\left(s\right)}{\sum_{s''\in S}\mathbf{Pr}\left(o|s''\right)\sum_{s'\in S}\mathbf{Pr}\left(s''|s,a\right)b\left(s\right)}, \tag{2}$$

where $P(o|s') = \frac{d_j^{-1}}{\sum_j d_j^{-1}}$ is static pattern similarity. d_j means the distance between the observation and the centroid j. The denominator can be considered as a normalization factor.

The thinking behind the formula is that: $\sum_{s'\in S}\mathbf{Pr}(s'|s,a)b(s)$ (For shorthand, it is denoted as PSSA.) is used to characterize the system dynamics and it modulates the static pattern similarity, so that the belief state inherently captures both spatial and temporal information regarding the raw input data.

The training process of a layer in DeSTIN can be understood roughly as follows: In the first step, every node obtains observations from the corresponding lower layer nodes, and DeSTIN calculates the "belief state" of the winning centroid based on the clustering results and computes the belief values of the nodes according to the above formula 1, after that DeSTIN will feed the belief values to the corresponding nodes in the higher level. Please recall that when computing the belief value of a node, DeSTIN will receive information, called "advice", from

[2] In DeSTIN, the belief state of a higher level node is called advice, which is the index of the winning centroid in the higher level node.

the higher level node. Repeating this from bottom to top, and DeSTIN outputs a "belief" value at the top level. Those belief values obtained from higher levels[3] can be regarded as a special kind of feature derived and abstracted from the raw data. The feature could have different uses, for example it can be fed into different classifiers, say KNN or SVM, to carry out pattern recognition. Or they can be fed into a more general cognitive system, to be correlated with other types of inputs such as auditory, verbal or sensory input, or with abstract knowledge obtained from language or structured knowledge bases.

3 EM-DeSTIN

The most important contribution of EM-DeSTIN algorithm results from replacing WTA with Online Expectation-Maximization (Online EM), an alternative unsupervised clustering algorithm, and the structure, the training and testing process of EM-DeSTIN are depicted in more detail as follows.

The overall structure of EM-DeSTIN is identical to that of DeSTIN (refer to Fig. 1). In the architecture of EM-DeSTIN, four nodes in a level are assembled into a single group, and the output of the group is associated with the input of a corresponding node in the upper level. There are several centroids in a node, and those centroids can be understood as "distribution of distributions" of the features contained in raw data in some sense. The primary difference between EM-DeSTIN and DeSTIN is the training process; Fig. 2 is a brief explanation of the training process of EM-DeSTIN.

The following steps detail the process of training one level in EM-DeSTIN.

1. Initialization actions need to be performed before learning begins, and it includes determining the number of levels according to input data, determining how many centroids will be included in a node, setting an initial value for each centroid, and initializing the PSSA table.
2. Each node receives a belief vector sent from the nodes in the lower level and advice sent from the parent node.
3. Every centroid in a node can be considered as a distribution of belief values about the contents of the spatial-temporal region referred to by the node. Concretely, the value of a centroid is a vector that has the same dimension as the belief vector a node has received. For all nodes in a level, EM-DeSTIN uses the step-wise online EM (sEM) algorithm to update those centroids. The sEM was proposed by Cappe and Moulines [15], which takes advantage of stochastic approximation theory.
4. EM-DeSTIN calculates the belief value for every centroid according to the updated results obtained from the above step, the updated PSSA table, and the "advice" sent from the parent node. Recall that the belief value of a node is a vector which includes all the values of the beliefs belonging only to itself; and that the advice passed from the parent node denotes the ordinal of a centroid in the upper level, which is closest to the input belief value.

[3] Depending on various applications, we can take the belief values that come from different numbers of levels as a "feature".

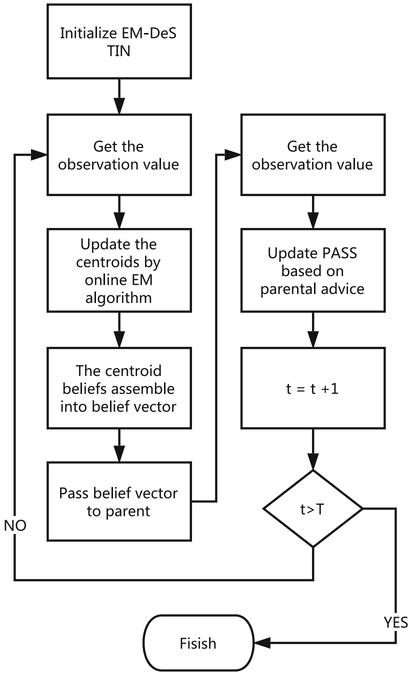

Initialize EM-DeS TIN
Get the observation value
Update the centroids by online EM algorithm
The centroid beliefs assemble into belief vector
Pass belief vector to parent
Get the observation value
Update PASS based on parental advice
t = t +1
t>T
NO
YES
Fisish

Fig. 2. The training process of EM-DeSTIN

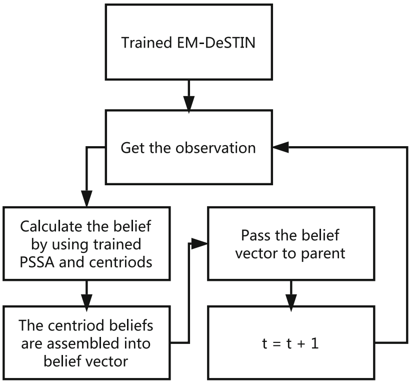

Trained EM-DeSTIN
Get the observation
Calculate the belief by using trained PSSA and centriods
The centriod beliefs are assembled into belief vector
Pass the belief vector to parent
t = t + 1

Fig. 3. The testing process of EM-DeSTIN

5. EM-DeSTIN will combine the belief values of the four nodes in a group into a vector, and output the vector to the corresponding node in the upper level.
6. The PSSA table will be updated according to the results obtained from Step 4 and the parent node's advice.

Figure 3 describes the testing process of EM-DeSTIN. It resembles the training process. The main difference is that the belief value centroids and PSSA table are not updated during the testing phase.

4 Experiments

We used a subset of the AR human face dataset[4] to carry out our experiment, and this dataset is widely used in the field of face recognition. The subset consists of 100 individuals and each person has 26 grayscale images, which are 165 * 120 pixels and 24 bits in depth. These images can be divided into two groups according to the dates of the shot. We take one group as training samples and the other group as the testing one. The samples of the dataset are displayed in Fig. 4.

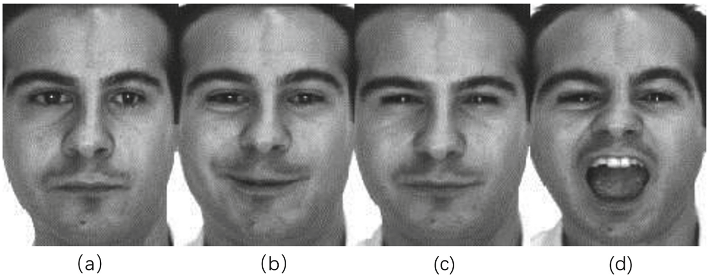


Fig. 4. Examples of the AR dataset

The meaning of Fig. 4(a) is Neutral expression; Fig. 4(b) is Smile; Fig. 4(c) is Anger and Fig. 4(d) is Scream. Gaussian noise and Salt-and-Pepper noise are added to the training samples. For the Gaussian noise, the mean was 0 and the variance was set to 0.01, 0.06 and 0.1 respectively. For the Salt-and-Pepper noise, the noise density was set to 0.01, 0.1 and 0.5 respectively.

In our experiments, we took the outputs of the top two layers of EM-DeSTIN and DeSTIN as input features to train two SVMs, and used the trained SVMs to classify the same testing samples separately. For our testing with other deep networks, we chose three different approaches, Deep Belief Nets (DBN) [16], Convolutional Neural Nets (CNN) [17] and Stacked Autoencoders (SAE) [18] to

Table 1. Emotion recognition of EM-DeSTIN, uniform DeSTIN, DBN, CNN and Autoencoder

	Gauss			Salt-pepper		
	0.01	0.06	0.1	0.01	0.1	0.5
DBN	0.35	0.33	0.30	0.41	0.38	0.32
CNN	0.43	0.41	**0.40**	0.45	0.42	**0.43**
SAE	0.41	0.40	0.38	0.41	0.39	0.728
DeSTIN	0.40	0.38	0.32	0.40	0.40	0.27
EM-DeSTIN	**0.4**	**0.43**	0.38	**0.49**	**0.45**	0.40

[4] http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html.

be equal. The source code of these three other approaches was obtained from DeepLearnToolbox[5].

Table 1 records the experimental results on different noise environment. The experiments show that the proposed algorithm is competitive at handling low to moderate level noises, and even in the case of loud-noise situation, EM-DeSTIN also has a good performance.

5 Conclusion

In this paper, we proposed an emotion recognition approach based on a deep learning architecture - Expectation-maximization Deep Spatial-Temporal Inference Network. The advantage of our method is that it has higher recognition rate than other deep learning methods when dealing with noisy facial expression pictures. It is the often case that the quality of images obtained by E-learning systems is not high. In future research, we will study how to use the transfer learning techniques [19] to improve performance of E-learning system or how to apply the deep learning method to solve the problem of Intelligent robots [20–22].

Acknowledgment. This work was supported by the National Natural Science Foundation of China (No. 61003014 and No. 61673328), the National Social Science Foundation (15BYY082) and the Natural Science Foundation of Fujian Province of China (No. 2017J01128).

References

1. Daradoumis, T., Bassi, R., Xhafa, F., Caballé, S.: A review on massive e-learning (MOOC) design, delivery and assessment. In: 2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), pp. 208–213. IEEE (2013)
2. Zaíane, O.R.: Building a recommender agent for e-learning systems. In: Proceedings of the International Conference on Computers in Education, pp. 55–59. IEEE (2002)
3. Binali, H.H., Wu, C., Potdar, V.: A new significant area: emotion detection in e-learning using opinion mining techniques. In: 3rd IEEE International Conference on Digital Ecosystems and Technologies, DEST 2009, pp. 259–264. IEEE (2009)
4. Sylwester, R.: How emotions affect learning. Educ. Leadersh. **52**(2), 60–65 (1994)
5. Oregan, K.: Emotion and e-learning. J. Asynchronous Learn. Netw. **7**(3), 78–92 (2003)
6. Russell, J.A.: Core affect and the psychological construction of emotion. Psychol. Rev. **110**(1), 145 (2003)
7. De Vicente, A., Pain, H.: Informing the detection of the students motivational state: an empirical study. In: International Conference on Intelligent Tutoring Systems, pp. 933–943. Springer (2002)
8. Torricelli, D., Goffredo, M., Conforto, S., Schmid, M.: An adaptive blink detector to initialize and update a view-basedremote eye gaze tracking system in a natural scenario. Pattern Recogn. Lett. **30**(12), 1144–1150 (2009)

[5] https://github.com/rasmusbergpalm/DeepLearnToolbox/.

9. Graesser, A.C., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R., Tutoring Research Group, et al.: Autotutor: a simulation of a human tutor. Cogn. Syst. Res. **1**(1), 35–51 (1999)
10. Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., Taylor, J.G.: Emotion recognition in human-computer interaction. IEEE Signal Process. Mag. **18**(1), 32–80 (2001)
11. Matsugu, M., Mori, K., Mitari, Y., Kaneda, Y.: Subject independent facial expression recognition with robust face detection using a convolutional neural network. Neural Netw. **16**(5), 555–559 (2003)
12. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
13. Jiang, M., Ding, Y., Goertzel, B., Huang, Z., Zhou, C., Chao, F.: Improving machine vision via incorporating expectation-maximization into deep spatio-temporal learning. In: 2014 International Joint Conference on Neural Networks (IJCNN), pp. 1804–1811. IEEE (2014)
14. Arel, I., Rose, D.C., Coop, R.: Destin: a scalable deep learning architecture with application to high-dimensional robust pattern recognition. In: AAAI Fall Symposium: Biologically Inspired Cognitive Architectures (2009)
15. Cappé, O., Moulines, E.: On-line expectation-maximization algorithm for latent data models. J. Roy. Stat. Soc.: Ser. B (Stat. Methodol.) **71**(3), 593–613 (2009)
16. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
17. LeCun, Y., et al.: Lenet-5, convolutional neural networks (2015). http://yann.lecun.com/exdb/lenet
18. Krizhevsky, A., Hinton, G.E.: Using very deep autoencoders for content-based image retrieval. In: ESANN (2011)
19. Jiang, M., Huang, W., Huang, Z., Yen, G.G.: Integration of global and local metrics for domain adaptation learning via dimensionality reduction. IEEE Trans. Cybern. **47**(1), 38–51 (2017)
20. Jiang, M., Zhou, C., Chen, S.: Embodied concept formation and reasoning via neural-symbolic integration. Neurocomputing **74**(1), 113–120 (2010)
21. Chao, F., Wang, Z., Shang, C., Meng, Q., Jiang, M., Zhou, C., Shen, Q.: A developmental approach to robotic pointing via human-robot interaction. Inf. Sci. **283**, 288–303 (2014)
22. Jiang, M., Yu, Y., Liu, X., Zhang, F., Hong, Q.: Fuzzy neural network based dynamic path planning. In: 2012 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 1, pp. 326–330. IEEE (2012)

Human Activities Transfer Learning for Assistive Robotics

David Ada Adama, Ahmad Lotfi(✉), Caroline Langensiepen, and Kevin Lee

School of Science and Technology, Nottingham Trent University,
Nottingham NG11 8NS, UK
ahmad.lotfi@ntu.ac.uk

Abstract. Assisted living homes aim to deploy tools to promote better living of elderly population. One of such tools is assistive robotics to perform tasks a human carer would normally be required to perform. For assistive robots to perform activities without explicit programming, a major requirement is learning and classifying activities while it observes a human carry out the activities. This work proposes a human activity learning and classification system from features obtained using 3D RGB-D data. Different classifiers are explored in this approach and the system is evaluated on a publicly available data set, showing promising results which is capable of improving assistive robots performance in living environments.

Keywords: Activity recognition · Activity classification · Assistive robotics

1 Introduction

Assistive robots deployed in living environments for applications such as elderly care should learn tasks by observing human carers performing routine duties. To achieve this goal, the assistive robots must be equipped with abilities to learn activities. This requires extracting descriptive information of the activities and classify them while they are performed by a human.

Learning human activities by an assistive robot can be classified under two methods [1]; *Independent Learning* which is concerned with learning an activity from scratch and learning by making use of transferred knowledge/information which is referred to as *Transfer Learning*. Independent learning is a method whereby an assistive robot learns to perform an activity independently without any prior knowledge of the activity. For example, an assistive robot learning an activity such as cooking (chopping vegetables) or opening a pill container without prior information of how a person would perform the activity. This requires more time in learning and more cost incurred which are limitations of the method. On the other hand, transfer learning methodology allows information acquired from prior experience to assist in learning an activity [3].

In the context of this paper, an assistive robot can learn to perform an activity from knowledge acquired as it observes a person perform similar activity. This

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_22

enables faster learning of activities and allows collaboration and adaptation of robots within living environments. Regardless of the method applied to learning an activity, the availability of descriptive information affects the understanding of an activity. Variations in information and understanding about an activity performed by a person and a robot performing similar activity can be defined as contained within a *knowledge gap* and transfer learning helps to bridge this gap.

Human activities are diverse in nature with imprecision, vagueness, ambiguity and uncertainty in information about the way activities are performed. Thus, variabilities are encountered when an assistive robot tries to learn activities. This affect correct classification of human activities which is relevant in improving the amount of knowledge that can be used by a robot in learning. To capture imprecisions and uncertainties, fuzzy logic has proven to be a suitable method which allows incorporation of imprecisions and uncertainty expressiveness within information [3,4] and thus can be applied to classify human activities. Combining this method with transfer learning would improve assistive robots learning human activities by observing while activities are performed. Other learning techniques applied to learning/classifying human activities are limited in their ability to handle vagueness, imprecision and uncertainties in activities when considering acquiring knowledge that can be transferred across different learners.

In this paper, a method for learning and classifying activities carried out by humans in the context of assistive robotics are presented. Set of features representing daily activities are extracted from human activities and these features are used as input to a classifier to find relevant structures within the features. Classification of activities is done by exploiting different classification techniques; a multiclass Support Vector Machine (SVM), K-Nearest Neighbour (K-NN) and also, Fuzzy C-means (FCM) clustering technique. A cross-validation test is performed on the trained classifiers to measure their performance in predicting activities. The aim of the proposed work presented in this paper is to build a human activity learning and classification system that can be incorporated in an assistive robot to improve human-robot interaction in living environments.

The structure of this paper is as follows: In Sect. 2, a review of related works in this area is presented. Section 3 gives details of the method applied to our approach for feature extraction and classification of activities. Initial results are presented in Sect. 5. Section 6 presents conclusions and future work to be undertaken.

2 Related Work

Learning and classification of human activities is often referred to as Human Activity Recognition (HAR) [5,6]. One of the main objectives is to extract descriptive information (i.e. features) from human activities to be able to distinctly characterize and classify one activity from another. An integral component of learning an activity is how information of the activity is obtained (i.e. observation). For human activities, information obtained using visual and non-visual sensors makes it a lot easier to understand and learn activities as they are performed. Visual sensors such as RGB cameras can be used to obtain descriptive information of an activity in $2D$. However, this information is limited in effectively characterizing an activity [7]. Additional depth information using RGB-D sensors provide several advantages as they are better suited for observing human activities to detect human pose used to build activity recognition systems.

To effectively characterize activities from information obtained using RGB-D sensors, machine learning and reasoning methods have been applied by many researchers [8–10]. These methods provide an understanding of how activities are learned and relationships between activities. However, there is some uncertainty that exist in how one actor performing an activity would differ from another actor performing similar activity. This hinders HAR systems from going mainstream.

Information obtained from RGB-D sensors gives very important information relevant for a robot to understand an activity. By exploring human pose detection using RGB-D sensors, activity recognition has seen more advancement in recent times [11,12]. Using RGB-D sensors extracts $3D$ skeleton data from depth images and body silhouette for feature generation. In [11], the RGB-D sensor is used to generate human $3D$ skeleton model with matching of body parts linked by its joints. They extract positions of individual joints from the skeleton in a $3D$ form *x, y, z.* Authors in [13] use similar RGB-D sensor to obtain depth silhouette of human activities from which body points information are extracted for the activity recognition system. Another approach is shown in the work in [14] where the RGB-D sensor is used to obtain orientation-based human representation of each joint to the human centroid in 3D space. Raw data obtained from these sensors have to be preprocessed. This process is carried out to reduce redundancy in data for better representation of features of an activity.

Classification of human activities is carried out by extracting relevant features from data obtained using RGB-D sensors. In our previous work a method for activity recognition using RGB-D data is proposed [15]. The $3D$ joints positions information extracted from the sensor are transformed into feature vectors by applying K-means clustering to group key postures of an activity. The posture features are used as input to a neural network for classification of human activities. Authors in [11] proposed a combination of multiple classifiers to form a Dynamic Bayesian Mixture Model (DBMM) to characterize activities using features obtained from distances between different parts of the body. Also, [16] applied statistical covariance of 3D joints (Cov3DJ) as features to encode the skeleton data of joint positions. Another approach seen in [17] used a sequence of joint trajectories and applied wavelets to encode each temporal sequence of joints into features.

3 Activity Features

In the proposed system, the process starts by obtaining RGB-D sensor information from the performed activities. The architecture of the proposed system is

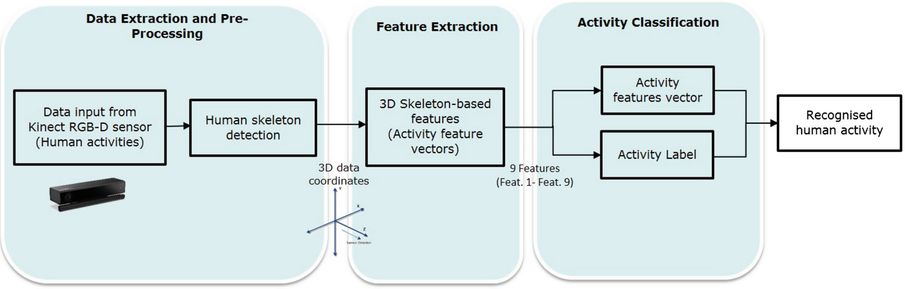

Data Extraction and Pre-Processing
Data input from Kinect RGB-D sensor (Human activities)
Human skeleton detection
3D data coordinates
Y
X
Z
Sensor Direction
Feature Extraction
3D Skeleton-based features (Activity feature vectors)
9 Features (Feat. 1- Feat. 9)
Activity Classification
Activity features vector
Activity Label
Recognised human activity

Fig. 1. Architecture of proposed system.

shown in Fig. 1. Incoming data is obtained using a Kinect RGB-D sensor [18] which tracks human joint movements and their transitions over time. Data preprocessing and 3D skeleton-based feature selection are performed before they are applied to the classifier. More details are provided below.

3.1 Data Pre-processing

Data is obtained from 3D $\{x, y, z\}$ skeleton detection of an actor performing an activity. The skeleton of the actor is tracked using an RGB-D sensor for obtaining positions of joints of the human body. The data representing an activity consist of N number of frames (observations). An example of frames of human activities obtained using the RGB-D sensor which shows the tracked skeleton of human joints is shown in Fig. 2 [19]. The Kinect RGB-D sensor considers the skeleton frame of reference from the sensor. However, for better representation of features of an activity, we consider the frame of reference for all joints relative to the torso centroid coordinates.

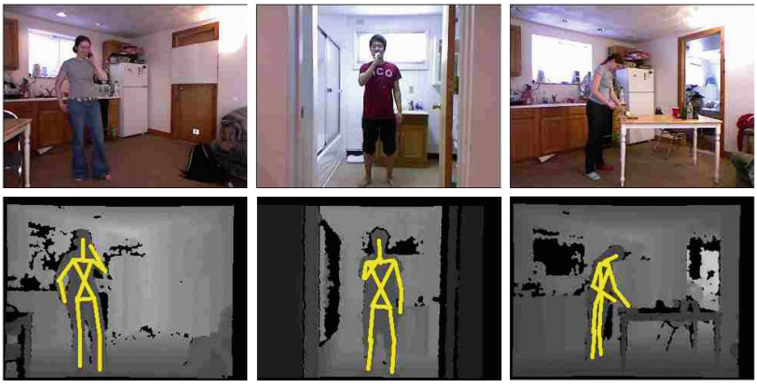


Fig. 2. Frames of human activities performed in a living environment extracted using an RGB-D sensor [19].

For a skeleton frame consisting of joints j, the torso centroid coordinate is represented as j_t. The distance between the i^{th} joint j_i and j_t is given as $d_i = j_i - j_t$. This distance is computed for all joints in each frame of an activity. After computing distances, each frame n is represented by a vector containing joints distance relative to the torso $V_n = \{d_1, d_2, d_3, \ldots, d_i\}$.

3.2 3D Skeleton-Based Features

Feature extraction is an important aspect of any activity recognition system as raw data obtained from activities do not provide enough information to allow implementing an activity recognition system. The joints distance vectors obtained from the pre-processing stage is converted into a set of useful features that model human activities.

Features obtained in human activity recognition systems can be computed using human skeleton joints coordinates obtained from an RGB-D sensor. The features are often based on raw joints positions and displacement-based representations when considering temporal and spatial data. In this work, displacement features from skeleton joint coordinates are used. We exclude temporal information to make the system independent of speed of joint movements.

The features used in this work are similar to the ones proposed by [11]. These features are obtained from joint displacement positions of a person performing an activity.

The features are based on distance between both left and right hands, as a lot of attention is drawn towards the pose of the hands when performing an activity. Distance between hands and head, between hip and feet, shoulder and feet, between the initial hand (for both hands and elbows) position of the first frame and the next frames. These are computed using the Euclidean distance equation given as $\delta_{(j_{b1}, j_{b2})}$.

$$\delta_{(j_{b1}, j_{b2})} = \sqrt{(j_{b1}^{x} - j_{b2}^{x})^2 + (j_{b1}^{y} - j_{b2}^{y})^2 + (j_{b1}^{z} - j_{b2}^{z})^2} \tag{1}$$

where the joints of a human skeleton are represented by j_b for b= {*face, hand, shoulder, hip, feet and torso*}. Each joint coordinate is represented in 3D {*x, y, z*}. The Euclidean distance computed represent features f of an activity.

To classify different activities, each activity is represented by a set of feature vectors which characterize the activity as explained above and classification is done on this feature vector. Therefore, an activity A is characterized by features $A = \{f_1, f_2, f_3, \ldots, f_m\}$, where f_m is the m^{th} feature vector for the activity.

3.3 Features Normalization

Features extracted from an activity can be heterogeneous and this could introduce problems during classification if one of the selected features varied more than another. To avoid this problem, data normalization is performed on the selected activity features and the normalized features are used as input to train and validate the classifiers. In order to normalize our features, the mean and standard deviation of each feature vector is determined and we create new feature set that has zero-mean and a unit standard deviation using Eq. 2. This is done to remove distortion due to data heterogeneity before classification is done with the normalized features.

$$\text{Normalized feature} = \frac{f_m - \mu_m}{s_m}, \tag{2}$$

where, s_m is the standard deviation and μ_m is the mean of an activity feature f_m.

4 Activity Classification

The final stage in learning human activities is classification of activities using the extracted feature vectors. This step aims to associate feature vectors to the correct activity. As stated in Sect. 1 different classification techniques are used in order to classify activities. Support Vector Machine (SVM), K-Nearest Neighbour (K-NN) and also, Fuzzy C-means (FCM) are frequently used in many classification problems and they are exploited here. However, the FCM algorithm is not commonly used but poses to be a good method for classifying activities. In the FCM algorithm, several features which characterize an object are assigned to different classes with different membership grades. A benefit of using this method for classification is that an initial knowledge of the feature vectors is not required as membership functions are formed automatically by the method.

4.1 Support Vector Machine (SVM)

Considering the application of SVM in classifying activities we apply a method used in [20] where a multi-class SVM is applied to activity recognition. The multi-class SVM is an extension of the SVM from binary classifier. A *"one against-one"* approach which is based on the construction of several binary SVM classifiers is stated to be the most suitable for practical use. This method is necessary for M classes dataset, where $M > 2$. A training phase is carried out during which the activity features are given as input to the multi-class SVM together with activity labels. In the test phase, activity labels are obtained from the classifier.

4.2 K-Nearest Neighbour (K-NN)

The K-NN is among one of the simplest machine learning algorithms and is a method of classifying objects based on closest training points in the feature space. An object is assigned to a class most common among its k nearest neighbours (where k is a positive integer) by a majority of votes of its neighbours. In most cases, Euclidean distance is used as the metrics in finding the nearest neighbours to an object. Applying this method in the proposed approach, in the training phase, the activity feature vectors and activity labels of the training set are stored. During the classification phase, the user defined constant k and unlabelled activity feature vectors are classified by assigning a label most frequent among the k training samples.

4.3 Fuzzy C-Means Algorithm

Fuzzy c-means (FCM) algorithm is a method of clustering which allows one piece of data to belong to two or more clusters. It is frequently used in pattern recognition. Although, FCM is primarily used to cluster data, it could also be employed as a classifier to provide a measure of belonging to each cluster. This is an interesting approach for activity recognition as it will provide a measure of membership to each of the identified classes. Readers are referred to [2] for more details about FCM.

5 Experimental Results

The proposed approach described in this paper is evaluated using publicly available human activity dataset, CAD-60 data set [12]. This data comprises RGB-D sequence of human activities acquired using an RGB-D sensor. 12 activities and an addition of a random + still activity performed by four different participants in five different locations namely; bathroom, bedroom, kitchen, living room and office environments. The activities are listed as follows, with the labels corresponding to the labels shown in the results figures in Figs. 3, 4 and 5.

A1 Rinsing mouth,
A2 Brushing teeth,
A3 Wearing Lens,
A4 Talking on the Phone,
A5 Drinking water,
A6 Opening pill container,
A7 Cooking (chopping),
A8 Cooking (stirring),
A9 Talking on the couch,

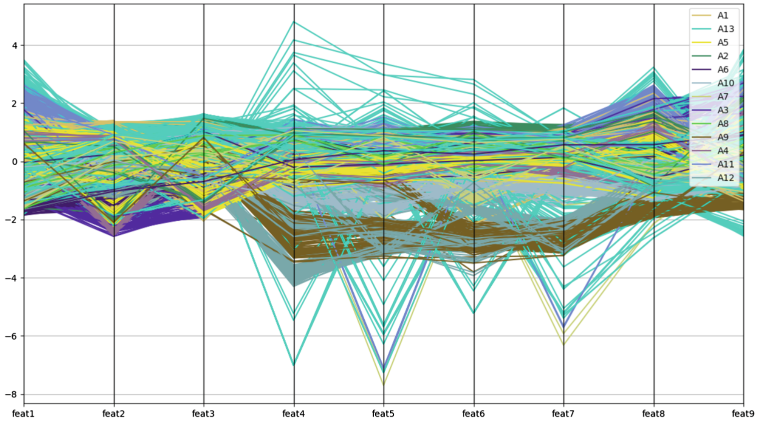
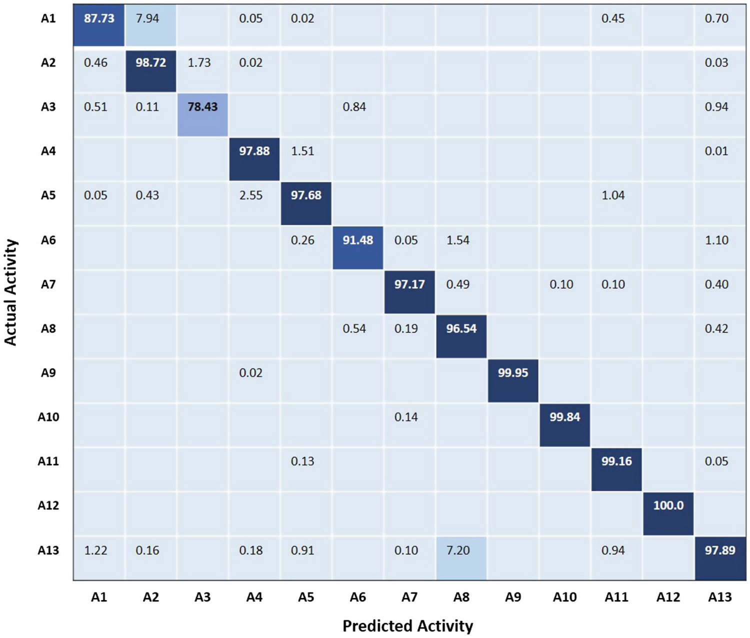
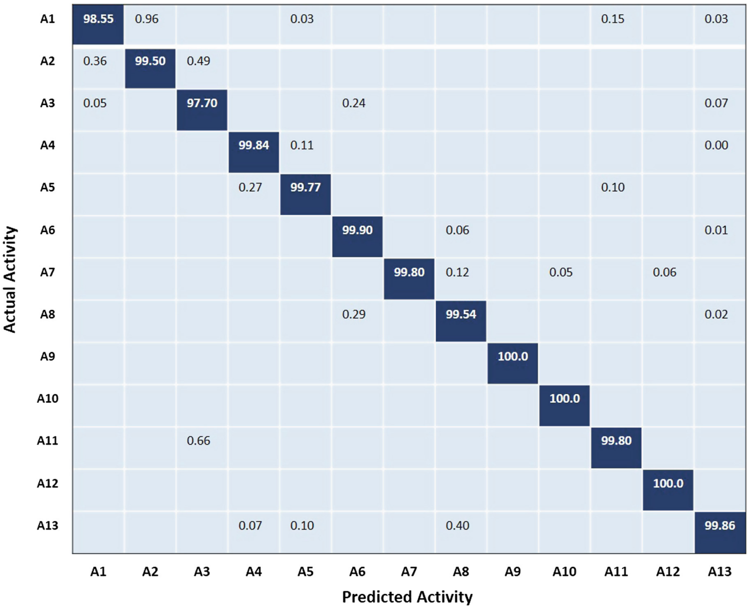


Fig. 3. Parallel coordinate plot showing selected 9 features ($feat1$–$feat9$) of 13 activities ($A1$–$A13$) obtained from the CAD-60 human activity dataset.



	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13
A1	87.73	7.94		0.05	0.02						0.45		0.70
A2	0.46	98.72	1.73	0.02									0.03
A3	0.51	0.11	78.43			0.84							0.94
A4				97.88	1.51								0.01
A5	0.05	0.43		2.55	97.68						1.04		
A6					0.26	91.48	0.05	1.54					1.10
A7							97.17	0.49		0.10	0.10		0.40
A8						0.54	0.19	96.54					0.42
A9				0.02					99.95				
A10							0.14			99.84			
A11					0.13						99.16		0.05
A12												100.0	
A13	1.22	0.16		0.18	0.91		0.10	7.20			0.94		97.89

Fig. 4. Confusion matrix plot showing the performance of SVM classifier for classification of 13 activities ($A1$–$A13$) obtained from the CAD-60 human activity dataset.



	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13
A1	98.55	0.96			0.03						0.15		0.03
A2	0.36	99.50	0.49										
A3	0.05		97.70			0.24							0.07
A4				99.84	0.11								0.00
A5				0.27	99.77						0.10		
A6						99.90		0.06					0.01
A7							99.80	0.12		0.05		0.06	
A8						0.29		99.54					0.02
A9									100.0				
A10										100.0			
A11			0.66								99.80		
A12												100.0	
A13				0.07	0.10			0.40					99.86

Fig. 5. Confusion matrix plot showing the performance of K-NN classifier for classification of 13 activities ($A1$–$A13$) obtained from the CAD-60 human activity dataset.

A10 Relaxing on couch,
A11 Writing on board,
A12 Working on computer and
A13 Random + still activity.

The first step is data pre-processing which is performed on the data set to obtain each joint coordinate relative to the torso coordinate. Features are then calculated from the pre-processed data using the method described in Sect. 3.2 and 9 features are obtained for each activity. These features are used as input to the classifiers. In Fig. 3, a parallel coordinate plot showing the 9 features selected across a sample of observations of the different activities is shown. This shows how features corresponding to different activities are appear to be similar, thus making the process of classification complicated. However, we observe some more disparity in features 4–7 which are expected to be easily separable when compared with the other features.

We present the classification results for SVM and K-NN classifiers in terms of *Precision* and *Recall* shown in Table 1 and confusion matrices presented in Figs. 4 and 5 for the overall classification of the activities. For testing of the trained classifier, we use a method of *leave-one-out* cross-validation strategy in which 70% of the data set is used in training the classifier and the rest 30% is used for testing and validation of the classifier.

Table 1. Result of SVM and K-NN classifier used in overall classification of 13 human activities obtained from the CAD-60 data set. The table presents precision and recall scores for both classifiers when 9 features are used for classification on 'have seen' person.

Activity	SVM classifier		K-NN classifier	
	Prec	Rec	Prec	Rec
Rinsing mouth	97.03	88.39	99.58	98.55
Brushing teeth	92.96	98.69	99.04	99.49
Wearing lens	98.61	76.69	98.83	97.70
Talking on the phone	97.29	97.78	99.66	99.84
Drinking water	97.05	97.86	99.75	99.77
Opening pill container	97.60	92.47	99.46	99.90
Cooking (chopping)	99.75	96.40	100.0	99.80
Cooking (stirring)	91.03	96.97	99.42	99.53
Talking on the couch	100.0	99.95	100.0	100.0
Relaxing on couch	100.0	99.89	99.94	100.0
Writing on board	97.77	98.95	99.75	99.80
Working on computer	100.0	100.0	99.94	100.0
Random + still activity	96.50	97.89	99.86	99.86
Average	97.35	97.02	99.63	99.73

It can be observed from the results presented in Table 1 that Using the SVM classifier, we obtain classification accuracy of 97.02% on the test activities data. In Fig. 4, a confusion matrix for SVM classification results is shown, where the last column of the matrix (i.e. column 13) has the random activity which is a neutral activity performed by the participants (activities that were not performed with high confidence). This is included to show the confidence of our approach. In Fig. 5, similar result is also shown when we use a K-NN classifier. However, with the K-NN classifier we attain an accuracy of 99.73% on the test activities data. Note that the results presented are for classification on *'have seen'* test activities data after the classifiers are trained.

For the classification using the FCM algorithm, 13 clusters are selected which represent the number of activities in the data set to be classified. The metrics usually applied to clustering results analysis are; *Purity*- which is an external evaluation criterion for cluster quality, *Normalized Mutual Information* (NMI)- and *Rand Index* (RI). The best result for FCM classification is obtained when we apply a fuzziness coefficient $\phi = 1.4$. Higher values of ϕ result in more overlap between clusters and lower values result in less overlap between clusters which could result in hard clustering. After clustering we obtain the results shown in Table 2.

Table 2. Fuzzy C-means classification result of 13 human activities obtained from the CAD-60 data set. The table shows the metrics used in evaluating the results when 9 features are used for classification

Evaluation metric	Score
Purity	0.55
Normalized Mutual Information (NMI)	0.52
Rand Index (RI)	0.30

6 Conclusions

In this work, classification methods for human activities using 9 features extracted from human activities data collected using an RGB-D sensor is presented. This is part of an on-going research for transfer learning of human activities using assistive robots. It can be observed from Fig. 3, the complexity of human activities using the selected features. This requires proper selection of informative features which provide relevant information that could be used in distinctly characterizing activities. Thus, future work will focus on feature extraction methods for human activities.

The purpose of classifying human activities is to be able to build a system to distinctly characterize activities as they are performed in living environments in order to have assistive robots learn to perform the activities.

References

1. Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. J. Big Data **3**, 9 (2016)
2. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
3. Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., Zhang, G.: Transfer learning using computational intelligence: a survey. Knowl.-Based Syst. **80**, 14–23 (2015)
4. Shell, J., Coupland, S.: Fuzzy transfer learning: methodology and application. Inf. Sci. **293**, 59–79 (2015)
5. Iglesias, J.A., Angelov, P., Ledezma, A., Sanchis, A.: Human activity recognition based on evolving fuzzy systems. Int. J. Neural Syst. **20**(05), 355–364 (2010)
6. Zhang, H., Yoshie, O.: Improving human activity recognition using subspace clustering. In: IEEE Machine Learning and Cybernetics (ICMLC), vol. 3, pp. 1058–1063 (2012)
7. Han, F., Reily, B., Hoff, W., Zhang, H.: Space-time representation of people based on 3D skeletal data: a review. Comput. Vis. Image Underst. **158**, 85–105 (2017)
8. Koppula, H.S., Gupta, R., Saxena, A.: Learning human activities and object affordances from RGB-D videos. Int. J. Robot. Res. **32**, 951–970 (2013)
9. Li, S.-Z., Yu, B., Wu, W., Su, S.-Z., Ji, R.-R.: Feature learning based on SAEPCA network for human gesture recognition in RGBD images. Neurocomputing **151**, 565–573 (2015)
10. Kviatkovsky, I., Rivlin, E., Shimshoni, I.: Online action recognition using covariance of shape and motion. In: IEEE Conference on Computer Vision and Pattern Recognition (2014)
11. Faria, D.R., Premebida, C., Nunes, U.: A probabilistic approach for human everyday activities recognition using body motion from RGB-D images. In: 23rd IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN, pp. 732–737. IEEE (2014)
12. Sung, J., Ponce, C., Selman, B., Saxena, A.: Human activity detection from RGBD images. In: Proceedings of the 16th AAAI Conference on Plan, Activity, and Intent Recognition (AAAIWS11-16), pp. 47–55. AAAI Press (2011)
13. Jalal, A., Kamal, S.: Real-time life logging via a depth silhouette based human activity recognition system for smart home services. In: 11th IEEE International Conference on Advanced Video and Signal-Based Surveillance, AVSS, pp. 74–80 (2014)
14. Gu, Y., Do, H., Ou, Y., Sheng, W.: Human gesture recognition through a kinect sensor. In: IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1379–1384. IEEE (2012)
15. Adama, D.A., Lotfi, A., Langensiepen, C., Lee, K., Trindade, P.: Learning human activities for assisted living robotics. In: Proceedings of 10th Conference on Pervasive Technology Related to Assistive Environments (PETRA 2017), Island of Rhodes, Greece, 21–23 June 2017
16. Hussein, M.E., Torki, M., Gowayyed, M.A., El-Saban, M.: Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, pp. 2466–2472, Beijing, China. AAAI Press (2013)
17. Wei, P., Zheng, N., Zhao, Y., Zhu, S.-C.: Concurrent action detection with structural prediction. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3136–3143. IEEE (2013)

18. Microsoft, Developing with Kinect for Windows. https://developer.microsoft.com/en-us/windows/kinect/develop
19. Cornell activity datasets CAD-60. http://pr.cs.cornell.edu/humanactivities/data.php
20. Cippitelli, E., Gasparrini, S., Gambi, E., Spinsante, S.: A human activity recognition system using skeleton data from RGBD sensors. Comput. Intell. Neurosci. **2016**, (2016). Article ID 4351435. 14 Pages

3D Simulation of Navigation Problem of People with Cerebral Visual Impairment

Yahya Qasim I. Al-Fadhili[1(✉)], Paul W.H. Chung[2], Baihua Li[2], and Richard Bowman[3]

[1] Computer Science Department, College of Education, Mosul University, Mosul, Iraq
y.q.i.al-fadhili@lboro.ac.uk

[2] Computer Department, School of Science, Loughborough University, Loughborough, UK
{P.W.H.Chung, B.Li}@lboro.ac.uk

[3] Great Ormond Street Hospital, London, UK
richard.bowman@gosh.nhs.uk

Abstract. Cerebral Visual Impairment (CVI) is a medical area that concerns the study of the effect of brain damages on the visual field (VF). People with CVI have difficulties in their mobility and they have behaviours that others find hard to understand due to their visual impairment. A branch of Artificial Intelligence (AI) is the simulation of behaviour by building computational models that help to explain how people solve problems or why they behave in a certain way. This paper describes a novel computational system that simulates the navigation problem that is faced by people with CVI. This will help relatives, friends, and ophthalmologists of CVI patients understand more about their difficulties in navigating their everyday environment.

The navigation simulation system is implemented using the Unity3D game engine. Virtual scenes of different living environment are also created using the Unity modelling software. The vision of the avatar in the virtual environment is implemented using a camera provided by the 3D game engine. Filters that mimic visual defects are created automatically and placed in front of the visual field of the avatar. The filters are based on the visual field charts of individual patients. Algorithms for navigation based on the limited vision have also been developed to demonstrate navigation problems because of the visual defects. The results showed different actions for the navigation behaviours according to the patients' vision, and the navigations differ from patient to another according to their different defects.

Keywords: Vision impairment simulation · Cerebral Visual Impairment · Imitation · AI based modeling and navigation

1 Introduction

Cerebral Visual Impairment (CVI) is medically defined as a neurological disorder caused by damage to the occipital lobes and/or to the visual pathways and it is associated with disturbed visual sense because of the brain deterioration rather than eye damage [1–3].

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_23

Visual perception is a function of the eyes and brain together, thus, it's the way that human knows and understands the world around them through what they see using both eyes and brain [4]. It is the process or ability by which sensory information from our sense eyes is transformed to produce the recognition of shape, size, and brightness of objects and the distance of how close or how far away an object is [5].

CVI affects a person's ability to see and recognize obstacles in their surrounding area, thus affect their ability to move around in their everyday environment. Therefore they often appear to be clumsy but it is due to their inability to see. There is the need to demonstrate the difficulties that CVI patients encountered so that their relatives and friends have a better understanding of the problems that they face in their everyday living. If the CVI patient is a child then the better understanding will also help teachers in schools to accommodate for their needs due to their vision deficiency.

This paper describes the design and implementation of a novel 3D virtual reality system that simulates the navigation problems faced by people with CVI. In this system an avatar will act as an individual moving around in a 3D environment and will avoid or bump into obstacles mimicking the behavior of the particular individual. The system is able to simulate the visual impairment of individual patients based on their visual field vision chart results. This serves to highlight the specific problems that individuals faced and how the environment may be adapted to suit their needs.

2 Related Work

Several studies have provided evidence that CVI is possibly a contributor of cognitive and intellectual dysfunction. Therefore, CVI is considered as a disease that requires educational and training intervention. In 2008, Dutton [6] developed a set of strategies to help parents and teachers to support children with CVI in different daily situations.

Instead of coping strategies, there are other projects that develop devices that would help the blind or partially blind. For example, the Google Glass developed through the Open Glass Project [7] can be considered as a hands-free smartphone. It responds to voice commands by taking a picture in front then processes the image, searches for a specified object of interest and indicates whether it exists within view. This makes it easier for CVI patients to identify and locate objects in their living environment [8].

Another device called Badge3D is a tag detection system. Barcodes (tags) are attached to selected objects in the living environment with the tags clearly visible for the recognition system to work. The system was designed specifically for indoor use to help visually impaired patients navigate their living environment independently. Like Google Glass, the system is voice activated [9].

Another project, called Hazard Perception Test (HPT), focused on detecting what CVI patients can see with their visual defects. Films of driving scenarios are used to measure the rate of responses of new drivers for detecting hazards using a computer-based test. The test focused on superior (upper) and inferior (lower) visual field deterioration. The study concluded that hazard detection is more affected by superior than inferior defects [10].

Research has also attempted to simulate and measure the reading performance of glaucoma patients. Patients tend to use their central visual field to gaze at the line and the word being read. An eye tracker was adjusted or distorted to match the visual field of the reader, as the centre of the reader's eyes is not the centre of the visual field due to their visual impairment. Eye movements were video recorded. The gazing position was shown by a red point over each word being read. The blur of the peripheral areas in the scene indicates the defects corresponding to visual field areas [11].

3 Aim and Overall Architecture

The aim of this project is to develop algorithms that simulate and illustrate what the CVI patients can see and how they would navigate with their vision impairment using a 3D modelling environment. Figure 1 shows the main steps of how the system is accomplished and the input and output for each sub-process.

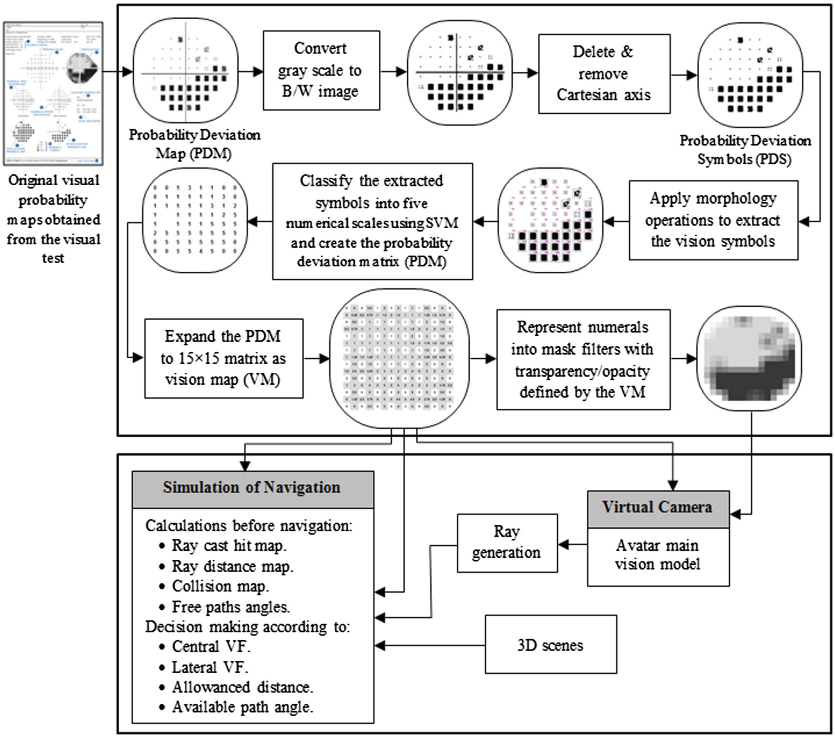

Original visual probability maps obtained from the visual test
Probability Deviation Map (PDM)
Convert gray scale to B/W image
Delete & remove Cartesian axis
Probability Deviation Symbols (PDS)
Apply morphology operations to extract the vision symbols
Classify the extracted symbols into five numerical scales using SVM and create the probability deviation matrix (PDM)
Expand the PDM to 15×15 matrix as vision map (VM)
Represent numerals into mask filters with transparency/opacity defined by the VM
Simulation of Navigation
Calculations before navigation:
• Ray cast hit map.
• Ray distance map.
• Collision map.
• Free paths angles.
Decision making according to:
• Central VF.
• Lateral VF.
• Allowanced distance.
• Available path angle.
Ray generation
Virtual Camera
Avatar main vision model
3D scenes

Fig. 1. System architecture – Cerebral Visual Impairment simulation of navigation behaviour

4 Methodology

4.1 Image Pre-processing

Visual field is tested using special device called perimetry. It produces a chart with many numeral, deviation and symbolical patterns that describe a person's vision. The last symbolic pattern in the chart which is the deviation probability map (DPM) represents the final calculations of the testing areas in a symbolic form. It consists of five scaled symbols and each represents the degree of the deficiency in specific visual spot.

Digitizing the image symbols of vision deficiency needs to be done first. This can be achieved by detecting the symbol region from the VF chart, and converting them into black & white symbol block image (B/W). The existed Cartesian coordinates were located and deleted using the property of the largest connected component/biggest object; see Fig. 2(a). A sequence of mathematical morphology operations were used to specify shapes in this input image. The values represented by a detected vision symbol block were derived from the number of pixels, the centroid of the vision symbol and symbol area boundaries, see Fig. 2(b).

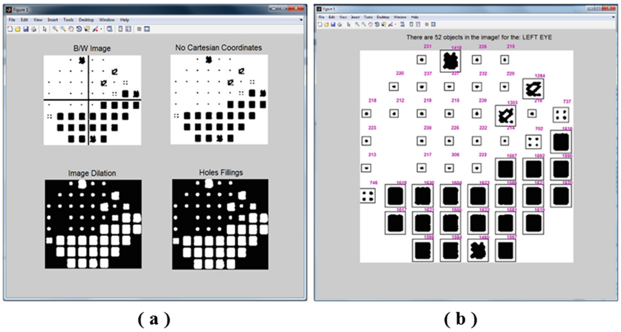


Fig. 2. (a) Image processing for features extraction and (b) object identification

The extracted symbols were saved as image patches individually. They were classified into five classes representing visibility levels using a supervised machine learning classifier support vector machine (SVM) [12, 13]. Training samples of visibility symbols were saved in five separated folders (representing the five classes) for learning and matching process as feature space. Then the extracted input symbols are classified and mapped to these output features. The result of classification is an integer number [1–5] refer to the degree of visual deterioration. These data are saved into a 2-D array of size 8 × 8. As shown in Fig. 3(b), the spots are digitized presenting significant visual differences via levels of transparency. The 8 × 8 array was expanded to 15 × 15; this is achieved by inserting other elements between two successive elements to smooth the transitions. Thus, the six gray scales [0, 1, …, 5] are extended to 21 transparency levels [0, 0.25, 0.5, 0.75, 1, 1.25, …, 4.25, 4.5, 4.75, 5] as shown in Fig. 3(c). The final numeral array of the clear and defect values represent the Vision Map which is considered as the main map that the proposed computational model depends on.

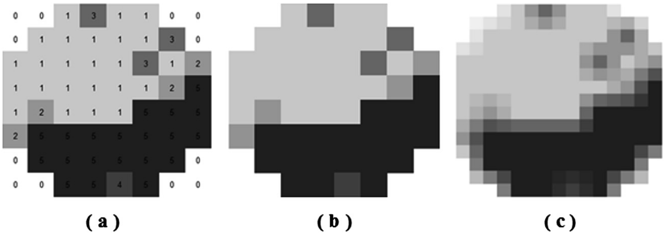


Fig. 3. Vision map: (a) classified six transparency levels in 8 × 8 vision map, (b) the discrete transition between transparency levels and (c) the expanded smoothed 15 × 15 vision map

Each visual level in the vision map was converted into a gray image with specified degree of transparency as a mask filter using Photoshop CS6. This image is in three channels with equal colour values; a fourth (alpha) channel was added for transparency using texture's advanced properties in Unity3D.

4.2 The Projection of the Vision Map

As the brain damages affect patients' eyes, parts of the vision area are confused and the clarity/blurring of the vision area is determined by the deteriorating degree, and thus they determine the grade of transparency or blur that used. Unity3D is a 3D game engine which is suitable for projecting the CVI patients' vision into 3D. An avatar can be used to represent the character of a patient in First Person Shooter (FPS) mode. The attached camera to the top of the avatar can be used to simulate the patient's eyes. Therefore it was used to project the mask filters onto it to make the game view window in Unity reflect the VF of the patient to clarify what this patient is really see.

A 15 × 15 array of plane objects was designed and centralized in front of the main camera to cover the whole vision view. Each one has a numeric value taken from the vision map. A specific transparent texture was attached to every plane programmatically and automatically. The filters mask moves within the avatar camera and project the blurred spots in the person's eyes during its navigation to emulate defect vision. Unity has basic functions and features that provide 3D physics to the avatar and other assets in the scene. Such physics include colliders and ray-casting. The project considered the vision rays as beams that start from the eye and reach any point in the 3D space. Therefore, vision rays were generated using the ray-casting physics feature to simulate vision beams. A second array (15 × 15) of planes also was designed and placed in front of the mask filter. These planes were gathered into one object to represents the rays object. The lengths of each ray depend on the numerical value taken from the vision map and produce the Ray Map. The bigger values (means high percentage of opacity) are represented by short beam in length. While small values (means high percentage of transparency) are represented by long beam. The avatar moves ahead and turns away depending on the movement and turning of the attached rays object according to a main algorithm script. Figure 4 shows the normal vision rays instantiation and representation, while Fig. 5 clarifies the deteriorated vision.

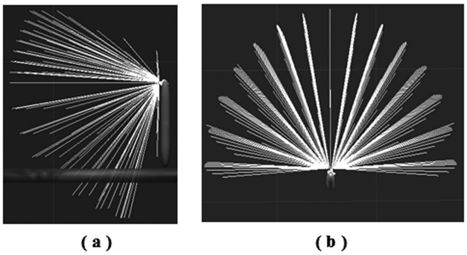


Fig. 4. Vision rays of normal vision, (a) side view and (b) upper view

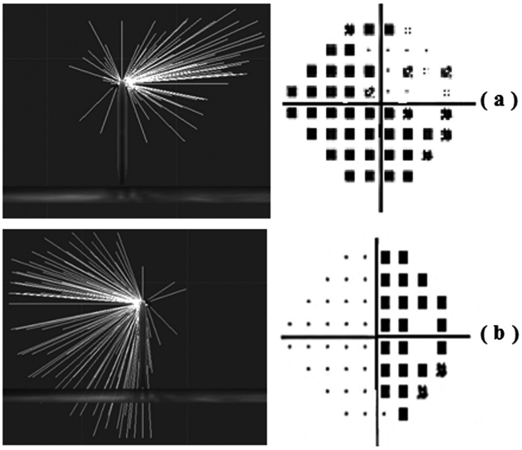


Fig. 5. Vision rays for deteriorated visibility (back views), (a) left superior and most of the interior VF defects and (b) right side VF defect

By using this, it facilitates the control of the avatar navigation according to whether the rays hit (colliding) or not. The length of each ray is calculated as follows:

$$Ratio = View_{depth}/5 \tag{1}$$

$$Ray_{length} = View_{depth} - Defect_{level} * Ratio \tag{2}$$

Where *ratio* is the percentage of the view depth to the largest scale in the pattern, $view_{depth}$ is the depth of the person vision, and $defect_{value}$ is the scale value extracted and calculated from the pattern.

According to the VF measurements for central, nasal and lateral fields [14], each ray was instantiated according to the attached C# script to each plane and rotated about the x and y axes according to the i^{th} and j^{th} of their positions in the vision map. The rotation was calculated as follows:

$$y_{rotation} = (j \times 12.857) - 90 \tag{3}$$

$$x_{rotation} = (i \times 4.286) - 30 \quad \text{for superior (upper) field} \tag{4}$$

$$x_{rotation} = (i \times 10.714) - 75 \quad \text{for inferior (lower) field} \tag{5}$$

where

$y_{rotation}$: the rotation about y-axis.
$x_{rotation}$: the rotation about x-axis.

4.3 Simulation of Navigation

The ray map shows the areas and the surrounding objects that can be seen or not. The hit-rays mean objects are detected; however, the non-hit rays mean objects are not detected. This principle facilitates the navigation through a flat platform like rooms, going upstairs or downstairs, and recognizing moving objects depending on the vision rays only.

According to the vision map array and the generated ray map array, a new (15 × 15) array that holds the generated ray physics were created representing the Raycast Map array. During the navigation and at each updated frame, the rays will hit or collide with the objects in the environment producing Ray Distance Map array. All maps are to be dependent on for further calculation during the navigation process. The fifteen columns of rays were rotated about y-axis from $+90^{o}$ to -90^{o} according to their position producing fifteen rotating angles. These fifteen angles will be considered as pathways angles. If the avatar turns away from any visible obstacle, therefor a decision will be made based on the most suitable of the fifteen angles and the clearest half of vision.

As vision, in fact, relies on the central vision field in the first place then relies on the lateral vision field, the navigation algorithm depends on this fact. Therefore, it starts checking the central field for the obstacles in the front paths. The lateral vision can be relied on selecting the proper angle for turning away and to avoid the objects on sides like walls.

The ratio representing the percentage of the view depth to the largest scale in the pattern is calculated. Then the two halves of the vision map values are counted to indicate and select which half of the vision is clearest and to be used for turning decision.

The navigation behaviour goes through three stages:

- Stationary situation: the real person (or avatar) starts navigation from a stopping situation. Even when it is moving, there is a tiny period of time when it transfers its position from step to step. This time is considered as stopping or pausing time. The stationary stage gives an enough time to the navigator to reason the surroundings.
- Reasoning situation: where the person recognizes the surroundings then necessary measurements is calculated to make a decision and take appropriate actions.
- Action situation: it's when the person (or avatar) decides what action to take. These actions can be step forward, stop and wait, or step backward.

These reasoning processes include the procedures of creating and calculating different maps. Therefore the avatar at each updated frame calculates and checks all maps as reasoning before taking the normal action which is to step forward.

5 Implementation and Results

A projection code was attached to the main camera in Unity to open and load the file of numeral values of VF and distribute them to corresponding planes. Then, a desired transparency texture is selected accordingly to project each visual area into specific position on the camera.

Results were taken from a set of VF patterns so that each pattern represents a visual deterioration. Figure 6 shows the original patterns of the PDM symbols taken from three VF chart, and the projected filters for the defect symbols onto a 3D world scene. The above first projection explores a person with clear and normal vision. While the middle projection displays the VF for a totally inferior with almost superior visual deteriorations. But the last projection displays the VF for right-side eyes deterioration.

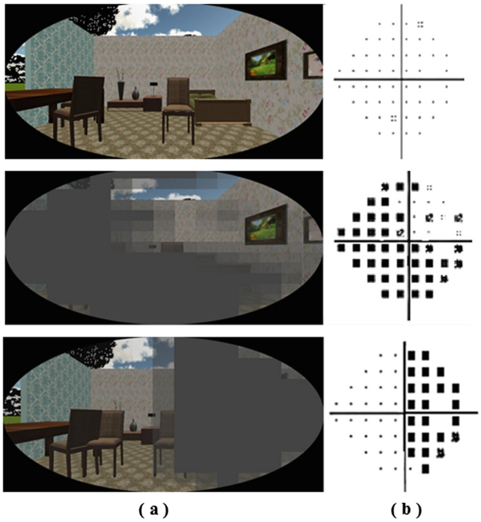
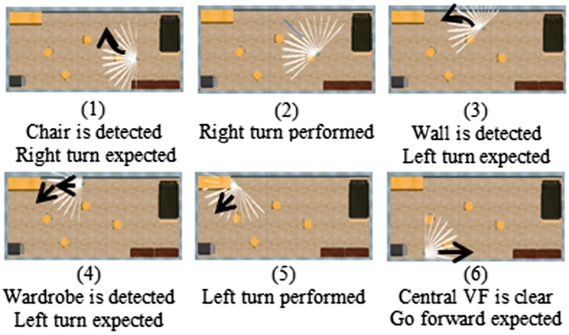


Fig. 6. (a) Simulation of visual impairment-projected VF patterns in a 3D scene and (b) original VF patterns

Three VF charts were projected in the scene and the navigation algorithm on them was executed. For a normal vision person, navigation simulation showed the expected behaviour as a normal person would act. Figure 7 shows the expected behaviour of normal VF person facing an obstacle. The performed actions of the avatar were matched what a real person would act. The images in this figure show a top view of the avatar navigation with its vision rays casting.


(1) Chair is detected Right turn expected

(2) Right turn performed

(3) Wall is detected Left turn expected

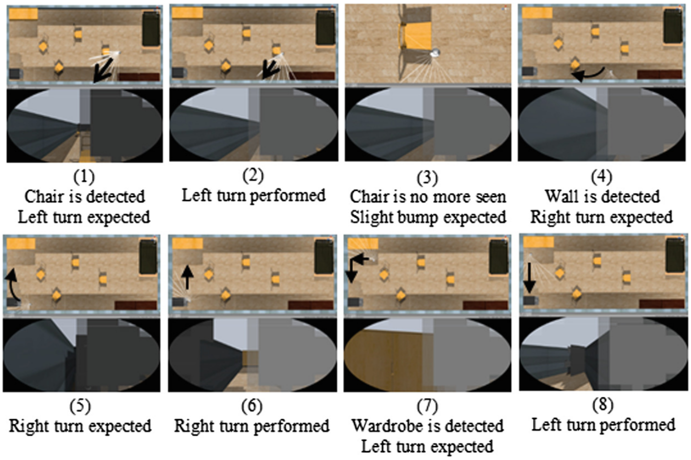
(4) Wardrobe is detected Left turn expected

(5) Left turn performed

(6) Central VF is clear Go forward expected

Fig. 7. The expected and the obtained behaviour for a person having normal vision

Refer to Fig. 5(b), another scenario for a right-side eye defect VF pattern was implemented, see Fig. 8. The camera view window which represents what the avatar sees is attached with the navigation view window representing the top view of the scene looking at the avatar to show what this child exactly sees. This will justify the reason of his behaviour and actions during his navigation. In Fig. 8(4), when a wall is detected, the avatar would not turn to the left (its best vision half) because a left side walk indicator is set. So, the avatar would turn to the right even the right side is totally unseen. This simulates the visually impaired person when he/she expects a path to the unseen side and turn away from it while his best visible side is facing obstacles.


(1) Chair is detected Left turn expected

(2) Left turn performed

(3) Chair is no more seen Slight bump expected

(4) Wall is detected Right turn expected

(5) Right turn expected

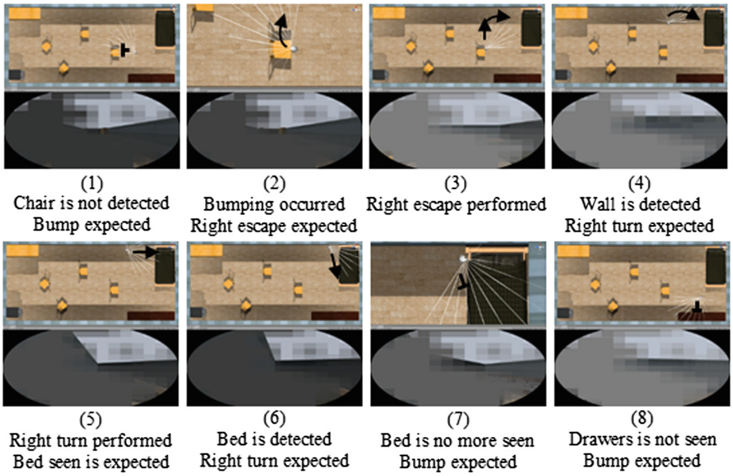
(6) Right turn performed

(7) Wardrobe is detected Left turn expected

(8) Left turn performed

Fig. 8. The expected and obtained behaviour for a person having defects in the right side of the eye

In Fig. 9, the simulation was executed on a situation of a child has the defects in the most of his VF, see Fig. 5(a). Only few spots in his right superior VF are clear. This person and as expected, he will bump with many obstacles but the walls and high objects during his navigation.



(1) Chair is not detected Bump expected

(2) Bumping occurred Right escape expected

(3) Right escape performed

(4) Wall is detected Right turn expected

(5) Right turn performed Bed seen is expected

(6) Bed is detected Right turn expected

(7) Bed is no more seen Bump expected

(8) Drawers is not seen Bump expected

Fig. 9. The expected and obtained behaviour for a person having defects in the inferior and left superior

6 Summary and Conclusions

The probability symbols of vision in the VF chart can be extracted and projected into 3D world to simulate visual impairment behaviours. The visibility regions in the projected world can be presented via different levels of texture transparency representing the impairment areas of vision defects. According to the value of visibility, a visual ray can be initiated, drawn and rotated for visual areas. Thus, the whole created rays will represent a zone of vision beams and can be used to guide navigation behaviors.

The objects in the surroundings can be seen when these rays hit them, while the objects that are not touched by the rays are considered not visible to the avatar. The ray hit and collision properties can facilitate and clarify how the patient can avoid obstacles, and/or how he avoids bumping into them. The coding algorithms can be set according to the normal human navigate behaviour.

The result of applying the algorithms on different VF charts showed different decisions for the turning away process. These decisions were taken depending on the better part of vision, walkable route and the existence of the lateral obstacles.

The current work is to project the 3D scene and the visual impairment defects using the HTC VIVE headset so that people without visual field impairment can experience for themselves what it is like to move around with visual field defects. Experiments are also planned to assess the behavior exhibit by people using the HTC VIVE against the avatar simulation.

Acknowledgment. This work was implemented during Mr Yahya Al-Fadhili was a PhD student at Loughborough University, UK. The authors thank the Iraqi (MOHESR) for the scholarship funding.

References

1. Bernas-Pierce, J., Pollizzi, J., Altmann, C., Lee, B., Hoyt, C.: Cortical visual impairment paediatric visual diagnosis fact sheet. SEE/HEAR. Blind Babies Foundation (1998)
2. Huo, R., Burden, S.K., Hoyt, C.S., Good, W.V.: Chronic cortical visual impairment in children: aetiology, prognosis, and associated neurological deficits. Br. J. Ophthalmol. **83**(6), 670–675 (1999)
3. Dutton, G.N., Saeed, A., Fahad, B., et al.: The association of binocular lower vision field impairment, impaired simultaneous perception, disordered visually guided motion and inaccurate saccades in children with cerebral visual dysfunction - a retrospective observational study. Eye **18**, 27–34 (2004)
4. Swift, S.H., Davidson, R.C., Weens, L.J.: Cortical impairment in children: presentation, intervention, and prognosis in educational settings. Teach. Except. Child. Plus **4**(5) (2008). Article 4
5. Nilsson, I.L., Lindberg, W.V.: Visual Perception: New Research. Nova Science Pub Incorporated, New York (2008)
6. Dutton, G.N.: Strategies for dealing with visual problems due to cerebral visual impairment. Scottish Sensory Centre (2008)
7. Duffy, M.: Google Glass Applications for Blind and Visually Impaired Users. VisionAware Blog, 1 May 2013
8. Owano, N.: OpenGlass apps show support for visually impaired. PHYSORG (2013)
9. Iannizzotto, G., et al.: Badge3D for visually impaired. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005) (2005)
10. Glen, F.C., Smith, N.D., Crabb, D.P.: (2014). What types of visual field defects are hazardous for driving? ARVO Annu. Meet. Abstr. (2014)
11. Crabb, D.P.: Personal Communication, with Professor David Crabb, School of Health Sciences, Division of Optometry & Visual Sciences, City University, London, UK (2014)
12. Vapnik, N.: Statistical Learning Theory, pp. 421–422. Wiley, New York (1998)
13. Cortes, C., Vapnik, N.: Support-vector-networks. Mach. Learn. **20**(3), 273 (1995)
14. Spector, R.H.: Visual Fields. In: Walker H.K., Hall, W.D., Hurst, J.W., (eds.) Clinical Methods: The History, Physical, and Laboratory Examinations, chap. 116 (1990)

A Fall Detection/Recognition System and an Empirical Study of Gradient-Based Feature Extraction Approaches

Ryan Cameron, Zheming Zuo, Graham Sexton, and Longzhi Yang(✉)

Department of Computer and Information Sciences, Northumbria University,
Newcastle upon Tyne NE1 8ST, UK
{ryan.cameron,zheming.zuo,g.sexton,longzhi.yang}@northumbria.ac.uk

Abstract. Physically falling down amongst the elder helpless party is one of the most intractable issues in the era of ageing society, which has attracted intensive attentions in academia ranging from clinical research to computer vision studies. This paper proposes a fall detection/recognition system within the realm of computer vision. The proposed system integrates a group of gradient-based local visual feature extraction approaches, including histogram of oriented gradients (HOG), histogram of motion gradients (HMG), histogram of optical flow (HOF), and motion boundary histograms (MBH). A comparative study of the descriptors with the support of an artificial neural network was conducted based on an in-house captured dataset. The experimental results demonstrated the effectiveness of the proposed system and the power of these descriptors in real-world applications.

Keywords: Fall detection · Local feature extraction · HOG · HMG · HOF · MBH · Artificial neural network

1 Introduction

Around one in three elders over the age of 65 living at home have at least one unexpected fall per year and this is doubled for those aged over 75 [1]. These falls can be serious or even fatal without timely medical help. Therefore, increasingly more research efforts have been spent on automatic fall detection, which can be grouped into three categories: audio-based, wearable sensors-based, and vision-based [2]. Audio-based fall detection systems usually suffer from the lack of reliable training data, complicated configuration process, and sensitivity to background noise [3]. Differently, wearable sensor-based systems (e.g., [4]) often experience difficulties in deploying sensors to participants (as participants are usually reluctant to wear sensors) and poor performance without the involvement of visual action features due to the physical limitations of sensors.

Vision-based systems therefore become a good choice, which may also be used for fall type recognition as different types of falls usually require different forms

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_24

of medical help. Various vision-based approaches have been proposed in the literature with different types of features used, such as silhouette-related features of human body [5], which can be either created using background separation method [6] or tracked using shape matching [7]. The background separation (or background noise removal) approach may also eliminate some useful or discriminative visual features during the background removal process. The shape matching (or template matching) approach is apt to over-fitting as it is based on a specific model though it is fast. Most recently, depth information is becoming a trend to be involved for extracting more accurate features of fall motions such as RGB-D that combines a RGB and depth information [8].

This paper proposes a fall detection and fall type recognition system, which is able to deal with raw colour or infrared data without the use of depth information (knowing that depth cameras are still not commonly deployed). Given a number of video clips, the proposed framework firstly extracts the spatial/temporal information using a gradient-based local feature extraction approach, then with some pre-processing techniques involved, an artificial neural network (ANN) is applied to the extracted features for classification. Note that there is a family of gradient-based approaches available, including the very recently proposed histogram of motion gradients (HMG) [9], the very first approach in this kind the histogram of oriented gradients (HOG) [10], histogram of optical flow (HOF) [11] and its extension motion boundary histograms (MBH) [12]. The proposed framework therefore integrates all these approaches for a comparative study on their effectiveness and efficiency in fall detection/recognition.

This work also made a publicly available fall detection/recognition data set captured in various lighting conditions to support the development of a real-time system. In order to enable the system working in any lighting conditions, both colour and infrared (IR sensor) information were captured separately using Microsoft Kinect sensor during day and night times. Note that only low resolution of 320 * 240 was applied for both colour and IR data sets in order to simulate the widely used CCTV systems, although Microsoft Kinect supports high resolution up to 1920 * 1080. Both data sets were utilised for ANN classifier training and for system validation by employing the cross validation approach. In order to get optimal solutions, the parameters of ANN were also empirically studied in the experimentation section.

The rest of the paper is organised as follows. The related work is reviewed in Sect. 2. The proposed system is presented in Sect. 3. The data and data capturing process are discussed in Sect. 4. Experimentation is reported in Sect. 5, and this work is concluded in Sect. 6 with future work pointed out.

2 Related Work

The three types of fall detection systems are reviewed in this section. Audio-based fall detection systems ensure the elderly helpless party monitored are not required to wear uncomfortable equipment as compared with wearable sensors. Furthermore, the performances of such systems are not affected by occlusions, from the perspective of computer vision. However, this technique suffers from a number of problems such as difficulty in creating reliable training data, complicated configuration process, and sensitivity to background noise [3]. Reliable training data is difficult to produce as people (elder seniors in particular) are reluctant to fall force on a hard floor, which may be necessary to replicate the sounds and vibrations of a real fall situation [13]. Additionally, the configuration process is particularly difficult and complicated in comparison to other fall detection systems, as the implementation of audio-based systems are usually environment-oriented [3].

Conventionally, sensory data collected from different sources are widely researched and implemented in action detection and recognition tasks [14]. The automatic fall detection systems propsoed in [15–17] are reliant on an integrated wearable accelerometer sensor [15–17], or a combination of accelerometers and gyroscopes in [16]. Dissimilar to the audio-based approaches, these systems (e.g., [4]) often experience difficulties in deploying sensors to participants and poor performance without the involvement of visual action features due to the physical limitations of sensors. For example, the accelerometer sensor may not accurately detect the series of events when a patient with slow heart failure sitting down to a chair at home alone. The gyroscopes used in [16] rely on an assigned threshold value to distinguish fall and non-fall actions, which is also the case for [15,18].

Vision-based systems recognise falls using the widely applied low-cost cameras. The majority pieces of work in this area are largely dependent on the removal of background noise and extracting only the foreground information by applying edge detection. The work of [19] extracts the features based on the coordinate data of the skeleton with large proportions of pixels and related spatial information ignored, and it applies fuzzy logic to determine the pose of the person. In [20], an occlusion robust method in indoor environment was proposed. This method uses the combined information of human centroid height relative to the ground and 3D human body velocity. The prior feature provides supplementary information when falls end on ground or nearby but is not sufficient when the fall is only partially visible to the camera (occluded). This problem was addressed by computing 3D human velocity prior to the occlusion. Similarly, in [21], the background is firstly subtracted to allow for useful motion and body posture information to be extracted and combined, using the timed Motion History Image method and an approximated ellipse around the human body respectively. However, these two pieces of work also share the situation that falls create large velocities or motions cannot be made in the case of slow falls, sharing a similar drawback to the use of accelerometers for fall detection. The shape matching aspect is also apt to over-fitting as it is based on a specific model, though it is fast. Most recently, depth information is becoming a trend to be involved for extracting more accurate features of fall motions such as RGB-D that combines a RGB and depth information [8].

3 Fall Detection and Action Recognition

The proposed fall detection and action recognition system is outlined in Fig. 1. It is an implementation of the well-known bag of features (BOF) framework [22] to support fall detection/recognition, which usually contains three major components including feature extraction, feature encoding, and classification. In this implementation, all the data instances (i.e., video or infrared clips) are processed first to be of the same length (or the number of frames), and thus feature encoding is omitted for simplicity and computational efficiency. In particular, each video clip is represented as a three-dimensional array of pixels with time being the third dimension. Following the modelling paradigm [23], each such array is usually divided into a number of blocks with varying number of frames (e.g., 1, 2, 6, or more) included in each block. A larger number of video frames per block means denser extracted features. Then, the extracted features are pre-processed using RootSIFT and PCA for data normalisation and dimensionality reduction, respectively. The generated features vectors are finally taken as inputs by an ANN classifier for fall detection/recognition.

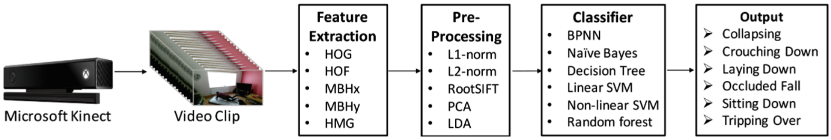


Fig. 1. The general framework for fall detection

3.1 Feature Extraction

Five histogram-based local feature extraction approaches, including HOG, HOF, MBHx, MBHy, and HMG, are integrated in the framework. Note that HMG is an enhancement of HOG as the acquisition of temporal motion information is considered, whereas HOF and MBH share the same process in calculating the optical flow. Therefore, HOG and HMG are introduced first, followed by HOF and MBH. Suppose that a video clip is formed by n frames $\{F_1, F_2, \cdots, F_n\}$. HOG calculates the gradient values of pixels for every frame F_i $(1 \leq i \leq n)$ and the resulting array for each frame is denoted as RF_i. This is usually achieved by applying a fast convolution operation on each frame with filter kernels of [-1, 0, 1] and $[-1, 0, 1]^T$ (i.e., Haar [24]). HMG enhances HOG by performing extra effective yet simple temporal derivative calculation prior to the HOG process on each adjacent pair of frames (F_j, F_{j+1}), $j \in \{1, 2, \cdots, n-1\}$, and denote the result TD_j. This operation allows the motion information, i.e., the temporal information, to be considered [9]. As such, HMG considers both temporal and spatial information.

The temporal motion in HMG can also be represented using the optical flow as demonstrated in HOF and MBH. Briefly, optical flow is the appearance motion of brightness patterns. It is usually represented as a complex vector whose magnitude and direction can be readily calculated to support the construction of histogram of gradients as discussed later. For efficiency, the Horn-Schunck (HS) [25] method is employed in this work for the calculation of the optical flow vectors

$\boldsymbol{F}_k (k \in \{1, 2, \cdots, n-1\})$, although other approaches such as Lucas-Kanade [26] can also be used. MBH differs from HOF in the extra step of obtaining gradients in vertical and horizontal directions. Given a complex optical flow matrix $\boldsymbol{F}_k$ calculated from adjacent video frames R_k and R_{k+1}, MBHx only takes its imaginary part, denoted as $Im(\boldsymbol{F}_k)$ for further calculation, whilst MBHy only takes the real part of $\boldsymbol{F}_k$, denoted as $Re(\boldsymbol{F}_k)$.

From this, the computation of gradients in the horizontal and vertical directions for HOG, HMG, MBHx (horizontal only) and MBHy (vertical only) can be jointly represented as:

$$Hor = \frac{\partial R}{\partial x} \qquad Ver = \frac{\partial R}{\partial y}, \tag{1}$$

where R denotes RF, TD, $\boldsymbol{F}_k$, $Im(\boldsymbol{F}_k)$ or $Re(\boldsymbol{F}_k)$ in HOG, HMG, HOF, MBHx or MBHy, respectively. From this, the magnitude (Mag) and orientation (θ) can be calculated as:

$$Mag = \sqrt{Hor^2 + Ver^2} \qquad \theta = arctan(\frac{Ver}{Hor}). \tag{2}$$

Note that the magnitude (Mag) and orientation (θ) in HOF can be readily calculated from the complex flow vector $\boldsymbol{F}_k$. Once the Mag and θ for each frame are obtained for the histogram-based approaches, the frame magnitude responses can be quantised into a number of orientation bins evenly spread over 0 to 2π. This is implemented using the bi-linear interpolation approach [27], which determines the contribution of each pixel to each bin based on its orientation. Then, the quantised responses are aggregated over blocks in the 3D feature space. The final extracted features can be generated by concatenating the responses over several adjacent blocks [9]. An empirical performance study using different number of frames per block is reported in Sect. 5.3.

3.2 Pre-processing

After features are extracted, RootSIFT [28] is applied for normalization. Specifically, in order to reduce computational complexity and achieve fast convergence in the later classifier training phase, the least absolute deviations (i.e., L1 normalization) is performed first to normalise the extracted features. This is followed by conducting extra square root (Hellinger kernel) to measure the similarity between these L1 normalised extracted features. Such combination of operations is jointly referred to as RootSIFT normalisation [28]. The response of a pixel is dominated by the closer bins in the bi-linear interpolation process [27]. The effect of RootSIFT is assigning a relatively bigger response value to the farther bin and thus mitigating the dominance from the closer bin. The inclusion of this process in the framework helps in achieving constant variance of the polynomial bin value, rather than being linearly dependent on the mean histogram.

PCA is employed for dimensionality reduction in this work, which is one of the most well-known multivariate analysis techniques [29]. In particular, PCA uses relatively smaller number of decorrelated dimensions of features to describe the feature space, by retaining the most closely correlated information and removing redundancy. The general procedure of PCA starts with calculating the covariance matrix for a given input of normalised features. It then computes the eigenvector with a number of the largest eigenvalues sorted in descending order. From this, the normalised features are projected to the lower dimensional feature space in line with the obtained eigenvector with large eigenvalues. The number of feature dimensions after PCA operation were investigated with results reported in Sect. 5.1.

3.3 Classifier

ANNs can be viewed as cooperated parallel computing machines which comprise of a certain number of simple structure processing units (i.e., neurons) that connected together for dealing with a single complex problem by learning and generalising from training data instances [30]. Amongst a number of ANNs, backpropagation neural network (BPNN) has become a common choice in tackling a large range of problems under supervised learning paradigm, due to its exceptional functional approximation ability [31]. In particular, the architecture of the BPNN used in this work comprises of three layers: input layer, hidden layer, and output layer. The input layer takes the extracted features for processing and the output layer recommends the action classification result. The hidden neurons enable the network to learn complex tasks by extracting progressively more meaningful features from the input vector [32].

The performance of BPNN relies heavily on a number of parameters including the number of hidden neurons, the learning rate, and the momentum. Too few neurons in the hidden layer usually lead to models which are unable to detect patterns correctly from the training data (i.e., under-fitting), whereas too many neurons often lead to unnecessarily long training time and poor generalisation ability (i.e., over-fitting). Learning by a neural network is essentially the process of minimising a global error function by an iterative process. As a quadratic approximation to the error function in the neighbourhood of the current data point in the weight space, the scaled conjugate gradients (SCG) supervised training algorithm is used in this work to avoid computationally expensive line-search [33]. In particular, SCG features fast convergence with stable performance, that is the performance does not degrade quickly along with the training time. The investigation on various number of hidden neurons is reported in Sect. 5.2.

4 Data Set

A fall and similar daily activity action dataset was captured for fall detection and action recognition. The data set includes video clips in three types of falls: collapsing, tripping over and occluded fall. Note that the type of falls provide crucial information in determining the severity level of falls and thus the types of health assistance needed. Also note that other home activities may be also useful in providing health services; three activities of daily life classes are also included in the data set, including crouching down, sitting and laying down. As there are 6 classes included in the data set, the data set is named as UNN6, which can be downloaded from www.lyang.uk/unn6-2017. The involvement of similar activities of daily life videos to fall actions also increase the difficulty of action detection and motion recognition tasks for more accurate system evaluation.

The UNN6 includes a colour clip set and an infrared (IR) clip set, with sample frames shown in Fig. 2. Each set contains 36 clips (i.e., 6 videos per class) that were recorded using the Microsoft Kinect v2 in an indoor environment with varying lighting conditions and dynamic backgrounds (e.g., moving pictures from TV in the background). For simulating real-world CCTV system and improving the computational efficiency of feature extraction, the resolution for all video clips are uniformly resized from 1920 * 1080 to 320 * 240. Each video clip in both sets are cropped into 150 frames with 25 frame rate applied, i.e., each video lasts for 6 s as a fall or similar action always happens within 6 s. The removal of the unnecessary parts may increase the recognition results, but it is difficult to accurately recognise the start and the end point of an action. In addition, the generated IR clips do not include any depth information, as depth cameras are still costly and not widely deployed. However, it is interesting to extend the data set in the future to include another subset of depth streams.

The data capturing process usually expects a large group of participants with different age, gender and body images, which helps in decreasing the difficulty of detection and improving the model generalisation capability for classification. However, as the work reported here is mainly based on an undergraduate final year project, the data was captured to support the development of a real-time fall detection and action recognition system, and to investigate the effectiveness and efficiency of the five local features descriptors in handling such task. Nevertheless, this dataset can also be a qualified candidate for a wide range of research communities including clinical science, bio-medical, computer vision, amongst other.

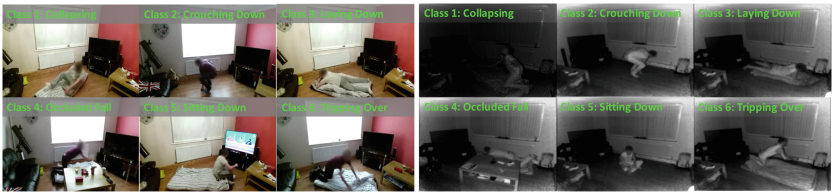


Fig. 2. Sample colour frames (left) and IR frames (right) in UNN6 dataset

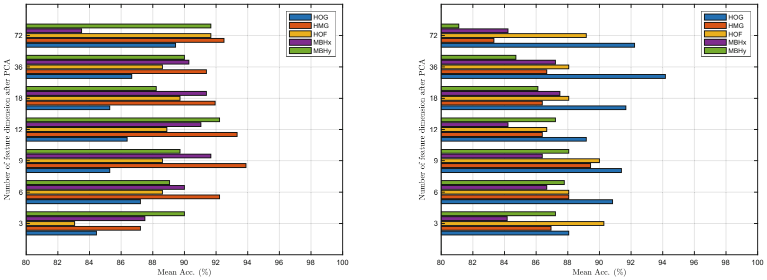
5 Experimentation

The proposed system was evaluated using data set UNN6. Also, an empirical comparative study was conducted to validate the five local feature descriptors (i.e., HOG, HMG, HOF, MBHx, MBHy) in fall detection and action recognition. All the experiments were carried out using Intel Core i7-4790 CPU @ 3.60GHz. The experimental accuracy was calculated as the average of 10 independent runs. In particular, the proportion of video clips used for training, testing, and validation are set to 70%, 15%, and 15%, respectively. The recognition performance of the five descriptors were studied using the colour and IR data sets separately. For all the experiments, 8 bins were used for all the descriptors; the learning rate and momentum for BPNN are automatically learned by the application of SCG. In particular, there are two parameters used in SCG, including the sigma which controls the changing weight for 2^{nd} order derivative approximation and lambda which regulates the uncertainty parameter in the Hessian matrix). The values of lambda and sigma in the experimentation were set to 5.0e-7 and 5.0e-5, respectively. The optimal performances were acquired by empirically investigating different parameters in three experiments, including the number of feature dimensions after PCA operation, the number of hidden neurons in the single hidden layer BPNN, and the number of frames included in each block, which are detailed below.

5.1 Experiment 1

The first experiment investigated the effect of varying the number of feature dimensions (FDs) after the PCA operation, by taking 1 frame per block during feature extraction and 30 hidden neurons in the BPNN. The performance of the colour set is shown in Fig. 3(a). Interestingly, HMG reached the highest mean performance of 93.89% using the smallest number of FDs (i.e., 9 FDs) within all five approaches, and MBHx also reached its peak performance of 91.67% using 9 FDs. However, the closest competitor MBHy achieved the best performance of 92.22% with more features, i.e., 12 FDs. In contrast, HOF and HOG achieved the best performance of 91.67% and 89.44 with 72 FDs used, which is the maximum number of features investigated in this experiment. It can be concluded that different local feature descriptors are of different noise tolerance, summarisation and generalisation ability in representing motions. This also well proved the argument that more features do not necessarily lead to better performance as more features may also introduce bigger noise [34].

The performances based on the IR set, as show in Fig. 3(b), were generally worse than those based on the colour set. The results led by the HOG have continually improved considerably using from 3 to 36 FDs except the use of 12 FDs, and HOG with 36 FDs reached the highest mean accuracy of 94.17% within all the descriptors. The rest four descriptors generally share similar increasing trend in the FDs ranging from 3 to 9 FDs and performances are decreased after 9 FDs. Using 9 FDs, HOG attained 91.39%; the peak performance were achieved in HMG (89.44%) and MBHy (88.06%); and moderate performance were obtained in HOF and MBHx with the accuracies of 90.00% and 86.39%, respectively. The time consumption against different numbers of features has also been investigated as shown in Fig. 4, due to its importance for the implementation of real-time system. In particular, the time was measured

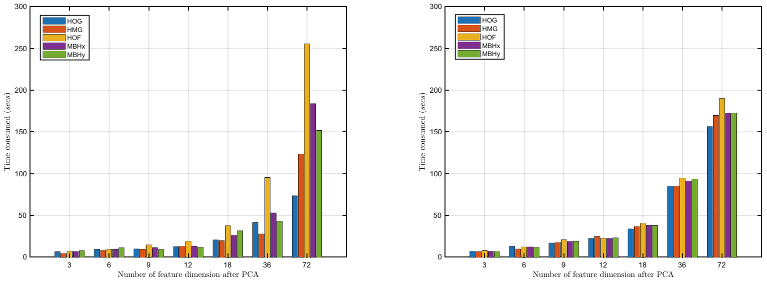


(a) Colour set

(b) IR set

Fig. 3. Classification performance on UNN6 with various number of features

by adding the PCA operation time plus the training and testing time of BPNN. From this figure, it is clear that time consumption increases exponentially with the increase FDs. Note that the use of 9 FDs has led competitive performance with reasonable time expense, and thus 9 FDs were used in the rest of the experimentation.



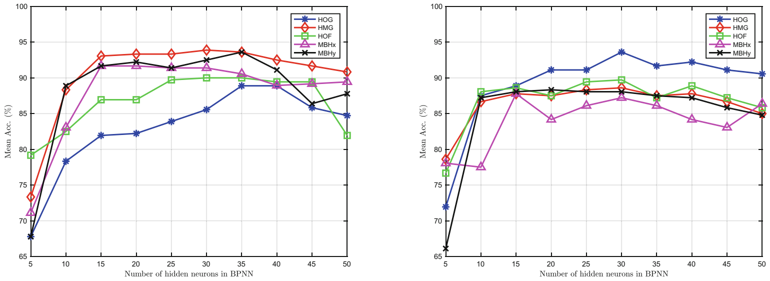
(a) Colour set

(b) IR set

Fig. 4. The time consumption on UNN6 with various number of features

5.2 Experiment 2

The number of hidden neurons was investigated in this section, with 9 FDs and 1 frame per block applied for feature extraction. The performance of the system using 5 to 50 neurons with a step of 5 is reported in Fig. 5. This figure shows a steady, steep increase in the mean classification accuracy from under fitting with only 5 hidden neurons up to 15 for the colour set and 20 for the IR set for all feature descriptors. At this point, the overall performance incline begins to slow and level out at its pinnacle which varies between 20 and 30 hidden neurons in general, with HOG and MBHy in the colour setting being two exceptions, requiring a larger number (i.e., 35) of hidden neurons for optimal performance.



(a) Colour set

(b) IR set

Fig. 5. Mean classification accuracy with various number of hidden neurons in BPNN

HMG yielded the best average results in the colour scenario, classifying 94.76% of videos correctly when using 30 hidden neurons; in this case MBHy was the closest competitor with 93.61% accuracy, and with 91.67%, 90.00% and 88.89% for MBHx, HOF and HOG, respectively. In the IR scenario, HOG, HMG, and HOF achieved the highest performance with 93.61%, 88.61%, 89.72%, using 30 hidden neurons, whilst MBHx and MBHy attained the best performance with 87.78% and 88.33% using 15 and 20 hidden neurons and this performance almost re-occurred when 30 hidden neurons used. For both cases, 30 hidden neurons typically led the optimal performance for the five feature extraction methods, before over fitting occurred.

5.3 Experiment 3

The third experiment investigated the possible redundancy that may occur during the process of extracting local features by changing the number of frames included in each block, followed the idea of dense features in [9], with the results listed in Table 1. It is presented that HMG, HOF and MBHx achieved 94.72% classification accuracies, with MBHy achieving 93.89%, and HOG achieving 90.28%, when using the colour data set. Noteworthy, HMG reached its peak performance with a lesser number of frames per block in comparison to all others (6 frames per block is used in HOF and MBHx). It is also shown that HMG outperformed all other four descriptors regardless of the number of frames per block.

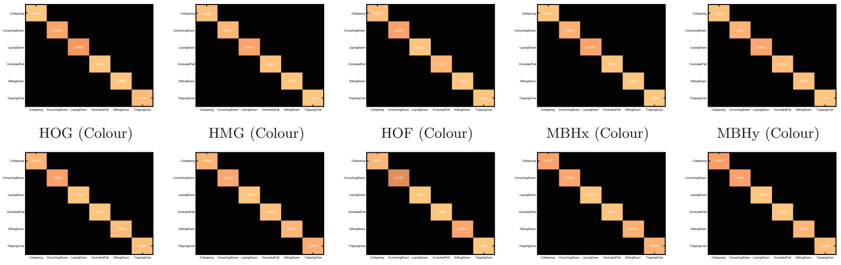
When it comes to the results from the IR set, HOG outperformed all other four descriptors including its temporal extension HMG (91.39%) and arrived at the best mean accuracy of 93.61% with six frame per block used. Additionally, in the case of only one frame per block considered, HOG (93.06%) is also performed better than all the rest best performance of its competitors on IR set. If any number of frames per block was allowed, HOF achieved the accuracy of 91.11% which outperformed its two extensions MBHx and MBHy with accuracies of 90.00% and 90.83%. Also, the temporal derivative approach employed in HMG has been proved to be computationally more efficient for motion temporal information calculation than the HS method used in HOF, MBHx, and MBHy.

Table 1. Classification performance using five local descriptors with a varied number of frames used per block

	UNN6: Colour (%)					UNN6: IR (%)				
	HOG	HMG	HOF	MBHx	MBHy	HOG	HMG	HOF	MBHx	MBHy
1	86.67	93.33	90.56	92.78	92.22	93.06	88.61	89.72	87.22	88.06
2	87.50	93.33	91.39	91.67	92.89	91.11	88.61	90.28	88.33	88.33
3	86.94	**94.72**	93.33	93.33	92.22	91.39	88.72	**91.11**	88.06	**90.83**
6	**90.28**	94.72	**94.72**	**94.72**	**93.89**	**93.61**	**91.39**	90.56	**90.00**	89.44

5.4 Discussions

Following the three experiments, the best performances for all five descriptors using both colour and IR data sets are summarised in Fig. 6. Based on the colour set, HMG attained the best mean accuracy of 94.72% with only 19 misclassified among 360 predictions (given that 10 independent runs were performed). Specifically, 11 of the 60 classification tasks for motion 'LayingDown' were wrongly predicted, which dramatically pulled down the overall performance. Similarly, except the HOF which misclassified only 1 in 60 prediction tasks, and all the rest four descriptors were not performed well for this class, i.e., 15/60 in HOG, 11/60 in MBHx and 10/60 in MBHy were misclassified. This is because high similarity between actions 'SittingDown' and 'LayingDown', and HMG misclassified 'LayingDown' as 'SittingDown' for 9 times within its 11 misclassification. In addition, there were 3 false negatives ('Collapsing' and 'OccludedFall' being wrongly predicted as 'LayingDown' for twice and once respectively), and 3 false positives ('LayingDown' wrongly classified as 'TrippingOver' for 2 times and 'Collapsing' for once). 'SittingDown' and 'TrippingOver' have been classified 100% successfully by HMG, HOF, and MBHx.



HOG (Colour) HMG (Colour) HOF (Colour) MBHx (Colour) MBHy (Colour)

HOG (IR) HMG (IR) HOF (IR) MBHx (IR) MBHy (IR)

Fig. 6. Final confusion matrices with parameters optimised on UNN6 using five feature descriptors. For colour set (upper row), HOG (90.28%), HMG (94.72%), HOF (94.72%), MBHx (94.72%) and MBHy (93.89%). For IR set (bottom row), HOG (93.61%), HMG (91.39%), HOF (91.11%), MBHx (90.00%) and MBHy (90.83%).

Regarding the IR set, HOG overall achieved the highest average accuracy of 93.61% using six frames per block. The 'CrouchingDown' class within all the six classes was most frequently misclassified, with 9/60 misclassified by MBHx, 11/60 misclassified by HMG, 12/60 misclassified by MBHy and HOG, and 17/60 misclassified by HOF. Note that the 20% misclassified 'CrouchingDown' motions were all mistakenly recognised as 'SittingDown' by the HOG, which was also common seen for all other descriptors in both colour and IR sets. Also, 'Collapsing' was misclassified 6 times as 'OccludedFall'; 'SittingDown' was mistakenly classified as 'CrouchingDown' for 3 times; and 'TrippingOver' was misclassified as 'SittingDown' twice.

To summarise, the HOG with the optimal parameter set in the experimentation scored 2 false negatives and 0 false positives, and it classified motions 'LayingDown' and 'OccludedFall' with 100% accuracies. Also, 'LayingDown' and 'OccludedFall' were classified 100% successfully by MBHy, motions 'LayingDown', 'OccludedFall' and 'TrippingOver' by HOF, and motion 'LayingDown' only by HMG and MBHx.

6 Conclusions

An automatic fall detection system was proposed in this paper implemented using gradient-based local feature extraction approaches and an artificial neural network. Also, the group of local feature extraction approaches integrated in the proposed system is empirically studied using a purposely captured fall detection/recognition data set in two formats (i.e., colour and infrared) to simulate various lighting conditions. The experimental results demonstrated the effectiveness of the proposed system and the power of the gradient-based local feature extraction approaches in handling fall detection/recognition tasks. Compared to the existing approaches, the proposed system is fully automatic, only uses low resolution motion information, and does not require noise reduction process. Consequently, it is able to perform in real-time which is of significant importance in practical use. Although promising, the proposed system needs to be evaluated with more datasets such as CDNET [35]. Also, it is interesting to compare the proposed approach with some existing ones such as [21]. Better performance is expected for the system by taking both colour and infrared information, which required further investigation. In addition, it is interesting to evaluate the approach when part of the body is occluded by a furniture.

References

1. Litvak, D., Zigel, Y., Gannot, I.: Fall detection of elderly through floor vibrations and sound. In: 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 4632–4635 (2008)
2. Zhu, L., Zhou, P., Pan, A., Guo, J., Sun, W., Wang, L., Chen, X., Liu, Z.: A survey of fall detection algorithm for elderly health monitoring. In: 2015 IEEE Fifth International Conference on Big Data and Cloud Computing, pp. 270–274 (2015)

3. Mohamed, O., Choi, H.J., Iraqi, Y.: Fall detection systems for elderly care: a survey. In: 2014 6th International Conference on New Technologies, Mobility and Security (NTMS), pp. 1–4 (2014)
4. AlZubi, H.S., Gerrard-Longworth, S., Al-Nuaimy, W., Goulermas, Y., Preece, S.: Human activity classification using a single accelerometer. In: 2014 14th UK Workshop on Computational Intelligence (UKCI), pp. 1–6. IEEE (2014)
5. Zhang, Z., Conly, C., Athitsos, V.: A survey on vision-based fall detection. In: Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments, p. 46. ACM (2015)
6. Mirmahboub, B., Samavi, S., Karimi, N., Shirani, S.: Automatic monocular system for human fall detection based on variations in silhouette area. IEEE Trans. Biomed. Eng. **60**(2), 427–436 (2013)
7. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Robust video surveillance for fall detection based on human shape deformation. IEEE Trans. Circuits Syst. Video Technol. **21**(5), 611–622 (2011)
8. Merrouche, F., Baha, N.: Depth camera based fall detection using human shape and movement. In: IEEE International Conference on Signal and Image Processing (ICSIP), pp. 586–590. IEEE (2016)
9. Duta, I.C., Uijlings, J.R.R., Nguyen, T.A., Aizawa, K., Hauptmann, A.G., Ionescu, B., Sebe, N.: Histograms of motion gradients for real-time video classification. In: 2016 14th International Workshop on Content-Based Multimedia Indexing (CBMI), pp. 1–6. IEEE (2016)
10. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005, vol. 1, pp. 886–893. IEEE (2005)
11. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR 2008, pp. 1–8, June 2008
12. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 428–441. Springer, Heidelberg (2006). doi:10.1007/11744047_33
13. Popescu, M., Mahnot, A.: Acoustic fall detection using one-class classifiers. In: 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 3505–3508 (2009)
14. Delahoz, Y.S., Labrador, M.A.: Survey on fall detection and fall prevention using wearable and external sensors. Sensors **14**(10), 19806–19842 (2014)
15. Hwang, J.Y., Kang, J.M., Jang, Y.W., Kim, H.C.: Development of novel algorithm and real-time monitoring ambulatory system using bluetooth module for fall detection in the elderly. In: 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2004. IEMBS 2004, vol. 1, pp. 2204–2207. IEEE (2004)
16. Luštrek, M., Kaluža, B.: Fall detection and activity recognition with machine learning. Informatica **33**(2), 205–212 (2009)
17. Kerdegari, H., Samsudin, K., Ramli, A.R., Mokaram, S.: Evaluation of fall detection classification approaches. In: 2012 4th International Conference on Intelligent and Advanced Systems (ICIAS), vol. 1, pp. 131–136. IEEE (2012)
18. Degen, T., Jaeckel, H., Rufer, M., Wyss, S.: Speedy: a fall detector in a wrist watch. In: ISWC, pp. 184–189 (2003)
19. Planinc, R., Kampel, M.: Introducing the use of depth data for fall detection. Personal Ubiquit. Comput. **17**(6), 1063–1072 (2013)

20. Rougier, C., Auvinet, E., Rousseau, J., Mignotte, M., Meunier, J.: Fall detection from depth map video sequences. In: Toward useful Services for Elderly and People with Disabilities, pp. 121–128 (2011)
21. Albawendi, S., Appiah, K., Powell, H., Lotfi, A.: Video based fall detection with enhanced motion history images. In: Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments, p. 29. ACM (2016)
22. Uijlings, J., Duta, I.C., Sangineto, E., Sebe, N.: Video classification with densely extracted hog/hof/mbh features: an evaluation of the accuracy/computational efficiency trade-off. Int. J. Multimed. Inf. Retr. **4**(1), 33–44 (2015)
23. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision (IJCV) **60**(2), 91–110 (2004)
24. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR 2001, vol. 1, pp. I-511-I-518 (2001)
25. Horn, B.K., Schunck, B.G.: Determining optical flow. Artif. Intell. **17**(1–3), 185–203 (1981)
26. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence - vol. 2, IJCAI 1981, pp. 674–679. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1981)
27. Wang, S., Yang, K.J.: An image scaling algorithm based on bilinear interpolation with VC++. Tech. Autom. Appl. **27**(7), 44–45 (2008)
28. Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2911–2918. IEEE (2012)
29. Jolliffe, I.: Principal Component Analysis. Wiley, Hoboken (2002)
30. Haykin, S.S.: Neural Networks and Learning Machines. Prentice Hall, Upper Saddle River (2009)
31. Liu, Y., Jing, W., Xu, L.: Parallelizing backpropagation neural network using mapreduce and cascading model. Comput. Intell. Neurosci. **2016**, 2842780:1–2842780:11 (2016)
32. De Villiers, J., Barnard, E.: Backpropagation neural nets with one and two hidden layers. IEEE Trans. Neural Netw. **4**(1), 136–141 (1993)
33. Møller, M.F.: A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw. **6**(4), 525–533 (1993)
34. Jensen, R., Shen, Q.: Are more features better? A response to attributes reduction using fuzzy rough sets. IEEE Trans. Fuzzy Syst. **17**(6), 1456–1458 (2009)
35. Goyette, N., Jodoin, P.M., Porikli, F., Konrad, J., Ishwar, P.: Changedetection.net: a new change detection benchmark dataset. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1–8. IEEE (2012)

Towards an Ontology of Trust for Situational Understanding

Owain Carpanini and Federico Cerutti[✉]

Cardiff University, Cardiff, UK
CeruttiF@cardiff.ac.uk

Abstract. In this paper we propose a computational methodology for assessing the impact of trust associated to sources of information in situational understanding activities—i.e. relating relevant information and form logical conclusions, as well as identifying gaps in information in order to answer a given query. Often trust in the source of information serves as a proxy for evaluating the quality of the information itself, especially in the cases of information overhead. We show how our computational methodology support human analysts in situational understanding by drawing conclusions from defaults, as well as highlighting issues that demand further investigation.

Keywords: Computational models of trust · Situational understanding · Uncertainty

1 Introduction

Individuals and organisations have access to a rich and diverse source of information that can be exploited for situational understanding activities—i.e. relating relevant information and form logical conclusions, as well as identifying gaps in information in order to answer a given query. However, an open and enduring problem resides in managing the impact of trust measurements in such activities.

We propose a computational methodology for assessing the impact of trust in situational understanding to support human analysts with a sound ontology and the ability to reason with partial information. While the aspect of reasoning with partial information shares similarities with other approaches to qualitative decision making [5], to our knowledge in this paper we propose the first ontology for trust in situational understanding. We ground our preliminary investigation in a case study on the Wakefield case on the alleged links between vaccination and autism that will help us describing our desiderata for supporting human analysts (Sect. 2). We then propose a methodology satisfying those desiderata in Sect. 3, and we illustrate it developing further our case-study. As this preliminary work is part of an ongoing investigation, we will extensively discuss future directions in Sect. 4.

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_25

2 Motivational Scenario and Desiderata

In [12] (now retracted), Wakefield et al. suggest a link between some vaccinations and autism. This paper triggered extensive studies that resulted in a retraction notice [10] that states: “it has become clear that several elements of the 1998 paper by Wakefield et al. are incorrect.”

Let us suppose that an analyst needs to answer the question “Do vaccinations cause autism?” (situation understanding task). For simplicity let us suppose they have access only to [12] and [10]: both are scientific papers discussing a medical topic, thus the analyst might assume that both of them can be highly trusted.

Our first desideratum is therefore (**Des1**) that trust has to be context-dependent: we can trust The Lancet on medical issues, but we should trust the Daily Mail on celebrities lifestyle.

Moreover, the analyst can see that [10] disputes [12]. Such a conflict might be automatically resolved if one of the sources is more trustworthy than the other: this requires (second desideratum, **Des2**) that trust needs to be expressed through an ordinal scale, i.e. it should be possible to determine whether, given the same context, a source of information is more trustworthy than another.

3 An Ontology of Trust for Situational Understanding

3.1 What is an Ontology?

An ontology comprises two components [1]: the vocabulary (TBox) and the assertions about individuals (ABox). The vocabulary consists of concepts, which denote sets of individuals; and roles, which denote binary relationships between individuals.

Elementary descriptions are atomic concepts and atomic roles. Complex descriptions can be built from them inductively with concept constructors. In abstract notation, we use the letters A and B for atomic concepts, the letter R for atomic roles, and the letters C and D for concept descriptions. Concept descriptions in $\mathcal{ALI}+$ are formed according to the following syntax rule: $\mathsf{C}, \mathsf{D} \rightarrow \mathsf{A}$ (atomic concept) $\mid \top$ (universal concept) $\mid \bot$ (bottom concept) $\mid \mathsf{C} \sqcap \mathsf{D}$ (intersection) $\mid \forall R.\mathsf{C}$ (value restriction, or range) $\mid \exists R.\top$ (limited existential quantification, or domain). R^{-} denotes the inverse role; roles can be transitive and symmetric. The syntax of $\mathcal{ALI}+$ can clearly be presented using the OWL 2 Web Ontology Language.[1] Due to space constraints we omit the formal description of semantics, that are given—as usual—by means of an interpretation. For ease of representation, in the following we will use a graph-based syntax, where nodes are either atomic concepts or individuals (identified by ♦), and edges are labelled with one of the following labels: *is-a* (representing sub-concept relation), *type* (membership assertion), or R (roles).

Moreover, as presented in [8], ORL is a language for expressing Horn clause rules that extends the OWL language in a syntactically and semantically

[1] https://www.w3.org/TR/owl2-overview/.

coherent manner. A rule has the form *antecedent* → *consequent*, where both *antecedent* and *consequent* are conjunctions of atoms written $a_1, \ \ldots, \ a_n$. Variables are prefixed with a question mark—e.g. ?x. The model-theoretic semantics for ORL is an extension of the semantics given by an interpretation. A rule is satisfied by an interpretation iff every binding—mapping to elements of the domain—that satisfies the antecedent also satisfies the consequent. An interpretation satisfies an ontology iff it satisfies every axiom (including rules) and fact in the ontology [8].

3.2 Our Proposal: SitUTrustOnto

Figure 1, together with Table 1, depicts SitUTrustOnto, our proposed ontology of trust for situational understanding.[2]

The concept Source contains sources of information, e.g. blog posts, twits, scientific papers.... The concept Query describes the situation that needs to be understood, e.g. specific queries an analyst has to address, such as "Do vaccinations cause autism?"

The concept TrustDescriptor contains trust descriptors: given our interest in supporting human analysts and not to replace them, we chose to rely on the *admiralty rating* [9] that identifies the following five descriptors listed in

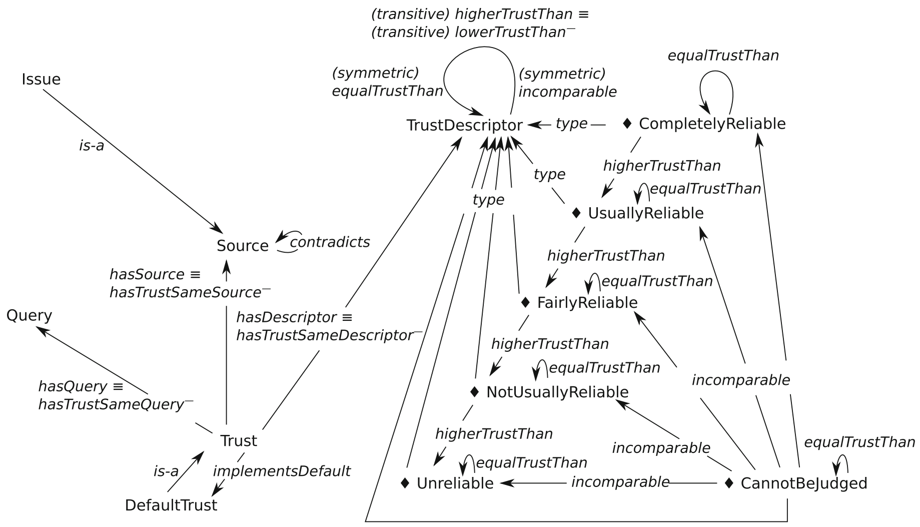


Fig. 1. Graphical representation of SitUTrustOnto: nodes with ♦ are individuals, otherwise atomic concept. Edges are labelled with *is-a* (subclass relations), with *type* (membership assertion); and with atomic roles. For instance, *(transitive) higherTrust* $\equiv$ *(transitive) lowerTrustThan*$^{-}$ describes the role *higherTrust* with domain and range Trust, such as it is the inverse of *lowerTrust*.

[2] Available at https://goo.gl/DLq6dz.

Table 1. ORL rules in SitUTrustOnto.

R1:	Trust(?x), DefaultTrust(?y), implementsDefault(?x, ?y), TrustDescriptor(?t), hasDescriptor(?y, ?t)	→ hasDescriptor(?x, ?t)
R2:	Source(?x), Source(?y), contradicts(?x, ?y), Trust(?ax), Trust(?ay), hasTrustSameSource(?x, ?ax), hasTrustSameSource(?y, ?ay), Query(?q), hasQuery(?ax, ?q), hasQuery(?ay, ?q), hasDescriptor(?ay, ?ty), hasDescriptor(?ax, ?tx), equalTrustThan(?tx, ?ty)	→ Issue(?x)
R3:	Source(?x), Source(?y), contradicts(?x, ?y), Trust(?ax), Trust(?ay), hasTrustSameSource(?x, ?ax), hasTrustSameSource(?y, ?ay), Query(?q), hasQuery(?ax, ?q), hasQuery(?ay, ?q), hasDescriptor(?ay, ?ty), hasDescriptor(?ax, ?tx), lowerTrustThan(?tx, ?ty)	→ Issue(?x)
R4:	{Issue(?x), Source(?y), contradicts(?x, ?y)}	→ Issue(?y)

decreasing order: *Completely Reliable*, *Usually Reliable*, *Fairly Reliable*, *Not Usually Reliable*, *Unreliable*; plus a sixth, incomparable, one, namely *Cannot Be Judged*. Figure 1 shows the six individuals belonging to TrustDescriptor, and their relationships expressed by *equalTrustThan*—identifying that two descriptors are equivalent; *higherTrustThan* and *lowerTrustThan*—expressing ordering, and thus satisfying (**Des2**); and *incomparable*—when two descriptors cannot be compared.

The concept Trust describes the relationship between a source of information, a query, and a trust descriptor, thus providing the context in which to assess the trust in a source of information for a given query. Please note that there is no role linking directly Source to TrustDescriptor, nor Source to Query, thus satisfying (**Des1**). Moreover, DefaultTrust is a sub-concept of Trust that provides *default* trust accounts between some types of queries and some sources of information. For instance, we might want to express that scientific papers addressing medical queries generally have high trust, and thus we can create a specific individual belonging to DefaultTrust. This means that when a new piece of information is added to the ontology, we can exploit defaults for assessing its trustworthiness using the rule R1 from Table 1 (see Sect. 3.3 for a complete example).

Finally, since different Sources can contradict each other, Issue is populated by the means of rules R2, R3, and R4 of Table 1. This is the case where two sources of information ◆a and ◆b, related to the same query, contradict each other, i.e. ◆a *contradicts* ◆b, and either they have the same level of trust (*equalTrustThan*), cf. R2 in Table 1; or the trust in ◆a is *lowerTrustThan* ◆b, cf. R3 in Table 1. This is based on the assumption that if ◆a *contradicts* ◆b, but ◆a is more trustworthy than ◆b, an analyst would accept ◆a and discard ◆b. We agree that this is not necessary the case, and further comments are outlined in Sect. 4. Finally, if a source of information also belong to Issue, also the sources it is in conflict with belong to Issue (cf. R4 in Table 1).

3.3 SitUTrustOnto and Our Case Study

Figure 2 depicts (in blue) SitUTrustOnto enriched with elements of the scenario discussed in Sect. 2. The query ♦ Do vaccinations cause autism? is a member of Vaccination, a sub-concept of Medicine, sub-concept of Query. The Wakefield et al. [12] paper ♦ https://goo.gl/83pRSA and the retraction notice [10] ♦ https://goo.gl/XpFQgK both are ScientificPapers serving as Source, and they are linked to the query through two individuals belonging to Trust, ♦ trust83pRSA and ♦ trustXpFQgK respectively.

We also describe the assumption that when dealing with a medical topic, a scientific paper is ♦ CompletelyReliable with ♦ defalutMedicineScientificPaper belonging to DefaultTrust, and linked to the trust assessments ♦ trust83pRSA and ♦ trustXpFQgK through the role *implementsDefault*.

As a result of automated reasoning (in red in Fig. 2), rule R1 is fired twice leading to assess both ♦ https://goo.gl/83pRSA and ♦ https://goo.gl/XpFQgK as ♦ CompletelyReliable. Because of that, and because ♦ https://goo.gl/XpFQgK *contradicts* ♦ https://goo.gl/83pRSA, they both belong to Issue (rules R2 and R4) thus flagging to the analyst the need for further investigation on these two sources of information.

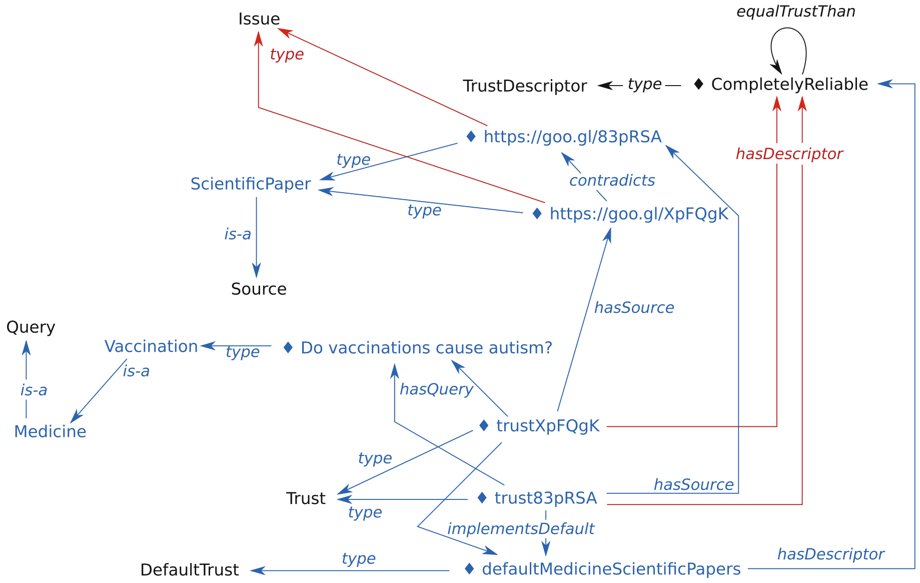


Fig. 2. SitUTrustOnto extended to include elements of the use case (in blue), and (some of) the inferred relationships (in red).

4 Conclusion

In this paper we proposed a computational methodology for evaluating trust associated to sources of information in situational understanding. In particular, we showed how our computational methodology supports situational understanding by drawing conclusions from defaults, as well as highlighting issues due to conflicts between sources of information that demand further investigation to be solved. For instance, in our case-study, the assumption that scientific papers share the highest level of trust when considered in the context of a scientific enquiry proved to be debatable.

This is a first investigation towards a support system for trust management in situational understanding. As part of future work we plan to evaluate techniques for automatic evaluation of trust: we will investigate how topic modelling—i.e. automatically identifying relevant topics in a written document, for instance using the Latent Dirichlet Allocation (LDA) [3]—and similarities of sources—e.g. articles in The Lancet are more similar to articles in the New England Journal of Medicine rather than to articles in the Daily Mail—can help suggesting trust measures for new pieces of information. Moreover, we will also investigate how to automatically identify problems with default assumptions, i.e. questioning whether there is enough evidence suggesting that a default assignment needs to be revisited. This will be part of a large empirical evaluation.

The notion of conflict in this preliminary paper is quite rudimentary: as part of future investigations we will exploit argumentation systems [2] and the Argument Interchange Format ontology [4] for their ability to reason about different types of conflicts. Moreover, following [11] where an argumentation system for supporting intelligence analysis is proposed, we will also investigate the relationship between trust and provenance of information [7].

Finally, as answers to situational understanding queries often require to fuse pieces of information into a single, coherent document, we will assess how to evaluate the trust of fused documents by building on qualitative decision under uncertainty [6].

References

1. Baader, F., Nutt, W.: Basic description logics. In: The Description Logic Handbook, pp. 43–95. Cambridge University Press (2003)
2. Besnard, P., Hunter, A.: Constructing argument graphs with deductive arguments: a tutorial. Argum. Comput. **5**(1), 5–30 (2014)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
4. Chesnevar, C.I., McGinnis, J., Modgil, S., Rahwan, I., Reed, C., Simari, G.R., South, M., Vreeswijk, G.A.W., Willmot, S.: Towards an argument interchange format. Knowl. Eng. Rev. **21**(04), 293 (2006)
5. Dubois, D., Fargier, H., Prade, H., Perny, P.: Qualitative decision theory: from savage's axioms to nonmonotonic reasoning. J. ACM **49**(4), 455–495 (2002)
6. Dubois, D., Prade, H., Rico, A.: Residuated variants of Sugeno integrals: towards new weighting schemes for qualitative aggregation methods. Inf. Sci. **329**, 765–781 (2016)
7. Hartig, O., Zhao, J.: Using web data provenance for quality assessment. In: First International Conference on Semantic Web in Provenance Management, pp. 29–34. CEUR-WS.org (2009)

8. Horrocks, I., Patel-Schneider, P.F.: A proposal for an OWL rules language. In: Proceedings of WWW 2004, pp. 723–731 (2004)
9. Prunckun, H.: Handbook of Scientific Methods of Inquiry for Intelligence Analysis. The Scarecrow Press, Lanham (2010)
10. The Editors of The Lancet: Retraction-Ileal-lymphoid-nodular hyperplasia, non-specific colitis, and pervasive developmental disorder in children. Lancet **375**(9713), 445 (2010). https://www.ncbi.nlm.nih.gov/pubmed/20137807
11. Toniolo, A., Norman, T.J., Etuk, A., Cerutti, F., Ouyang, R.W., Srivastava, M., Oren, N., Dropps, T., Allen, J.A., Sullivan, P.: Agent support to reasoning with different types of evidence in intelligence analysis. In: Proceedings of AAMAS 2015, pp. 781–789 (2015)
12. Wakefield, A., Murch, S., Anthony, A., Linnell, J., Casson, D., Malik, M., Berelowitz, M., Dhillon, A., Thomson, M., Harvey, P., Valentine, A., Davies, S., Walker-Smith, J.: Ileal-lymphoid-nodular hyperplasia, non-specific colitis, and pervasive developmental disorder in children. Lancet **351**(9103), 637–641 (1998)

Intelligent Transportation

Traffic Condition Analysis Based on Users Emotion Tendency of Microblog

Shuru Wang[1,2,3], Donglin Cao[1,2,3](✉), Dazhen Lin[1,2,3], and Fei Chao[1,2]

[1] Cognitive Science Department, Xiamen University, Xiamen, China
wsr9210@foxmail.com, {another,dzlin,fchao}@xmu.edu.cn
[2] Fujian Key Laboratory of Brain-Inspired Computing Technique and Applications, Xiamen University, Xiamen, China
[3] Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, Minjiang University, Fuzhou, China

Abstract. Analysis of traffic condition is of great significance to urban planning and public administration. However, traditional traffic condition analysis approaches mainly rely on sensors, which are high-cost and limit their coverage. To solve these problems, we propose a semi-supervised learning method which uses the social network data instead and analyzes the traffic condition based on user's emotion tendency. First we train the Gated Recurrent Unit (GRU) model to estimate the sentiment of microblog with traffic information, then using the emotional tendency to predict whether traffic jams happen or not. In order to reduce the data annotated by manpower, we propose a new idea to employ the Conditional Generative Adversarial Networks (CGAN) to generate samples which are as a supplement to the training set of GRU. Finally compared with the GRU model trained by solely the manual annotation data, our method improves the classification accuracy by 4.07%. We also use our model to predict the time and roads of traffic jams in 4 Chinese cities which is proved to be effective.

Keywords: Traffic condition · Sentiment analysis · Microblog · Sample generation · Generative Adversarial Networks

1 Introduction

With the progress of urbanization in China, population and traffic flow grow rapidly and traffic jam becomes a common problem in urban. Terrible traffic condition makes urban run inefficiently as well as lead to pollution and public security problem. How to acquire the real-time traffic information easily and analyze the traffic condition in time have turned into urgent need for city development.

Traditional traffic information is collected by devices which has some shortcomings:

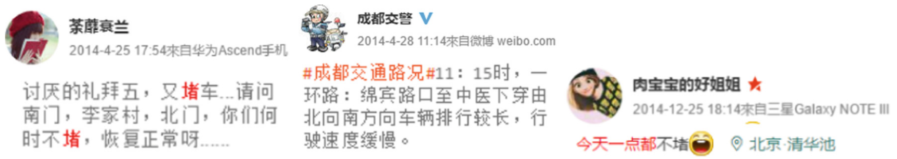
(1) **High-cost:** the devices like loop sensors are expensive and building and maintaining the device will spend a lot of money.
(2) **Limited coverage:** the sensors would like to be located in main blocks rather than alleys, the damage of devices also limits their coverage.

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_26

(3) **Limited information:** it's possible to predict a traffic jam but hard to know why the traffic is blocked by sensor-based method.

To solve these problems, we would like to use social network data which offers a cheap and easy way to gain rich and real-time traffic data, the semantic information in data is good to get the details of traffic jams, like location reasons and so on.

Figure 1 shows three microblogs with traffic information including intense personal emotion which can help us to estimate whether the traffic condition is good or not. We propose a way to analyze the traffic condition base on the emotion tendency of microblog. A tough problem of sentiment analysis is the large demand of labeled data, the common way is use the emoticon to tag the data automatically. In this paper, we propose a fresh method to use the generative model to increase the training set.



荼蘼衰兰
2014-4-25 17:54来自华为Ascend手机
讨厌的礼拜五，又堵车...请问南门，李家村，北门，你们何时不堵，恢复正常呀......

(a) negative

成都交警 V
2014-4-28 11:14来自微博 weibo.com
#成都交通路况#11：15时，一环路：绵宾路口至中医下穿由北向南方向车辆排行较长，行驶速度缓慢。

(b) neutral

肉宝宝的好姐姐 ★
2014-12-25 18:14来自三星Galaxy NOTE III
今天一点都不堵 北京·清华池

(c) positive

Fig. 1. The microblogs involving traffic clues. The content of 1(a): Nasty Friday, I met the traffic jam again…it's really confused me that when do the South Gate, Li village and North Gate can stop jamming and get normal. The content of 1(b): #the traffic condition in Chengdu# At 11:15, Yihuan road: south heading vehicle queue between Mianbin road and the hospital of traditional Chinese medicine is long and proceeds slowly. The content of 1(c): There is no traffic jam today [laugh].

Our contributions in this paper are:

(1) We present a lowcost and wide-coverage method to estimate the traffic condition by sentiment analysis of social network.
(2) We build a model based on GRU to analyze the tendency of the microblog with traffic clues in Sina Weibo, the positive tendency shows the traffic is in good condition, while negative ones reveal that traffic jams may have happen. We employ the model to predict the roads and time that may meet the traffic jams in 4 Chinese cities which demonstrates that our method is effective.
(3) To alleviate the problem of lacking annotation data and improve the quality of training set, we propose a new idea that to train GAN using the data labeled by emoticon to generate data which can increase the size of classifier's training set and improve the generalization of the model.

This paper is organized as follows: we first introduce the study on traffic condition analysis based on social network and related work about sentiment analysis in Sect. 2, then we show how to build the model in Sect. 3. Section 4 discusses the effect of our model and analyze the traffic condition of 4 Chinese cities. Finally conclusion and future work are described in Sect. 5.

2 Related Work

2.1 Traffic Condition Analysis Based on Social Network

Traditional traffic information is mainly collected by questionnaire method, detectors and large-scale probe vehicles, then evaluate the traffic status by image processing or modeling methods [1–4], which can reflect the traffic condition in time, but it is an expensive way to build and maintain the devices, what's more, their popularities are highly affected by regional geography and economy.

With the popularity of social network, more studies come to focus on capturing the traffic information from social network. Wang *et al.* [5] gave an efficient way to collect the microblogs with traffic information and considered the events related to traffic condition to make a traffic congestion estimation. Combining microblogs with geographic information, Chen *et al.* [6] summarized the spatio-temporal statistics of Xi'an traffic jams.

At present the research on learning traffic condition from sentiment analysis is a few and the way to analyze the sentiment is relatively simple, Cao *et al.* [7] proposed a rule-based approach which estimates the emotion of microblog by emotional dictionary to analyze 2 traffic events. Shekhar *et al.* [8] detected traffic jams based on the sentiment of Facebook and Twitter of 2 Indian cities by dictionary based method and analyzed the reasons which may cause the congestion.

2.2 Sentiment Analysis on Social Network

There are mainly two approaches on text sentiment analysis, one is dictionary based method, and the other is based on the statistical learning.

Dictionary Based Method. The key idea of dictionary based method is that considering the emotion of key words can get the sentiment of text. To save the emotion value of words, experts have built some dictionaries, such as the Wordnet [9], General Inquirer (GI) [10], Hownet [11] and so on. Turney *et al.* [12] proposed an unsupervised learning method to build a 2-emotion dictionary by calculating the mutual information between seed emotional words and other words.

Dictionary based method offers an efficient and simple way to analyze the text sentiment, but it demands lots of experts to build and update the dictionaries, what's more, it fails to consider the semantic links between words.

Statistical Learning Based Method. The process of the method is first to extract the text features, then to train the model to learn the emotion from the features.

Pang *et al.* [13] trained Support Vector Machine (SVM), naive Bayes and maximum entropy models on 8 linguistics features respectively, and found that SVM performed best which achieved 82.9% accuracy on determining whether a movie review is positive or negative. Go *et al.* [14] applied maximum entropy model trained on tweet and gained 83% accuracy on binary sentiment classification problem. Considering 3 features, Liu *et al.* [15] compared 3 machine learning models on Chinese microblog sentiment analysis. Recently, deep learning models have been employed in text sentiment analysis. Kalchbrenner *et al.* [16] proposed a hierarchical Convolutional Neural

Network (CNN) to as sentence-level model and extended the model into the chapter-level model by Long Short Term Memory (LSTM).

The disadvantage of statistical learning based method is that it calls for large labeled data, which urges researchers to find a cheap way to label the data. Hu *et al.* [17] proposed an unsupervised learning method to learn text sentiment based on the data tagged by emoticon. Another solution is to adopt a generative model to generate the data which satisfies the distribution of microblogs to supplement the training set. Generative Adversarial Networks (GAN) [18] is the most popular model for it can generate sample faster and better without Markov chains and inference. As we know few studies have done on sentiment analysis of microblogs based on sample generation, this paper is an attempt on this method.

In brief, the traditional traffic condition analysis approaches based on sensors are expensive and limited on coverage, thus more and more researchers turn to use the social network data. At present the research on learning traffic condition based on emotion tendency of microblog is a few and the way to analyze the sentiment is relatively simple, mainly using the emotion dictionary. A better way on sentiment analysis is statistical learning based method, but it calls for large labeled data for training model, the common solution is to use the emoticon as weak label. In this paper, we analyze the traffic condition based on user's emotion tendency by a deep learning model and to solve the problem of data annotation, we propose a sample generation method to increase the training set.

3 Traffic Condition Analysis Based on Users' Emotion

Our approach can be divided into two parts as shown in Fig. 2, first is the sample generation, we use emoticons as weak labels to tag the microblogs to train CGAN so that CGAN can generate the data which is not only consistent with the microblog

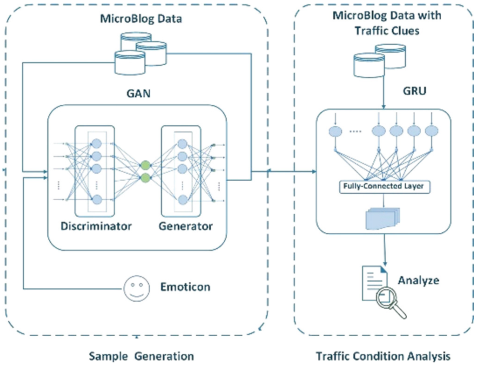


Fig. 2. The process of our approach can be divided into two parts: (1) Sample generation: the CGAN is trained on a weak-label learning method whose data is labeled by emoticons. (2) Traffic condition analysis: we train a GRU model to capture the sentiment tendency, the training dataset is the mixture of microblog and data generated by CGAN. Determining the traffic condition depends on the sentiment tendency analysis for microblog.

distribution but also with specific emotional tendency, the following is to train a GRU model to capture the sentiment tendency based on microblogs and generative data, then determine the traffic condition based on the sentiment.

3.1 Sample Generation

Conditional Generative Adversarial Networks. GAN is composed of a generative model G and a discriminative model D. We define the microblog distribution as $P_{data}(x)$, for any z which belongs to random distribution $P_{noise}(z), G$ maps the z into $G(z)$ to approximate to the microblog distribution $P_{data}(x)$. The task of D is to judge whether the input satisfies the $P_{data}(x)$. $D(x)$ denotes the probability that x comes from the microblog distribution rather than generative distribution:

$$x_g = G(z) \tag{1}$$

$$D(x_g) = p(x_g \in P_{data}(x)) \tag{2}$$

The D and G are trained simultaneously by playing a MiniMax game, the GAN gets optimal when both models reach the Nash Equilibrium, the value function is:

$$minmax\, V(D,G) = E_{x \sim p_{data}(x)}[log(D(x))] + E_{z \sim p_{noise}(z)}[log(1 - D(G(z)))] \tag{3}$$

The GAN trained in unsupervised way is able to capture the distribution of microblog, but may fail to learn its sentiment tendency. In order to generate the data with emotion for training the classifier, we build a conditional GAN by adding the sentiment information t into D and G, as shown in Fig. 3:

$$x_g = G(z|t) \tag{4}$$

$$D(x_g) = p(x_g \in P_{data}(x|t)) \tag{5}$$

$$minmaxV(D,G) = E_{x \sim p_{data}(x)}[log(D(x|t))] + E_{z \sim p_{noise}(z)}[log(1 - D(G(z|t)))] \tag{6}$$

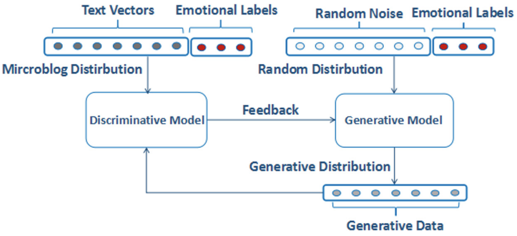


Fig. 3. The structure of conditional GAN. Compared with the original GAN, we add the emotional labels to guide the generative model to generate the data with certain emotion. The input of G are the random noise and its emotional label while the output is the generated data whose sentiment is consistent with input's label. The microblogs transformed into the vectors with their labels and the generative data are both the input of D, discrimination results made by D optimize the whole model.

Emoticon Based Weak Labeling. There must be sufficient training data so that the conditional GAN can learn the distribution of microblog. Here we transform the emoticons into sentiment labels. First we collect 314 emoticons from 5,700 labeled microblogs, then count their frequency on 3 kinds of emotion respectively, finally make an emotional score table of emoticon as shown in Table 1.

Table 1. The emotional score table of emoticon

Emoticon	Positive	Negative	Neutral
爰你 (love you)	0.822236	0.022613	0.155151
泪 (tear)	0.176554	0.740819	0.082627
偷笑 (titter)	0.649351	0.064935	0.285714

For any microblog, the emoticons it including are defined as $E_1, E_2, \ldots, E_n$. $S(E_i[pos])$ in Eq. (7) means the positive score of the E_i. We sum up the all scores of E_i on 3 types of emotion and calculate the averages respectively, take the emotion whose average is maximum as the final tendency of the microblog:

$$Label([E_1, E_{2,\ldots,} E_n]) = argmax\left\{\frac{1}{n}\sum\nolimits_{i=1}^{n} [S(E_i[pos]), S(E_i[neg]), S(E_i[neu])]\right\} \tag{7}$$

3.2 Traffic Condition Analysis

User Sentiment Analysis. We employ the GRU model as a classifier which is the improved by LSTM. As shown in Fig. 4, the training data is the mixture of annotation data and generative data. The test data are the microblogs involving traffic clues, the sentiment analyzed by classifier can help us to determine if the traffic condition is good or not, we give an evaluation index as follows.

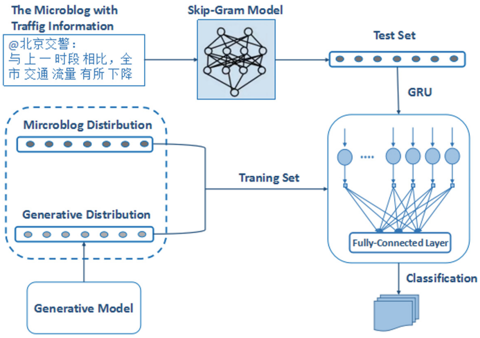


Fig. 4. The process of training the classifier. The training data consists of ordinary microblogs and generative sample while the test data is the microblogs with traffic information. According to the sentiment analyzed by classifier, we can determine the traffic condition is good or not by the value of emotion index.

Traffic Emotion Index. The emotion index (EI) is the intensity and tendency of emotion and can be used to describe the traffic condition. Observing from data, we find that the microblog with neutral emotion is mainly posted by official accounts, the describe of which is usually objective and neutral, but the content of which is with positive or negative information in some ways, thus we assume that half of neutral microblog is positive and the other is negative. For the traffic event T_j (like the traffic condition in Beijing), the emotion index is the proportion of the number of negative microblogs plus half the number of neutral ones to all microblogs related to T_j. As shown in Eq. (8):

$$EI\left(T_j\right) = \frac{2\sum_{i=1}^{N_j} I(y_i = neg) + \sum_{i=1}^{N_j} I(y_i = neu)}{2N_j} \tag{8}$$

EI = 0.5 suggests no sign of emotion and thus indicates the traffic is normal. EI > 0.5 means T_j is negative and reveals that there may be some traffic jams while EI < 0.5 implies a positive emotion and therefore a good traffic condition. The absolute difference between EI and 0.5 reflects a degree for the prediction.

4 Experiments

4.1 Dataset

All of our data is crawled from Sina Weibo. We collected 17,512,946 microblogs with emoticons. 16,520,489 microblogs were weakly labeled by the method mentioned in Sect. 3.2, including 13,344,594 data with positive tendency, 2,756,486 data is negative and 419,409 data is neutral. The reasons for this distribution are that:

(1) Microblog conveys users' emotion and since people tend to express themselves under a non-neutral emotion, most microblogs are labeled as positive or negative.

(2) Most emoticons are positive so that a large portion of the data is labeled as positive. There are 199 positive emoticons, 75 for negative and 45 for neutral in the score table of emoticon.

For the data to train CGAN, we selected 600,000 microblogs randomly while the dataset of classifier consists of 4,200 labeled training and 1,500 labeled test microblogs, more details are shown in Table 2.

Table 2. The data set of GRU model and CGAN

	GRU model		CGAN
	Training set	Test set	Training set
Positive	1,400	500	200,000
Negative	1,400	500	200,000
Neutral	1,400	500	200,000

In terms of data for analyzing the traffic, we took “traffic flow”, “accident”, “traffic” as key words and crawled the microblogs from 2011 to 2014. After removing the duplicate microblogs and noise like advertisements, we gathered 6,538 data about “traffic flow”, 24,923 about “accident” and 19,438 about “traffic”.

4.2 The Effect of Users’ Emotion Model

The GRU model and CGAN are built on the Keras which runs on the top of Theano. Before feeding in both models, microblogs are transformed into 128d vectors by a Skip-Gram model. It is a distributed representation of text named word2vec, which considers the sequential information of text and turns word into a low-dimensional, continuous and dense vector.

There are 2 CNNs making up the CGAN. The G model has 2 deconvolution layers while the D model has 3 convolution layers, more details are described in [19]. Finally we adopted the data generated by the model which has been trained for 21 iterations and 70 iterations.

The GRU model consists of a GRU layer and a fully connected layer, the input of which is a 3d tensor. The first one is the size of training set, the second one is set to 64, which is the word number of every microblog and the last is the vector size of word. The output is classification result. The loss function is cross-entropy, the optimizer is Root Mean Square Propagation (RMSProp). The model was trained 120 times.

We tested the performance of GRU trained by different mixture of generative data and the annotated data. The baseline is the performance on the model train by 5,700 annotated data (4,200 training samples and 1,500 testing samples). Our results are summarized in Table 3 and evaluated by classification accuracy. The classifier trained by microblogs and generative data with 70 training iterations preforms best with accuracy of 62.0%, compared with the baseline, the accuracy rate is increased by 4.07%. Furthermore, results show that both GRU and SVM are improved by combining the annotated data and generative data, which means that we can annotate less data to obtain a better model by adding the generative data.

Table 3. The classification accuracy of GRU model and SVM (5700_ori denotes 5,700 annotated data, gen21denotes generative data with 21 training iterations, gen70 denotes generative data with 70 training iterations, 5700_gen21 denotes 5,700 generative data with 21 training iterations, 5700_gen70 denotes 5,700 generative data with 70 training iterations)

Data set	GRU	SVM
5700_ori (baseline)	57.93%	61.13%
5700_gen21	41.07%	33.13%
5700_gen70	36.40%	33.33%
5700_ori+gen21	61.40%	61.67%
5700_ori+gen70	**62.00%**	**61.87%**

The details about how the changes go with the size of generative data are shown in Fig. 5. In this experiment, we only use 5,700 annotated data and add different size of generative data. We can see that the performance of GRU and SVM have some improvement at first, and then become worse and worse. This is because the CGAN is good at generate the continuous data, like pictures, but not good at generate the semantic sequence data like text. Thus, the generative data is not as good as annotated data. It is clearly shown in Table 3 that the experiment with 5,700 generative data only obtain 41.07% and 33.13% accuracy in GRU and SVM respectively. Therefore, a large number of generative data may increase the noise for training and lead to the decrease of accuracy.

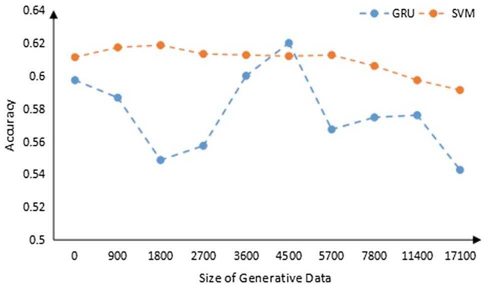
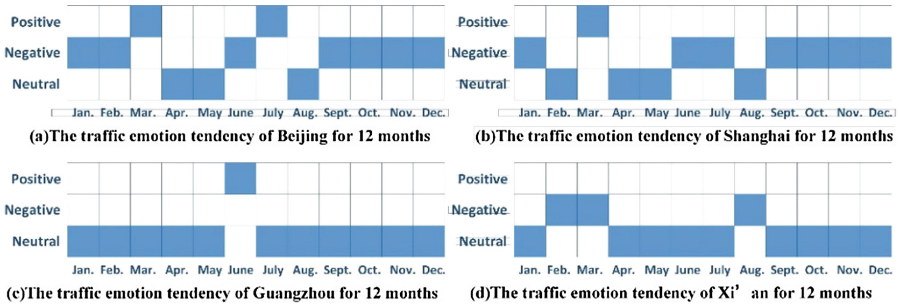


Fig. 5. The accuracy under different size of generative data.

4.3 Traffic Condition Analysis on Chinese Cities

We extracted the data about “Beijing”, “Guangzhou”, “Shanghai” and “Xi’an” from the dataset of traffic, then employed our model to estimate the emotion of the data and analyze the traffic condition of these cities by chronological order.

Figure 6 shows the traffic tendency of 4 cities by month. Each column of subgraph expresses the emotion intensity of the month, the blue one means what emotion is leading. Observing from the figure, we can find that the emotion on traffic in Beijing



(a)The traffic emotion tendency of Beijing for 12 months

(b)The traffic emotion tendency of Shanghai for 12 months

(c)The traffic emotion tendency of Guangzhou for 12 months

(d)The traffic emotion tendency of Xi’ an for 12 months

Fig. 6. The traffic emotion tendency of 4 cities for 12 months

and Shanghai almost seems to be negative, which means both cities are often in bad traffic condition, so that people would like to complain and express the feeling of disappointment. The emotion in Guangzhou is mainly neutral with positive tendency as a whole, compared with the other cities, the traffic condition in Guangzhou may be the best.

Most cities present negative emotion in January and February as the Fig. 6, maybe because the Chinese New Year is in 2 months, there will be a great number of people coming back to the hometown in January and leave for work in February, which brings great pressure on traffic.

We also summarized the rules by the emotion index of 4 cities' traffic in 24 h, the result is shown in Fig. 7. Among the 4 cities, the EI of Beijing is almost greater than 0.5 besides at wee hours which indicates that Beijing is under heavy traffic most of the day. Guangzhou's EI fluctuates around 0.5, showing that its traffic pressure is less than others. By analyzing the time when the EI is greater than 0.5, we can conclude that there are four periods, which is 7:00–9:00, 11:00–13:00, 14:00–15:00 and 19:00–21:00, in a day that always show heavy traffic. And these periods make sense since they match the time workers on and off their duties.

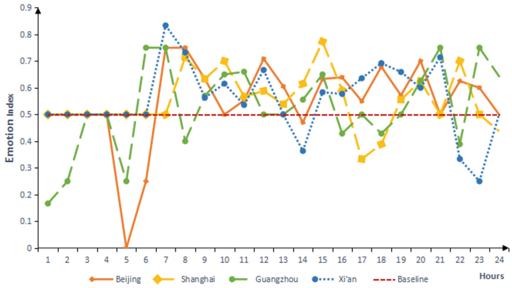
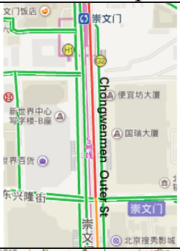
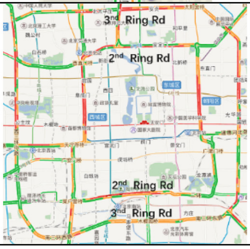
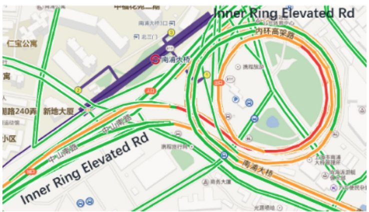
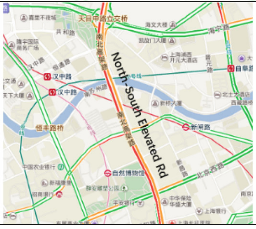


Fig. 7. The emotion index of 4 cities' traffic in 24 h. The baseline is EI = 0.5 which means the emotion is neutral, the value greater than 0.5 means emotion is negative while less than 0.5 means the emotion is positive which reveals that traffic may be in bad condition.

In order to confirm our approach, we collected road information in 4 cities by the hour from the microblog whose emotion is negative and consider these roads in traffic jams. The predict results are compared with the historical road condition data recorded by Baidu Map. Table 4 shows the part of results ("Beijing" and "Shanghai"), the first 3 columns are place and time that we expect to be stuck in traffic while the figures are the historical information from Baidu Map, the red color means the road is in heavy traffic, the green means the road is clean and the orange means the traffic is slow, from the Table 4, it's proved that our approach is a valid way to determine whether the traffic is in bad condition or not.

Table 4. Roads in traffic jams predicted by us and the road condition predicted by Baidu map

City	Roads in Traffic jams Predicted by Our Model	Time	Historical Road Condition from Baidu Map
Beijing	Chongwenmen Outer Street	11:00	 文门饭店 崇文门 Chongwenmen Outer St 便宜坊大厦 新世界中心写字楼-B座 国瑞大厦 世界百货 东兴隆街 崇文门 北京搜秀影城
	3rd Ring Road and 2nd Ring Road	18:00	
Shanghai	Inner Ring Elevated Road	8:00	 Inner Ring Elevated Rd 内环高架路 仁宝公寓 新地大厦 Inner Ring Elevated Rd 南浦大桥
	North-South Elevated Road	17:00	 天目中路立交桥 汉中路 North-South Elevated Rd 新闸路 自然博物馆

5 Conclusions and Future Work

This paper proposes an approach to estimate the traffic condition whether good or not through sentiment analysis on microblog with traffic information. To alleviating problem of lacking annotation data and improve the quality of training data, we trained CGAN to generate data as approximate distribution of microblog, so as to increase the annotation data. With less manual annotation data, we can obtain a better model in sentiment classification. We also analyze the traffic condition on 4 Chinese cities and summarize the rules of traffic condition by time.

There is still difference between generative data and text data. Next we will improve the quality of generative sample like combine the LSTM with GAN so that to make a better analysis on traffic condition and do more study on traffic condition analysis like accident detection.

Acknowledgement. This work is supported by the Nature Science Foundation of China (No. 61402386, No. 61305061, No. 61502105, No. 61572409, No. 81230087 and No. 61571188), Open Fund Project of Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University) (No. MJUKF201743), and Education and scientific research projects of young and middle-aged teachers in Fujian Province under Grand No. JA15075. Fujian Province 2011 Collaborative Innovation Center of TCM Health Management and Collaborative Innovation Center of Chinese Oolong Tea Industry—Collaborative Innovation Center (2011) of Fujian Province.

References

1. Zhang, J., He, Y.-L., Wei, R.: Analysis of traffic participants' waiting tolerance from investigative questionnaires. Transport Stand. (2010)
2. Schneider, W., Arsenal, R.: Mobile phones as a basis for traffic state information. Intell. Transp. Syst. **13**(15), 782–784 (2005)
3. Zhang, C.B., Yang, X.G., Yan, X.P.: Traffic data collection system based on floating cars. Comput. Commun. **24**(5), 31–34 (2006)
4. Guo, D.H., Cui, W.H.: Trajectory mining for live traffic condition retrieving. J. Wuhan Univ. Technol. Transp. Sci. Eng. **34**(1), 6–9 (2010)
5. Wang, S., He, L., Stenneth, L., et al.: Citywide traffic congestion estimation with social media. In: SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 1–10. ACM (2015)
6. Chen, H., Zhang, X., Zhao, Y., et al.: Study on spatio-temporal distribution of Xi'an traffic congestion based on micro-blog. J. Shaanxi Norm. Univ.
7. Cao, J., Zeng, K., Wang, H., et al.: Web-based traffic sentiment analysis: methods and applications. IEEE Trans. Intell. Transp. Syst. **15**(2), 844–853 (2014)
8. Shekhar, H., Setty, S., Mudenagudi, U.: Vehicular traffic analysis from social media data. In: International Conference on Advances in Computing, Communications and Informatics. IEEE (2016)
9. Wang, F., Wang, H., Xu, K., et al.: Characterizing information diffusion in online social networks with linear diffusive model. In: IEEE International Conference on Distributed Computing Systems, pp. 307–316. IEEE (2013)
10. Yang, J., Leskovec, J.: Modeling information diffusion in implicit networks. In: IEEE International Conference on Data Mining, pp. 599–608. IEEE (2011)
11. Ma, H., Zhou, D., Liu, C., et al.: Recommender systems with social regularization. In: Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February, DBLP, pp. 287–296 (2011)
12. Turney, P.D., Littman, M.L.: Measuring praise and criticism: inference of semantic orientation from association. ACM Trans. Inf. Syst. **21**(4), 315–346 (2003)
13. Pang, B., Lee, L., et al.: Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, vol. 10, pp. 79–86 (2002)

14. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Proj. Rep. Stanford **1**, 12 (2009)
15. Liu, Z., Liu, L.: Empirical study of sentiment classification for Chinese microblog based on machine learning. Comput. Eng. Appl. **48**(1), 1–4 (2012)
16. Kalchbrenner, N., Blunsom, P.: Recurrent convolutional neural networks for discourse compositionality. Comput. Sci. (2013)
17. Hu, X., Tang, J., Gao, H., et al.: Unsupervised sentiment analysis with emotional signals. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 607–618. ACM (2013)
18. Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
19. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)

Fuzzy Bi-objective Chance-Constrained Programming Model for Timetable Optimization of a Bus Route

Hejia Du[1], Hongguang Ma[2], and Xiang Li[1](✉)

[1] School of Economics and Management,
Beijing University of Chemical Technology, Beijing 100029, China
lixiang@mail.buct.edu.cn

[2] School of Information Science and Technology,
Beijing University of Chemical Technology, Beijing 100029, China

Abstract. Timetable optimization is essential to the improvement of a bus operating company's economic profits, quality of service and competitiveness in the market. The most previous researches studied the bus timetabling with assuming the passenger demand is certain but it varies in practice. In this study, we consider a timetable optimization problem of a single bus line under fuzzy environment. Assuming the passenger quantity in per time segment is a fuzzy value, a fuzzy bi-objective programming model that maximizes the total passenger volume and minimizes the total bus travel time under a capacity rate constraint is established. This chance constrained programming model is formulated with the passenger volume and capacity rate under certain chance constraints. Furthermore, a genetic algorithm of variable length is designed to solve the proposed model. Finally, we present a case study that utilizing real data obtained from a major Beijing bus operating company to illustrate the proposed model and algorithm.

Keywords: Timetable optimization · Bi-objective programming · Fuzzy chance-constrained programming · Genetic algorithm

1 Introduction

Bus industry is going to be caught in a dilemma which is impacted by the operation of subway. Thus it is necessary to improve the bus company's competitiveness in the market. Since the goal of each bus operation company is to achieve high efficiency, good quality and cost-effectiveness, public transit planning plays a vital role for a bus company. Guihaire and Hao (2008) summarized the five processes of the public transit planning: routes designing, frequencies determining, timetable setting, vehicle scheduling and crew scheduling. The timetabling is an intermediate step influencing the service quality and the subsequent problems, hence it is a critical step. Besides, timetabling will help increase a bus operating company's competitiveness through improving its operation profits

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_27

and efficiency. Therefore, timetable optimization of a bus line is of great importance to increase the profits, service quality and attractiveness of a bus operating company in the market.

There are lots of researches focusing on bus scheduling from different perspectives. Aiming to minimize passengers' waiting time at the transfer nodes, Ceder et al. (2001) established a model to generate a timetable to maximize the synchronization of a given network of buses. Zhao and Zeng (2008) studied the route network design, vehicle headway and timetable assignment. Yan and Chen (2002) constructed a scheduling model based on a multiple time-space network to maximize profits of a bus operating company. Wu et al. (2016) studied the re-planning of a bus network timetable considering headway-sensitive passenger demand, uneven headway, service regularity, flexible synchronization and the existing bus timetable. A multi-objective re-synchronizing of bus timetable was proposed. In order to improve the service quality of bus, some researchers optimized timetable to increase the reliability. Based on analytical development and micro simulations, Salicrú et al. (2011) presented an approach to generate run-time values to optimize run-time and improve the operating process. In order to make the bus operator provide more reliable transit service, Yan et al. (2012) developed a robust optimization model with aiming to minimize the sum of the expected value of the random schedule deviation and its variability multiplied by a weighting value. Arhin et al. (2016) defined a new variable in total bus stop time to improve the bus transit reliability. From the prospective of passengers, the bus departure timetable should be optimized to reduce the users' cost. Considering that the average waiting time at stops, Amin-Naseri and Baradaran (2014) investigated the accurate formulas for estimating average waiting time at stops, and developed a simulation model to evaluate the performance of proposed formulations. Parbo et al. (2014) presented a timetable optimization model with minimizing the waiting time when transferring. In order to tackle the bus demand fluctuations in transit operation, Sun et al. (2015) built three different models for hybrid vehicle, large vehicle and small vehicle to optimize the timetable. The results indicate that the hybrid vehicle size model excels the other two modes both in the total time and total cost.

The above researches used fixed parameters to represent the passenger demand, but the passenger number varies in practice. The certain passenger volume can not reflect the actual daily passenger demand where uncertain disturbances occur. To solve this problem, some researches assumed that the disturbance is stochastic and established stochastic optimization models. Assuming the waiting time, walking time, in-vehicle time are random variables, Tong and Wong (1998) formulated a dynamic transit assignment model based on a transit network. The scheduled timetables are used to describe the movement of vehicles. From the users' point of view, Yan et al. (2006) established a stochastic-demand scheduling model with minimizing the sum of cost of fleet flows and expected cost in all scenarios to set a timetable. From the perspective of operator, Vissat et al. (2015) proposed a stochastic model of a particular bus route and get a better timetable for the service with less financial risk of penalties, punctuality and reliability.

Based on the above literature review, we find that timetable optimization has drawn great attention and became a hot topic in recent years. However, the application of fuzzy optimization method in timetable optimization is paid less attention, though there are full of uncertainty in bus operation environments. Consequently we consider to optimize the bus timetable in fuzzy environments. In this paper, we assume the passenger volume is a fuzzy value to incorporate the fuzziness. While processing data of passenger volume, a justifiable granularity principle is applied to approximate membership functions to get fuzzy values (Pedrycz and Homenda 2013). With maximizing the total passenger volume and minimizing the total bus travel time under a capacity rate constraint, a fuzzy bi-objective chance-constrained programming model for timetable optimization is established within the framework of credibility theory, a branch of mathematics for studying the behavior of fuzzy phenomena (Liu and Liu 2002).

The remainder of this paper is organized as follows. Section 2 proposes the passenger volume objective, travel time objective and capacity rate constraint and the fuzzy bi-objective chance constrained programming model. In Sect. 3, an genetic algorithm of variable length is designed to solve the proposed model. Section 4 gives a case study to illustrate the efficiency of the proposed model and algorithm. Finally, Sect. 5 is the conclusion.

2 Model Formulation

The proposed model aims at increasing the economic profits and operation efficiency from the perspective of bus company by optimizing the timetable for a fixed bus line. The following terms have been defined and commonly used in our fuzzy bi-objective chance-constrained programming model (see Table 1).

2.1 Passenger Volume Objective

Passenger volume is essential to the inter-city bus transit company since that the carrier is a corporation in pursuit of profits. In the modern public transportation system, every bus line is not isolated but exists in a transport network, and it will be influenced by its common bus lines. Common bus lines are those routes share common sections and the passengers must select which he probably take (Chriqui and Robillard 1975). Many scholars studied a classical passenger assignment model on a transport network (Nguyen et al. 2001; Cominetti and Correa 2001; Cepeda et al. 2006). In this study, we take the factors of common-lines into account and apply this passenger assignment model to forecast passenger volume on a bus line

$$y_i = \sum_{j \in S} y_j \frac{f_i}{\sum_{j \in S} f_j} \tag{1}$$

where s is the set of strategies which can be chosen by passengers, $i \in s$ donates bus line in the transport network, f_i represents a differentiable effective frequency function of line i, the probability of buses' arriving of line i is $f_i / \sum_{j \in S} f_j$.

Table 1. List of notations

Notations	Description
Indices and parameters	
i, j	Bus stop of the line, $i, j = 1, 2, \cdots, I$
k	The k-th bus trip, $k = 1, 2, \cdots, K$
K	The number of bus trip in the working time of a day, $K \in [K^l, K^u]$
l	The l-th time segment, $l = 1, 2, \cdots, L$
h	The h-th time interval, $h = 1, 2, \cdots, H$
n_{li}	The counts of focused buses arrived at stop i in l-th time segment
m_{lij}	The count of common line buses arrived at stop i and heading for stop j in l-th time segment
$p1$	The set of time segments in morning rush hours
$p2$	The set of time segments in evening rush hours
S_{ll}	Left point time of the l-th time segment
S_{lr}	Right point time of the l-th time segment
u, v	The shift of bus
T_s	The first departure time of the bus in a day
T_e	The last departure time of the bus in a day
H_{min}	The minimum schedule interval
H_{max}	The maximum schedule interval
$t_{k,i}$	Arriving time at stop i of the k-th bus trip
C	The capacity of a vehicle (number of seats plus the maximum allowable standees)
α_0	The capacity rate, and $0 < \alpha_0 \leq 1.0$
$T_i^{i+1}(t_{k,i})$	Travel time from stop i to $i+1$ of $t_{k,i}$, $i = 1, 2, \cdots, I-1$
Fuzzy parameters	
$Qlij$	The number of passengers arrived at stop i and destined for stop j in l-th time segment
Decision variables	
$t_{k,1}$	Departure time of the k-th bus, i.e., arriving time at stop 1 of k-th bus trip

For a specific bus line, the average passenger demands have usually served as inputs in the production of the final fleet routes and timetables (Yan et al. 2006). Due to the uncertainties, we denote passenger volumes as fuzzy values. There are I bus stops in a specific bus line, and several bus lines sharing the passenger flow with the researched bus line between in any two stops. Hence the researched bus line is competitive with other common bus lines. We divide the bus operating time of a day into L time segments, and denote that P_{lij} is the passenger volume which form stop i to stop j in time segment l. n_{lij} and m_{lij} are defined as the number of arrival buses of researched bus line and other common bus lines in time segment l. The passenger volume that take the focused bus line form stop i to stop j in time segment l is denoted as

$$Q_{lij} = \frac{n_{lij}}{m_{lij} + n_{lij}} P_{lij}.$$

Since passenger volume is a fuzzy value, the total passenger volume that the researched bus line carry in a day is a fuzzy value, which is $Q = \sum_{l=1}^{L} \sum_{i=1}^{I-1} \sum_{j=i+1}^{I} Q_{lij}$. We maximize the β-optimistic value of total passenger volume based on the optimistic value criteria

$$Q_{\sup}(\beta) = \max \left\{ \bar{Q} \mid \mathrm{Cr} \left\{ \sum_{\mathrm{l=1}}^{\mathrm{L}} \sum_{\mathrm{i=1}}^{\mathrm{I-1}} \sum_{\mathrm{j=i+1}}^{\mathrm{I}} \frac{\mathrm{n_{lij}}}{\mathrm{m_{lij}} + \mathrm{n_{lij}}} \mathrm{P_{lij}} \geq \bar{\mathrm{Q}} \right\} \geq \beta \right\} \tag{2}$$

where β is a predetermined confidence level.

2.2 Travel Time Objective

Transit travel time influence service quality, operating cost and efficiency (Bertini and El-Geneidy 2004). For the sake that the timetable can achieve the effect of staggering peak, we consider to minimize the total bus travel time

$$T = \sum_{k=1}^{K} (t_{k,I} - t_{k,1}). \tag{3}$$

It is obviously for the bus carrier that increasing the departing frequency of buses when there are lots of people need to take buses can increase the passenger number. However in Chinese cities especially in metropolis, traffic congestions always occur during rush hours when many people travel to their jobs. Therefore the buses departed in rush hours are usually caught in traffic, which will reduce the efficiency of bus operation. Thus we seek the balance between passenger volume and bus travel time by minimizing the total bus travel time.

2.3 Capacity Rate Constraint

The bus transit companies pursue the maximum of passenger flow which represents profits. However, more passengers in a bus can reduce the safety and comfort of passengers in this vehicle. Thus the capacity rate can not exceed the standards prescribed by the government in order to guarantee passengers' comfort and security. Consequently, the capacity rate of a vehicle should conform to a determined value. The largest capacity rate is represented usually by the peak-hour cross-section passenger volume, which refers to the largest passenger dispatching volume transited by a rail transit line per unit time within the passenger flow rush hour (Zhu et al., 2015). The largest capacity rate is denoted as

$$\alpha_p = \frac{1}{C} \sum_{l} \sum_{i=1}^{f} \sum_{j=f+1}^{I} Q_{lij}, \ l = p, \ p = p1, p2, \ f = 1, 2, \ldots . I$$

where $p1$ and $p2$ denote the sets of time segments in morning rush hours and evening rush hours respectively. The probability that the largest capacity rate is less than prescribed requirement, is more than γ

$$\mathrm{Cr}\left\{\alpha_{\mathrm{p}} \leq \alpha_0\right\} \geq \gamma,\ \mathrm{p}=\mathrm{p1}, \mathrm{p2} \tag{4}$$

where α_0 is the capacity rate prescribed by the government, α_p is the actual capacity rate, and γ is a predetermined level.

2.4 Model

$$\begin{aligned}
\max\ & Q_{\sup}(\beta) && (5)\\
\min\ & T && (6)\\
\text{s. t.}\ & \mathrm{Cr}\left\{\alpha_{\mathrm{p}} \leq \alpha_0\right\} \geq \gamma,\ \mathrm{p}=\mathrm{p1}, \mathrm{p2} && (7)\\
& n_{li} = \max\{u-v+1 | S_{ll} \leq t_{u,i}, t_{v,i} \leq S_{lr}\},\ \forall l, i && (8)\\
& t_{k,i} + T_i^i(t_{k,i+1}) = t_{k,i+1},\ \forall k, i = 1, 2, \ldots, I-1 && (9)\\
& H_{min} \leq t_{k,1} - t_{k-1,1} \leq H_{max},\ k = 2, 3, \ldots, K && (10)\\
& t_{k-1,i} < t_{k,i},\ \forall i, k = 2, 3, \ldots, K && (11)\\
& t_{1,1} = T_s,\ t_{K,1} = T_e && (12)\\
& T_i^{i+1}(t_{k,i}) = [T_i^h]_{I\times H},\ h = 1, 2, \ldots, H,\ i = 1, 2, \ldots, I-1. && (13)
\end{aligned}$$

There are two objective functions in the mathematical model. The first objective function maximizes the passenger volume that who take the researched bus line. The second objective function (6) minimizes total bus travel time in a day. The constraint (7) guarantees the chance that the largest capacity rates are less than prescribed requirement, are more than γ. Function (8) defines the number of focused buses arrived at stop i in l-th time segment. Constraint (9) guarantees that the arrival time at a stop is the sum of the arrival time at the last stop and the travel time between these two stops. Constraint (10) defines that the departure interval can not exceed a standard prescribed by the government because of the buses' public nature. Constraint (11) is the principle of "first-in first-out" which guarantees that the bus which departure earlier will arrive earlier at every stop. Constraint (12) defines the first and last departure times of the bus in a day which are affected by its public interest. Equation (13) defines the travel time from a stop to the next stop in each time interval.

The first objective function maximizes the passenger volume $Q_{\sup}$ and the second objective function minimizes total bus travel time T in a day. According to the single-objective optimization methods, it is easy to calculate the range for each objective. Here, we use Q_{max} and Q_{min} to denote the maximum and minimum total passenger volume, and use T_{max} and T_{min} to denote the maximum and minimum total travel time. Furthermore, the compromise model is formulated to maximize the linearly weighted objective function

$$\max \quad \lambda \frac{Q_{\sup}(\beta) - Q_{min}}{Q_{max} - Q_{min}} - (1-\lambda) \frac{T - T_{min}}{T_{max} - T_{min}} \tag{14}$$

where λ is a nonnegative real number, which denotes the preference of the decision-maker on the two objectives. If the first objective is more important than the second one, $\lambda > 0.5$ is set, otherwise, $\lambda < 0.5$ is set.

3 Genetic Algorithm of Variable Length

Genetic algorithm (GA) is a computational model for simulating the biological evolution process of natural selection and genetic mechanism. Since genetic algorithm first proposed by Holland (1975), it has been widely studied, experimented and applied by many researchers (Whitley 1994; Jones et al. 1997; Deb et al. 2002). In this section, we design a GA of variable length for solving the proposed fuzzy bi-objective chance-constrained programming model.

Representation Structure: A chromosome $v = (x_1, x_2, \ldots, x_K)$ (see Fig. 1) consists of the departure time of every bus trip. K is the total bus trip times of a day, $x_1 = T_s$ and $x_K = T_e$ are respectively used to denote the departure times of the first and last bus trip in a day.

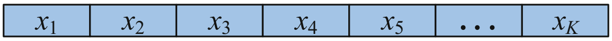


Fig. 1. A chromosome.

Initialization: Firstly, an integer *pop_size* is defined as the size of population and a real number K from $[K^l, K^u]$ is generated randomly. Secondly, randomly generate *pop_size* chromosomes for the initialized population. x_1, i.e., a prescribed time T_s, is used to denote the departure time of first bus trip in a day. The last departure time of bus in a day is x_{K^l}, i.e., a prescribed time T_e. Randomly generate $K - 2$ numbers u_i $(i = [1, K - 2])$ from $[H_{min}, H_{max}]$, and there is a relation $x_{i+1} = x_i + u_i$. Thirdly, if $(x_K - x_{K-1}) \in [H_{min}, H_{max}]$, a chromosome is obtained. Otherwise, repeat the second process until the results satisfy the third requirement and get a feasible chromosome. Repeat the above procedures for *pop_size* times. Denote the generated chromosomes as $\mathbf{v}_i$, $i = 1, 2, \ldots, pop\_size$.

Evaluation Function: Evaluation function assigns each chromosome a probability of reproduction so that its likelihood of being selected is proportional to its fitness relative to the other chromosomes in the population. That is, the chromosomes with higher fitness will have more chance to produce offspring. For each $\alpha \in (0, 1)$, we define the evaluation function as follows

$$Eval(\mathbf{v}_i) = \alpha(1 - \alpha)^{i-1}, \; i = 1, 2, \ldots, pop\_size.$$

Then we rearrange these *pop_size* chromosomes from good to bad based on the order relationship, i.e., evaluation function. Then $\mathbf{v}_1$ is the best chromosome, and $\mathbf{v}_{pop\_size}$ is the worst one.

Selection Process: The method of spinning the roulette wheel is used here to select chromosomes which breed a new generation. The chromosomes with larger fitness are typically more likely to be selected.

Crossover Operation Base Multi-periods: Firstly, we define a parameter P_c to denote the probability of crossover. Generate a random number r_i from $[0, 1]$, and select the chromosome $\mathbf{v}_i$ if $r_i \leq P_c$. Without loss of generality, the crossover operation is introduced on a pair of chromosomes $\mathbf{u}_1 = (x_{11}, x_{12}, \ldots, x_{1K^1})$ and $\mathbf{u}_2 = (x_{21}, x_{22}, \ldots, x_{2K^2})$. Firstly, divide the operating time into N segments $[T_j, T_{j+1}]$ $(j = 1, 2, \ldots, N)$, in which $T_1 = T_s$ and $T_{N+1} = T_e$. Secondly, generate N real numbers m_j from $[T_j, T_{j+1} - t]$ randomly (t represents an hour). Then m_j are used as the time start points of each segment and periods P_j are determined. Thirdly, for these two chromosomes, judge and determine the sections (an hour a section) in the periods P_j. Then exchange the correspond sections and get two new chromosomes. Finally judge the length of each chromosome. If the length of the chromosome $K^3(K^4) \in [K^l, K^u]$, a child is obtained (see Fig. 2). Otherwise, repeat the above process until get two qualified chromosomes. Repeat the above process *pop_size* times.

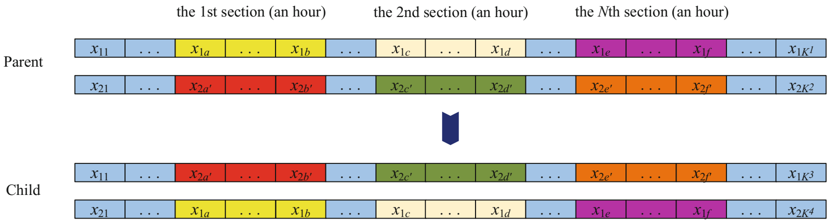

the 1st section (an hour)
the 2nd section (an hour)
the Nth section (an hour)
Parent
x_{11} . . . x_{1a} . . . x_{1b} . . . x_{1c} . . . x_{1d} . . . x_{1e} . . . x_{1f} . . . x_{1K^1}
x_{21} . . . $x_{2a'}$ . . . $x_{2b'}$ . . . $x_{2c'}$ . . . $x_{2d'}$ . . . $x_{2e'}$ . . . $x_{2f'}$ . . . x_{2K^2}
Child
x_{11} . . . $x_{2a'}$ . . . $x_{2b'}$ . . . $x_{2c'}$ . . . $x_{2d'}$ . . . $x_{2e'}$ . . . $x_{2f'}$ . . . x_{1K^3}
x_{21} . . . x_{1a} . . . x_{1b} . . . x_{1c} . . . x_{1d} . . . x_{1e} . . . x_{1f} . . . x_{2K^4}

Fig. 2. Crossover process

Mutation Operation: Firstly we define a parameter P_m to denote the probability of mutation. In mutation operation, the following processes should be repeated *pop_size* times. Randomly generate a real number r_i from $[0, 1]$. If $r_i < P_m$, the ith chromosome is chosen as the parent of mutation. For each parent generate a real number n from $[T_s, T_e - t]$ randomly (t represents an hour) firstly. Then n is used as the time start point and a section (an hour) T is determined. Secondly, judge and determine the part $(x_a, \ldots, x_b)$ that are in the section T. Thirdly, operate mutation process on the selected section. Randomly generate a real number q in $[H_{min}, H_{max}]$, and determine the first point $x_{a'} = x_{a-1} + q$ in the mutating section. The next point's value is equal to the sum of last point and random number q. Then repeat the third procedure until get a value $x_{b'}$ satisfies that $(x_{b+1} - x_{b'}) \in [H_{min}, H_{max}]$. Fourthly, repeat the above processes for N times and get a new chromosome. Finally judge the length of the new chromosome k^2: if $k^2 \in [K^l, K^u]$, a child is obtained (see Fig. 3). Otherwise, repeat the above process until get a qualified chromosome.

General Procedure: The general procedure are summarized in Algorithm 1.

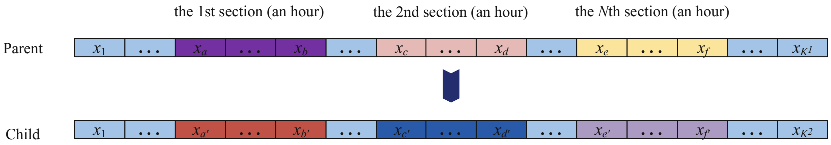

the 1st section (an hour)
the 2nd section (an hour)
the Nth section (an hour)
Parent
x_1 ... x_a ... x_b ... x_c ... x_d ... x_e ... x_f ... x_{K1}
Child
x_1 ... $x_{a'}$ ... $x_{b'}$ ... $x_{c'}$ ... $x_{d'}$ ... $x_{e'}$ ... $x_{f'}$ ... x_{K2}

Fig. 3. Mutation operation

```
Algorithm 1.   Genetic Algorithm
Step 1   Randomly Initialize pop_size chromosomes
Step 2   Calculate the objective values for all chromosomes
Step 3   Evaluate the fitness of each chromosome via the objective values
Step 4   Select the chromosomes by spinning the roulette wheel
Step 5   Update the chromosomes by using crossover and mutation
Step 6   Repeat the second to fifth steps for a given number of cycles
Step 7   Report the best found chromosome as the suboptimal solution
```

4 Case Study

To illustrate how well our model can be applied in reality, we present a case study utilizing statistics which are obtained from a major inter-city bus carrier in Beijing, China. Yuntong 128 bus route runs from its northern terminal Beijing Business School Station in Changping District to its southern terminal Laiguangying North Station in Chaoyang District. It is 21.44 km long with 31 bus stops including the starting and ending terminals.

4.1 Data Collection

Utilizing data in a month (October 4th, 2016 to November 3rd, 2016) from the bus carrier, the operation situations of 31 days (21 working days and 10 nonworking days) were investigated for statistical data. Considering the reality, such as bus repair and maintenance, we give an interval [70, 80] in which the trip times are permitted.

4.2 Data Processing

Pedrycz and Homenda (2013) proposed a justifiable granularity principle which is a formation of a significant representation of a collection of numeric values. The process of granulation of information is defined by Pedrycz and Song (2012), which is a transformation operating on the set of numeric data. Under the framework of granular computing, Zhou et al. (2017) summarized the construction methods of forming the triangular membership function. The detailed procedures for generating triangular membership function are in the following (See Algorithm 2).

Algorithm 2. Design of triangular set (a, b, c) based on justifiable granularity principle

Step 1 Given a collection of numeric data $\Omega = \{x_1, x_2, \ldots, x_N\}$

Step 2 Take b as the mean value of Ω, i.e., $b = \sum_{i=1}^{N} x_i/N$

Step 3 Denote the membership degree of x_i as $\mu(x_i)$, which is computed by

$$\mu(x_i) = \begin{cases} \dfrac{x_i - a}{b - a} & \text{if } x_i < b \\ \dfrac{b - x_i}{c - b} & \text{if } x_i \geq b \end{cases}$$

Step 4 Solve the following maximization problem to determine a

$$\max \sum_{a \leq x_i < b} \mu(x_i) \cdot \exp(-\alpha|b - a|)$$

where $\sum_{a \leq x_i < b} \mu(r_i)$ means the objective of maximizing the number of covered numeric data points, while $\exp(-\alpha|b - a|)$ is the objective of minimizing the support length $|b - a|$, α is a positive parameter

Step 5 Likewise, solve the following maximization problem to determine c

$$\max \sum_{b \leq x_i \leq c} \mu(x_i) \cdot \exp(-\alpha|c - b|)$$

where $\exp(-\alpha|c - b|)$ is the objective of minimizing the support length $|c - b|$.

4.3 Results

Based on the bus operators' preference and experiences, we set the following parameters: $\lambda = 0.7$, $\alpha_0 = 0.7$, $\beta = 0.9$ and $\gamma = 0.8$. Aiming to select the optimal value of P_c and P_m, we solve the model by setting different parameters in the GA. The relative errors of compromise objective values which are defined by (Maximal objective-Actual objective)/Maximal objective × 100%, and the relative errors of running time which are defined by (Actual objective-Minimal objective)/Minimal objective × 100%, are calculated to compare the results obtained from different parameters. The maximal objective value and the minimal running time are the corresponding maximum and minimal of all the computational results we calculated. The detailed results with setting different P_c and P_m are listed in Table 2. Obviously, the P_c and P_m in No. 9 are chosen to be used in the GA.

Table 2. The optimal value of P_c and P_m

No.	P_c	P_m	Compromise objective value	Running time	Relative error (%)	
					Compromise objective value	Running time
1	0.8	0.2	0.4348	4327	15.34	37.85
2	0.8	0.4	0.5079	4375	1.11	39.38
3	0.8	0.6	0.5039	4411	1.89	40.52
4	0.8	0.8	0.3885	3139	24.36	0.00
5	0.7	0.3	0.3902	4368	24.03	39.15
6	0.7	0.5	0.4618	4501	10.09	43.39
7	0.7	0.7	0.4608	3178	10.28	1.24
8	0.7	0.9	0.4719	3149	8.12	0.32
9	0.6	0.4	0.5136	3154	0.00	0.48
10	0.6	0.6	0.3804	3185	25.93	1.47

We run the proposed GA with the parameters set above and obtain the optimal timetable. The convergence of objective value is shown in Fig. 4 which indicates the proposed GA is effective to solve the proposed model. The departure times at each stop of each bus trip is shown in Fig. 5.

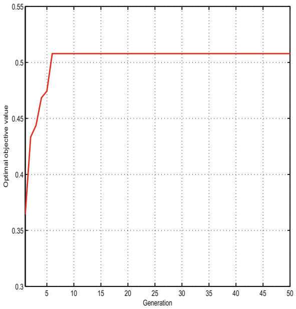


Fig. 4. Convergence

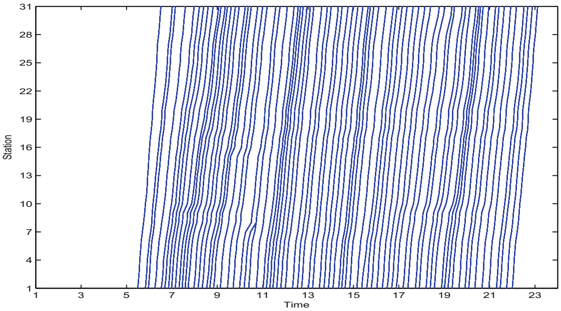


Fig. 5. The obtained timetable

5 Conclusion

In this paper, we proposed a fuzzy bi-objective chance-constrained programming model for a bus timetable optimization problem firstly. Secondly, a genetic algorithm of variable length was designed to solve the proposed model. Finally, a case study was presented to show that the model and algorithm were efficient. In this work, we only considered that passenger volume is fuzzy value. However, due to the weather, traffic conditions and other uncertain factors in practice, more practical aspects should be taken into account in future studies.

References

Guihaire, V., Hao, J.K.: Transit network design and scheduling: a global review. Transp. Res. Part A **42**, 1251–1273 (2008). doi:10.1016/j.tra.2008.03.011

Ceder, A., Golany, B., Tal, O.: Creating bus timetables with maximal synchronization. Transp. Res. Part A **35**, 913–928 (2001). doi:10.1016/s0965-8564(00)00032-x

Zhao, F., Zeng, X.: Optimization of transit route network, vehicle headways and timetable for large-scale transit networks. Eur. J. Oper. Res. **186**, 841–855 (2008). doi:10.1016/j.ejor.2007.02.005

Yan, S.Y., Chen, H.L.: A scheduling model and a solution algorithm for inter-city bus carriers. Transp. Res. Part A **36**, 805–825 (2002). doi:10.1016/s0965-8564(01)00041-6

Wu, Y.H., Yang, H., Tang, J.F., Yu, Y.: Multi-objective re-synchronizing of bus timetable: model, complexity and solution. Transp. Res. Part C **67**, 149–168 (2016)

Salicrú, M., Fleurent, C., Armengol, J.M.: Timetable-based operation in urban transport: run-time optimisation and improvements in the operating process. Transp. Res. Part A **45**, 721–740 (2011). doi:10.1016/j.tra.2011.04.013

Yan, Y.D., Meng, Q., Wang, S.A., Guo, X.C.: Robust optimization model of schedule design for a fixed bus route. Transp. Res. Part C **25**, 113–121 (2012). doi:10.1016/j.trc.2012.05.006

Arhin, S., Noel, E., Anderson, M.F., Williams, L., Ribisso, A., Stinson, R.: Optimization of transit total bus stop time models. J. Traffic Transp. Eng. **3**(2), 146–153 (2016). doi:10.1016/j.jtte.2015.07.001

Amin-Naseri, M.R., Baradaran, V.: Accurate estimation of average waiting time in public transportation systems. Transp. Sci. **49**(2), 213–222 (2014). doi:10.1287/trsc.2013.0514

Parbo, J., Nielsen, O.A., Prato, C.G.: User perspectives in public transport timetable optimisation. Transp. Res. Part C **48**, 269–284 (2014). doi:10.1016/j.trc.2014.09.005

Sun, D., Xu, Y., Peng, Z.R.: Timetable optimization for single bus line based on hybrid vehicle size model. J. Traffic Transp. Eng. **2**(3), 179–186 (2015). doi:10.1016/j.jtte.2015.03.006

Yan, S.Y., Chi, C.J., Tang, C.H.: Inter-city bus routing and timetable setting under stochastic demands. Transp. Res. Part A **40**, 572–586 (2006). doi:10.1016/j.tra.2005.11.006

Vissat, L.L., Clark, A., Gilmore, S.: Finding optimal timetables for Edinburgh bus routes. Electron. Notes Theor. Comput. Sci. **310**, 179–199 (2015). doi:10.1016/j.entcs.2014.12.018

Wu, J.F.: A real-time origin-destination matrix updating algorithm for on-line applications. Transp. Res. Part B Methodol. **31**(5), 381–396 (1997). doi:10.1016/s0191-2615(97)00001-5

Pedrycz, W., Homenda, W.: Building the fundamentals of granular computing: a principle of justifiable granularity. Appl. Soft Comput. **13**(10), 4209–4218 (2013). doi:10.1016/j.asoc.2013.06.017

Liu, B., Liu, Y.K.: Expected value of fuzzy variable and fuzzy expected value models. Trans. Fuzzy Syst. **10**(4), 445–450 (2002). doi:10.1109/tfuzz.2002.800692

Chriqui, C., Robillard, P.: Common bus lines. Transp. Sci. **9**(2), 115–121 (1975). doi:10.1287/trsc.9.2.115

Nguyen, S., Pallottino, A., Malucelli, F.: A modeling framework for passenger assignment on a transport network with timetables. Transp. Sci. **35**(3), 238–249 (2001). doi:10.1287/trsc.35.3.238.10152

Cominetti, R., Correa, J.: Common-lines and passenger assignment in congested. Transp. Sci. **35**(3), 250–267 (2001). doi:10.1287/trsc.35.3.250.10154

Cepeda, M., Cominetti, R., Florian, M.: A frequency-based assignment model for congested transit networks with strict capacity constraints: Characterization and computation of equilibria. Transp. Res. Part B: Methodol. **40**(6), 437–459 (2006). doi:10.1016/j.trb.2005.05.006

Wong, S.C., Tong, C.O.: Estimation of time-dependent origin-destination matrices for transit networks. Transp. Res. Part B: Methodol. **32**(1), 35–48 (1998). doi:10.1016/s0191-2615(97)00011-8

Bertini, R.L., El-Geneidy, A.M.: Modeling transit trip time using archived bus dispatch system data. J. Transp. Eng. **130**(1), 56–67 (2004). doi:10.1061/(asce)0733-947x

Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. University of Michigan Press, Oxford (1975)

Whitley, D.: A genetic algorithm tutorial. Stat. Comput. **4**(2), 65–85 (1994). doi:10.1007/bf00175354

Jones, G., Willett, P., Glen, R.C., Leach, A.R., Taylor, R.: Development and validation of a genetic algorithm for flexible docking. J. Mol. Biol. **267**(3), 727–748 (1997). doi:10.1006/jmbi.1996.0897

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002). doi:10.1109/4235.996017

Pedrycz, W., Song, M.: Granular fuzzy models: a study in knowledge management in fuzzy modeling. J. Approx. Reas. **53**(7), 1061–1079 (2012). doi:10.1016/j.ijar.2012.05.002

Zhou, J.D., Li, X., Kar, S.: Time consistent fuzzy multi-period rolling portfolio optimization with adaptive risk aversion factor. J. Ambient Intell. Hum. Comput, 1–16 (2017). doi:10.1007/s12652-017-0478-4

Zhu, G.Y., Yang, C.G., Huang, D., Zhang, P.: A combined forecasting model of urban rail transit peak-hour cross-section passenger volume. In: Cota International Conference of Transportation Professionals, pp. 1040–1052 (2015)

Solving Dial-A-Ride Problems Using Multiple Ant Colony System with Fleet Size Minimisation

Twinkle Tripathy, Sarat Chandra Nagavarapu, Kaveh Azizian,
Ramesh Ramasamy Pandi, and Justin Dauwels(✉)

Nanyang Technological University,
50 Nanyang Avenue, Singapore 639798, Singapore
{twinkle,sarat,kazizian,jdauwels}@ntu.edu.sg, ramesh006@e.ntu.edu.sg

Abstract. This paper proposes an ant colony optimization (ACO) based algorithm to minimise the fleet size required to solve dial-a-ride problem (DARP). In this work, a static multi-vehicle case of DARP is considered where routes of multiple vehicles are designed to serve customer requests which are known a priori. DARP necessitates the need of high quality algorithm to provide optimal feasible solutions. We employ an improved ACO algorithm called ant colony system (ACS) to solve DARP. The fleet minimisation is also achieved by using ACS. In summary, multiple ACS are employed to minimise the fleet size while generating feasible solutions for DARP. Furthermore, the theoretical results are also validated through simulations.

Keywords: Dial-a-ride problem · Fleet size minimisation · Ant colony optimisation

1 Introduction

People rely on transportation heavily to travel from one place to another as a part of their daily routine. The need for more flexible solutions to provide high-quality service for passengers is of significant importance in the evolution of public transportation systems. Demand-responsive transit of door-to-door type is the preferred form of transportation service especially for elderly or physically challenged people and is operational in several urban areas. Dial-a-ride problem refers to the problem of designing scheduled vehicle routes to serve passenger requests in the form of pick-up and drop-off. It is also called as 'demand responsive transport' wherein the customer demands/requests are specified as pick-up and drop-off locations and their respective time windows. A group of vehicles are dispatched from a starting node labelled as depot, as and when customers request for vehicles.

The passengers intimate the time window for pick-up and drop-off in DARP. This imposes an excessive constraint on the transporter if the time window is very narrow. The service quality criteria include waiting time, travelling time and departure time. One of the key objectives is to use minimum set of vehicles

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_28

to accommodate as many passengers as possible and deliver high quality service by meeting all the constraints. Moreover, minimising the overall travel distance by evading the longest route in tandem with providing superior passenger comfort and safety is also an important factor. In a static multi-vehicle DARP, the passenger demands and the transportation demands are known beforehand and each of the passenger demand has to be served by different vehicles simultaneously and within a specified time window.

Dial-a-ride problem was first introduced by Wilson et al. [1] in 1971. DARP has been formulated mathematically as a special type of VRP [2] taking the service related constraints into account. Several methods such as exact, heuristic and meta-heuristic methods exist in the literature to solve dial-a-ride problems. Healy et al. have developed a sacrificing algorithm [3] to solve DARP by extending the features of local search. Cordeau et al. have mathematically formulated the static DARP as a mixed integer programming problem [4]. A tabu search procedure for DARP [5] was introduced by Cordeau et al. to evaluate the neighbourhood that adjusts the time it takes to reach the nodes on the routes. This procedure has been applied to static multi-vehicle DARP, which enabled them to obtain feasible solutions and an overall improvement in the solution quality. The tabu search is often used to solve Dynamic DARP due to its solution quality but requires longer simulation runtime. Therefore, a parallel computation of tabu search heuristic [6] has been introduced for Dynamic DARP to speed up the algorithm. Several variants of DARP along with the modelling issues involved in problem formulation have been presented in [7]. A grouping genetic algorithm [8] and an adaptive neighbourhood algorithm [9] were developed for static DARP. These algorithms consider only time window constraints and maximum ride time constraints to solve static DARP.

Ant colony optimisation (ACO) is one of the novel meta-heuristic techniques to solve dial-a-ride problems. ACO was first introduced by Dorigo [11] based on the behaviour of real ants. ACO based algorithms have been used to resolve a sizeable number of combinatorial and discrete optimisation problems. They have been classified under the category of algorithms which are called as *approximate* type [15] as they focus mainly on getting good solutions by foregoing guaranteed optimal solution. Two different pheromone update rules [16] have been introduced to evaluate the solution quality and performance of the artificial ants. The elitist ant system [10] has been modified with an evaporation factor [17] for pheromone and an elitist strategy based on ranking has been followed in the implementation of pheromone deposition to improve the solution quality.

The work presented in this paper derives its motivation from the novel work on vehicle minimisation for VRP by Gambardella et al. [13]. In the paper, we extend the vehicle minimisation problem for a static dial-a-ride problems while introducing the modifications necessary for our framework. Owing to the increasing need of energy efficient systems, fleet size or vehicle count minimisation is one of the most important problems of today. In this paper, fleet size minimisation is achieved while minimising the routing cost with higher priority being assigned for the former. Multiple ant colony systems are used to address both of the minimisation problems. The remainder of the paper is organized as follows: Sect. 2 introduces the terminology associated with dial-a-ride problem and ant colony optimisation. Section 3 formulates the dial-a-ride problem. Section 4 details the multiple ant colony system (MACS) algorithm to solve DARPs. It also discusses the MACS based fleet size minimisation algorithm. Section 5 illustrates the simulation results. Conclusion is presented in Sect. 6.

2 Terminology

The vehicle routing terminology used in this paper is presented here.

- *Depot:* The initial and terminal node for each vehicle.
- *Pick-up points (P):* The location at which the passenger is picked up.
- *Drop-off points (D):* The location at which the passenger is dropped-off.
- *Nodes (N):* The set of all pick-up points, drop-off points and depot.
- *Fleet (K):* The set of m vehicles serving the customer requests.
- *Arcs (E):* The set of paths joining the nodes.
- *Route* (T_k): The maximum route duration of k^{th} vehicle.
- *Departure time* (D_i): The time at which a vehicle leaves node i.
- *Arrival time* (B_i): The time at which a vehicle begins its service at node i.
- *Service time (d):* The time taken by the vehicle to provide service at the i^{th} node.
- *Waiting time*(W_i^k): The duration for which vehicle k waits at node i before starting the service.
- *Maximum ride time (L):* The total time spent by each of the passengers in a vehicle.
- L_i^k: The time spent by the i^{th} customer in the k^{th} vehicle.
- *Total distance traveled:* The total distance traveled by all the vehicle to serve the customers.
- *Vehicle load* (Q_i^k): It is the number of passengers on the vehicle k after visiting node i.
- q_i: The load available at the i^{th} node.
- C^K: The capacity of the k^{th} vehicle.
- e_i: It is the earliest time when service can begin at the i^{th} node.
- l_i: It is the latest time beyond which the i^{th} node can not be serviced.
- d_{ij}: The distance between nodes i and j.
- t_{ij}: Time required to travel from the i^{th} to the j^{th} node.
- c_{ij}: Routing cost to travel from the i^{th} to the j^{th} node.
- *Vehicle time* (t^k): It denotes the current time of vehicle k.
- x_{ij}^k: It is a binary decision variable. It is 1 if arc (i, j) is traversed by vehicle k and 0 otherwise.

The next section formulates the dial-a-ride problem.

3 Problem Formulation

A set of n customer requests for vehicles where each customer specifies a pickup and drop-off location pair. Passengers also specify their preferred time windows during which they have to be served. All vehicles start from the depot and return to it once routing is complete. The model for static DARP is formulated on the assumption that the transportation requests are previously known.

In DARP, we assume that a fleet of m vehicles represented by set K have to serve n customer requests. The customer requests lead to n pick up and n drop off nodes. The nodes form a directed graph (N, A) graph where $N = P \cup D \cup \{0, 2n+1\}, P = \{1, \dots, n\}$ and $D = \{n+1, \dots, 2n\}$. Nodes 0 and $2n+1$ represent the origin and destination depots. Any arc (i, j) denotes a path from the i^{th} to the j^{th} node. The time to travel the arc (i, j) is t_{ij} and the cost of routing is c_{ij}. Thus, the formulation of the dial-a-ride problem [14] is as follows,

$$\text{Min} \quad \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} c_{ij} x_{ij}^k, \tag{1}$$

subject to,

$$\sum_{j \in N} x_{0j}^k = 1 \quad \forall k \in K, \tag{2}$$

$$\sum_{i \in D} x_{i,2n+1}^k = 1 \quad \forall k \in K, \tag{3}$$

$$\sum_{j \in N} x_{ji}^k - \sum_{j \in N} x_{ij}^k = 0 \quad \forall i \in P \cup D, k \in K, \tag{4}$$

$$\sum_{k \in K} \sum_{j \in N} x_{ij}^k = 1 \quad \forall i \in P \cup D, \tag{5}$$

$$\sum_{j \in N} x_{ij}^k - \sum_{j \in N} x_{n+i,j}^k = 0 \quad \forall i \in P, k \in K, \tag{6}$$

$$Q_j^k \geq (Q_i^k + q_j) x_{ij}^k \quad \forall i \in N, \forall j \in N, \forall k \in K, \tag{7}$$

$$\max(0, q_i) \leq Q_i^k \leq \min(C_k, C_k + q_i) \quad \forall i \in N, \forall k \in K \tag{8}$$

$$B_j^k \geq (B_i^k + d_i + t_{ij}) x_{ij}^k \quad \forall i \in N, \forall j \in N, \forall k \in K, \tag{9}$$

$$e_i \leq B_i^k \leq l_i \quad \forall i \in N, \forall k \in K, \tag{10}$$

$$L_i^k = (B_{n+i}^k - (B_i^k + d_i)) x_{ij}^k \quad \forall i \in P, \forall k \in K, \tag{11}$$

$$t_{i,n+i} \leq L_i^k \leq L \quad \forall i \in P, \forall k \in K, \tag{12}$$

$$B_{2n+1}^k - B_0^k \leq T_k \quad \forall k \in K, \tag{13}$$

The objective of this model, given by Eq. (1), is to minimise the overall routing cost of all the vehicles required to serve the maximum number of customers. The constraints specified in Eqs. (2) and (3) ensure that all the vehicle routes start and end at the depot. Equations (4) and (5) make it mandatory for every node to be visited by only one vehicle such that any vehicle reaching a particular node must exit it. The customers that are picked up are always delivered to their corresponding drop-off locations as guaranteed by constraint in Eq. (6). Additionally, every vehicle must not exceed its maximum capacity and maximum ride time constraints must hold for the vehicles and the customers.

Equation (7) updates the load of the k^{th} vehicle once it reaches node j from node i. Constraint in Eq. (8) ensures that the capacity of vehicle k is not exceeded at any of the nodes. The time consistency is maintained by Eq. (9) which ensures that node j can be serviced only after it is reached. Equation (10) specifies the time window of the service at each location and makes sure that service begins only within the windows. The constraints in Eqs. (11) and (12) guarantee that the maximum ride time of a customer is not exceeded. Finally, the constraints on the route duration of vehicle k are maintained by following Eq. (13).

Having laid down the constraints, now the next section delves into solving DARP using ACO.

4 Multiple Ant Colony System

In this section, we employ ant colony optimisation (ACO) to solve DARP. ACO is a population based search technique that is widely used to solve combinatorial optimisation problems like vehicle routing, dial-a-ride problem, pick-up and delivery, etc. It was first introduced by Dr. Marco Dorigo in his PhD thesis [11].

ACO techniques are inspired form the natural behaviour of ants which is explained as follows. Ants lay pheromone trails on their paths; if other ants follow that path, they lay more pheromone on the path. The higher the pheromone levels in a path, the higher is the probability of that path to be picked up, resulting in an autocatalytic behaviour. Pheromone trails evaporate with time, so shorter paths can accumulate pheromones faster compared to the longer paths. This phenomenon results in ants being able to narrow down to the shortest routes to their food sources.

Many variants of ACO techniques have been proposed in the literature, like elitist ant system, rank based ant system, max-min ant system, ant colony system (ACS) [12], etc. This work employs ant colony system to solve the static DARP. While solving DARP, multiple iterations of tour building are run. In each iteration, certain number of ants build their tours. The pheromone trails are updated based on the length of the tours and propagated to the next iteration. Pheromone update can be done either when the ants choose the subsequent node (local update) or once the tours are completely built by the ants (global update). In ACS, the global pheromone update happens only for the best route by using the following rule,

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho \varDelta \tau_{ij}^{bs} \tag{14}$$

where $\varDelta \tau_{ij}^{bs} = 1/c_{bs}$ and c_{bs} is the best cost obtained so far in the iterations. The local update of pheromone happens each time a vehicle leaves a particular node by employing the following pheromone evaporation rule,

$$\tau_{ij} = (1-\zeta)\tau_{ij} + \zeta\tau_0. \tag{15}$$

where τ_0 is the initial pheromone level. In standard ACO, an ant selects next node j from any node i based on a probability distribution for which two factors are taken into account, pheromone level τ_{ij} and visibility η_{ij}. The next node j belongs to the neighbourhood $\mathcal{N}_i^k$ of node i. $\mathcal{N}_i^k$ consists of all nodes that can be visited from i without violating the time window constraint of the next node and capacity constraint of vehicle k. Visibility η_{ij} is a design parameter and could be a function of distance, time, etc. depending on the problem in hand. The relative importance of τ_{ij} and η_{ij} in the probability distribution can be controlled by using the gain β. ACS introduces two new parameters: q and q_0. Based on all of these parameters, the next node j is chosen using the following rule,

$$j = \begin{cases} \arg\ \max_{l \in \mathcal{N}_i^k}\{\tau_{il}\ \eta_{il}^{\beta}\} & \text{if } q \leqslant q_0 \\ \mathcal{J} & \text{otherwise} \end{cases} \tag{16}$$

where $\mathcal{J}$ is a random number selected using probability,

$$p_{ij}^k = \frac{\tau_{ij}\eta_{ij}^{\beta}}{\sum_{l \in \mathcal{N}_i^k} \tau_{il}\eta_{il}^{\beta}}. \tag{17}$$

In (16), q is a random variable distributed uniformly in the interval $[0, 1]$ and q_0 is a constant in the interval $[0, 1]$. When $q < q_0$, the ants always choose the node that is favoured by both the pheromone levels and the visibility factors. If $q \geqslant q_0$, then ants choose the next node based solely on the probability distribution. Thus, while choosing the next node, one can explore new paths or focus on the best routes by altering the parameter q_0 and q.

ACS has proved to be a powerful and competitive meta-heuristic algorithm which is performs well in comparison with the existing methods to solve DARP. We aim to minimise the fleet size while solving DARP, motivated by the earlier research by Gambardella et al. in [13] wherein the fleet size is minimised for vehicle routing problem using multiple ACS (MACS). To make the problem more suitable for DARP, we modify the formulation as described elaborately in this section. Along with minimising the routing cost of all the vehicles, the other key objective here is to minimise the fleet size to serve as many customers as possible while meeting all the constraints.

4.1 Fleet Size Minimisation

In order to minimise the fleet size to serve the customers, we formulate the problem as DARP with time windows and hierarchical objectives (DARPTW-HO). There are two objectives here: vehicle count minimisation and routing cost minimisation. The main algorithm relies on two ant colony systems. While one of the ant colony systems ACS-VEI aims to minimise the vehicle count, the other one ACS-TIME strives to minimise the routing cost. The global best solution is updated based on these two systems. Fleet minimisation is given higher priority

Algorithm 1. Master Algorithm

```
Let N be all the nodes to be visited by $m$ vehicles.
while there are unserved feasible nodes do
    For the current node $i^{th}$, the neighbourhood $N_i^k \subset N$ is the set of all nodes that can be selected without violating any constraints.
    The node $j \in N_i^k$ which has the least euclidean distance from $i$ is selected as the next node.
end while
The global solution is initialised with the nearest-neighbour solution.
The number of nodes served $s_{gbest}$ is recorded.
The vehicle_count is set to the available number of vehicles.
stop_flag = 0.
while stop_flag = 0 do
    ACS-TIME is initialised with the global solution.
    if $s_{best} \geqslant s_{gbest}$ then
        if routing cost is reduced then
            Update the global solution.
        end if
    end if
    ACS-VEI is initialised with vehicle_count as input.
    if $v$ <vehicle_count then
        vehicle_count = $v$.
        The global solution is updated.
    else
        stop_count = 1.
    end if
end while
```

compared to routing cost minimisation resulting in a hierarchy in the objectives. So, if the ACS for vehicle minimisation generates a feasible solution that has a cost greater than that given the ACS for routing cost minimisation, the solution of ACS-VEI updates the global solution. For both the ACS, the visibility η_{ij} is modified to incorporate an additional time window factor as explained below,

$$\eta_{ij} = 1/(\max(1, (t^k - e_j))d_{ij}). \tag{18}$$

This modification ensures that the node whose time window is approaching faster than the other nodes has higher probability of being picked up. The ant colony systems ACS-TIME and ACS-VEI are called inside the main algorithm 1. In Algorithm 1, the initial solution is generated using nearest neighbour heuristic. The solution is recorded as the best global solution. Then, ACS-TIME is initialised with the global solution. The global solution is updated only if the solution generated using ACS-TIME is improved. Then, ACS-VEI is initialised with one less than the number of vehicles currently being used. If ACS-VEI generates a feasible solution which serves as many nodes as the those served by the global best solution or more, then the global solution is updated. The process stops only when the ACS-VEI does not give a feasible solution anymore.

Algorithm 2. ACS-TIME

```
Let N be all the nodes to be visited by a fleet of size vehicle_count.
while there are unserved feasible nodes do
    For the current node $i^{th}$, the neighbourhood $N_i^k \subset N$ is the set of all nodes that can be selected without violating any constraints.
    The node $j \in N_i^k$ which has the least euclidean distance from $i$ is selected as the next node.
end while
The local best solution is updated with this solution.
The number of nodes served $s_{best}$ is updated.
The pheromone level is updated using the rule $\tau_{ij} = (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij}^{bs}$ where $\Delta\tau_{ij}^{bs}$ is the local best cost.
while the current iteration number is less than equal to the maximum number of iterations do
    for each ant do
        $N_i^k$ is the feasible nodes that satisfy all the constraints.
        The next node $j \in N_i^k$ is chosen using Eq. (16).
        Pheromone is updated locally as suggested by Eq. (15).
    end for
    The local solution is updated.
    The number of nodes served $s_{local}$ is recorded.
    Pheromone is update globally by following the rule in Eq. (14).
    if local solution is better than the local best solution then
        if $s_{local} \geqslant s_{best}$ then
            The local best solution is updated.
        end if
    end if
end while
```

ACS-TIME is described in Algorithm 2. As mentioned in the previous paragraph, ACS-TIME starts with the solution generated using nearest neighbour heuristic. In nearest neighbour heuristic, the next node is chosen on the basis of distance. The nearest node in the neighbourhood of the current node is chosen as the next node provided it does not violate any of the constraints discussed before. Next, the ACO algorithm attempts to improve this solution. The best solution is generated using ACS-TIME is then output to Algorithm 1.

ACS-VEI is summarised in Algorithm 3. This algorithm attempts to minimise the fleet size such that maximum number of customers can be served while ensuring that none of the constraints are violated. Algorithm 3 starts with one vehicle less than the current vehicle count and initialises the solution using nearest neighbour heuristic. The local best solution and pheromone levels are updated accordingly. This becomes the initial solution for the ACO algorithm and the pheromone levels are initialised according to this solution. Then, the solution is improved by running multiple iterations of the ACO algorithm. The local best solution is updated only if the number of nodes served exceeds those served by the local best solution. If the number of nodes served are the same,

Algorithm 3. ACS-VEI

```
Set $v$ = vehicle_count − 1.
Let N be all the nodes to be visited by a fleet of size $v$.
while there are unserved feasible nodes do
    For the current node $i$, the neighbourhood $N_i^k \subset N$ is the set of all nodes that can be selected without violating any constraints.
    The node $j \in N_i^k$ which has the least euclidean distance from $i$ is selected as the next node.
end while
The local best solution is updated with this solution.
The number of nodes served $s_{best}$ is updated.
The pheromone level is updated using the rule $\tau_{ij} = (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij}^{bs}$ where $\Delta\tau_{ij}^{bs}$ is the local best cost.
while the current iteration number is less than equal to the maximum number of iterations do
    for each ant do
        $N_i^k$ is the feasible nodes that satisfy all the constraints.
        The next node $j \in N_i^k$ is chosen using Eq. (16).
        Local pheromone update happens using Eq. (15).
        Number of served nodes $s_{local}$ is updated.
        if $s_{local} \geqslant s_{best}$ then
            The local best solution is updated.
            $s_{best} = s_{local}$
        end if
    end for
    The local solution is updated.
    Global pheromone update is made using Eq. (14).
end while
if $s_{best} < s_{gbest}$ then
    v = vehicle_count.
end if
```

then the one which has smaller cost is given priority. This best solution is then fed back to Algorithm 1. The algorithms stop running once the solution generated by ACS-VEI does not improve anymore. This would mean that by reducing the fleet size any further, the algorithms become unable to serve as many nodes as they could serve before the fleet size reduction. This ensures that fleet size minimisation is always given higher priority. Now in the next section, we proceed to validate the algorithms by means of simulations.

5 Simulation Results

In this section, we analyse the performance of the proposed fleet size minimisation algorithm for DARPTW-HO. Simulations have been carried out in MATLAB R2016b on an Intel i5 processor running on Mac-OS with 4GB RAM. The results are performed on the standard benchmark DARP instances pr01 and pr02 [18]. The tuning of the parameters is carried out as suggested by Dorigo and Stutzle [10]. The vehicle routes generated using MACS are plotted in Fig. 1 where the nodes in black, red and green represent the depot, the pick up points and the drop-off points, respectively. The route in blue represents the sequence of nodes served by the first vehicle and the route in orange denotes the same for the second vehicle. We benchmark the performance of the algorithm using nearest neighbour heuristic. The benchmark instances are input to Algorithm 1. This algorithm activates the algorithms for fleet size or routing cost minimisation. In this case, both the fleet size and routing cost minimisation are done using nearest neighbour approaches. The global solution is updated only when the fleet size and/or routing cost minimisation algorithm generates a superior solution.

Case 1: Test instance pr01

In this dataset, there are 24 requests resulting 49 nodes comprised of 24 pick up points, 24 drop-off points and a depot. There are 3 vehicles of capacity $C_k = 6$ with maximum route duration of $T_k = 480\,\text{min}$ for all the vehicles $k \in \{1, 2, 3\}$. Ride time constraint of each customer is $L = 90\,\text{min}$.

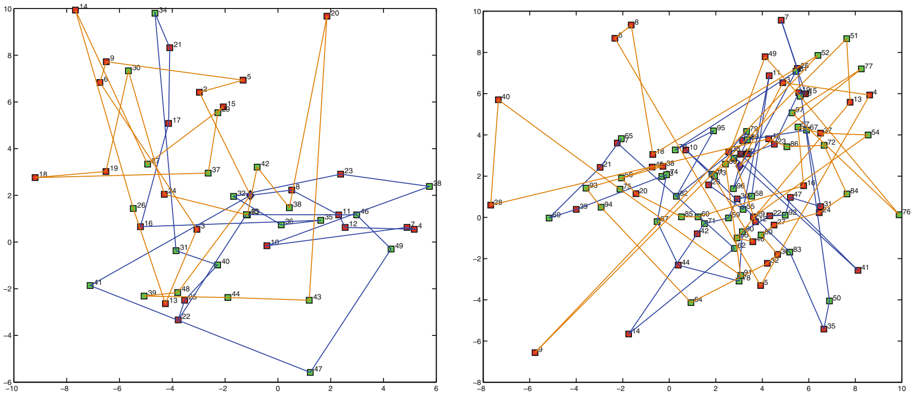
Multiple Ant Colony System
The parameters of the ACS are chosen as follows: $\rho = 0.5$, $\zeta = 0.1$, $\beta = 5$, $q_0 = 0.45$. The optimal fleet size for this problem instance turns out to be 2 for which the routing cost is 205 km. These results clearly indicate that the fleet size that needs to serve all the requests can be reduced to 2. The vehicle routes for this example are illustrated in Fig. 1(a).

Nearest Neighbour
The nearest neighbour heuristic gives an optimal vehicle of 3 and the resulting routing cost as 227.3 km. In this case, it is not optimal to reduce the fleet size as any reduction in fleet size reduces the number of served requests.

Case 2: Test instance pr02

In this dataset, there are 48 requests resulting 97 nodes comprised of 48 pick up points, 48 drop-off points and a depot. There are 5 vehicles of capacity $C_k = 6$ with a maximum route duration of $T_k = 480\,\text{min}$ for all the vehicles $k \in \{1, 2, 3\}$. Ride time constraint of each customer is $L = 90\,\text{min}$.

Multiple Ant Colony System
The parameters of the ACS are chosen as follows: $\rho = 0.5$, $\zeta = 0.1$, $\beta = 5$, $q_0 = 0.45$. The optimal fleet size for pr-02 is 2 with the corresponding routing cost as 371.06 km. For this particular case, the results show that the fleet size can be reduced to 2 while ensuring that all the requests are served. This is validated from the vehicle routes plotted in Fig. 1(b).



(a) Test instance pr01.

(b) Test instance pr02.

Fig. 1. Vehicles' routes.

Nearest Neighbour
The instances are now tested using nearest neighbour heuristic and the resulting optimal fleet size is 5 with a routing cost of 467.17 km. In this scenario, any reduction of fleet size reduces the total number of nodes served. Hence, the optimal fleet size is same as the available fleet size.

6 Conclusion

DARP is an extremely challenging routing problem that addresses the issues faced by transportation industry. The success of DARP lies in the effective utilisation of ride time, finding suitable routes to minimise the routing cost and using a fleet of minimum possible size. Fleet minimisation is a very crucial aspect not only for the transportation industry but also for the whole world as minimum fuel consumption is the need of the hour! This paper implements an ant colony optimisation based algorithm to solve dial-a-ride problem while minimising the fleet size. The algorithm mainly consists of two ACS, one to minimise the routing cost and the other to minimise the fleet size. Given any problem instance, the algorithms presented in the paper can indicate whether or not the fleet size can be minimised while serving maximum requests. Furthermore, the theoretical results are also validated through various simulations carried out in MATLAB. This work can be extended by performing multi-objective optimisation for dial-a-ride problems to minimise the fleet size as well as the routing cost and incorporating dynamic requests.

References

1. Wilson, N.H., Sussman, J.M., Wong, H.K., Higonnet, T.: Scheduling algorithms for a dial-a-ride system. Massachusetts Institute of Technology, Urban Systems Laboratory (1971)

2. Psaraftis, H.N.: An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. Transp. Sci. **17**(3), 351–357 (1983)
3. Healy, P., Moll, R.: A new extension of local search applied to the dial-a-ride problem. Eur. J. Oper. Res. **83**(1), 83–104 (1995)
4. Cordeau, J.-F., Laporte, G.: The dial-a-ride problem: models and algorithms. Ann. Oper. Res. **153**(1), 29–46 (2007)
5. Cordeau, J.-F., Laporte, G.: A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transp. Res. Part B: Methodol. **37**(6), 579–594 (2003)
6. Attanasio, A., Cordeau, J.-F., Ghiani, G., Laporte, G.: Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. Parallel Comput. **30**(3), 377–387 (2004)
7. Cordeau, J.-F., Laporte, G.: The dial-a-ride problem (DARP): variants, modeling issues and algorithms. 4OR: A Q. J. Oper. Res. **1**(2), 89–101 (2003)
8. Rekiek, B., Delchambre, A., Saleh, H.A.: Handicapped person transportation: an application of the grouping genetic algorithm. Eng. Appl. Artif. Intell. **19**(5), 511–520 (2006)
9. Pisinger, D., Ropke, S.: A general heuristic for vehicle routing problems. Comput. Oper. Res. **34**(8), 2403–2435 (2007)
10. Dorigo, M., Stutzle, T.: Ant Colony Optimization. The MIT Press (2004)
11. Dorigo, M.: Optimization, learning and natural algorithms, Ph.D. thesis, Politecnico di Milano, Italy (1992)
12. Dorigo, M., Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput. **1**(1), 53–66 (1997)
13. Gambardella, L.M., Taillard, E., Agazzi, G.: MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale (1999)
14. Paquette, J., Cordeau, J.-F., Laporte, G., Pascoal, M.M.B.: Combining multicriteria analysis and tabu search for dial-a-ride problems. Transp. Res. Part B: Methodol. **52**, 1–16 (2013)
15. Blum, C.: Ant colony optimization: introduction and recent trends. Phys. Life Rev. **2**(4), 353–373 (2005)
16. Tan, W.F., Lee, L.S., Majid, Z.A., Seow, H.V.: Ant colony optimization for capacitated vehicle routing problem. J. Comput. Sci. **8**(6), 846–852 (2012)
17. Bullnheimer, B., Hartl, R.F., Strauss, C.: Applying the ant system to the vehicle routing problem. In: Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization, pp. 285–296. Kluwer Academic Publishers, Dordrecht (1999)
18. http://alpha.uhasselt.be/kris.braekers/

Bus Scheduling Timetable Optimization Based on Hybrid Bus Sizes

Haitao Yu[1,2], Hongguang Ma[3], Hejia Du[4], Xiang Li[4], Randong Xiao[2], and Yong Du[2](✉)

[1] School of Computer Science and Engineering, Beihang University, Beijing 100191, China
[2] Beijing Transportation Information Center, Beijing 100161, China
duyong@bjjtw.gov.cn
[3] School of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China
[4] School of Economics and Management, Beijing University of Chemical Technology, Beijing 100029, China

Abstract. For bus carriers, it is the most basic and important problem to create the bus scheduling timetable based on bus fleet configuration and passenger flow demand. Considering different technical and economic properties, vehicle capacities and limited available number of heterogeneous buses, as well as the time-space characteristics of passenger flow demand, this paper focuses on creating the bus timetables and sizing the buses simultaneously. A bi-objective optimization model is formulated, in which the first objective is to minimum the total operation cost, and the second objective is to maximum the passenger volume. The proposed model is a nonlinear integer programming, thus a genetic algorithm with self-crossover operation is designed to solve it. Finally, a case study in which the model is applied to a real-world case of a bus line in the city of Beijing, China, is presented.

Keywords: Bus timetable · Hybrid sizes · Load factor · Fleet configuration

1 Introduction

Public transit planning is of great importance for a bus carrier to improve its operation efficiency, service quality and market competitiveness. Timetable design is an vital part of public transit planning, in which the departure time of each bus trip is determined. Bus timetabling is a principal stage, since its solution determines transit service quality and subsequent subproblems (i.e., the vehicle and crew scheduling).

Lots of previous researches have focused on bus timetable design problem with consideration of passenger waiting time, in-vehicle time, operation cost and so on. Yan et al. (2006) assumed the passengers' demands are stochastic and established a model to minimize the operation cost. In order to increase

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_29

the bus carrier's service reliability, Yan et al. (2012) developed a robust model to minimize the total expected value of the random schedule deviation and its variability. Aiming to reduce the financial risk of penalties and increase the punctuality and reliability of service, Vissat et al. (2015) proposed a stochastic model of a particular bus route to optimize the timetable. Wong and Tong (1999) assumed the waiting time, walking time, in-vehicle time are random variables, and proposed a dynamic transit assignment model based on a network. Ceder et al. (2001) optimized the synchronization of a transit network to minimum the waiting time of passengers at transfer nodes. Yan and Chen (2002) proposed a model from the bus carrier's perspective to maximizing its profit. Wu et al. (2016) optimized the timetable of a bus network based on a draft timetable.

The timetable optimization is really important in bus scheduling. However, the majority of previous literatures focused on setting optimized timetables to minimum the cost of bus carrier and passengers with a certain bus size whose capacity is specific. Since the passengers demand fluctuates at different time periods, the bus operating efficiency is challenged, and the resource wastes occur in off-peak hours frequently. Current timetables are commonly designed to feature even-headway departures which cannot guarantee efficient operation when the passengers demand fluctuates (Ceder et al. 2013). Most of the existing studies on timetable optimization models researched a certain type of vehicles and a few considered of hybrid vehicles. Sun et al. (2015) built three different models for hybrid vehicles, large vehicles and small vehicles to tackle the passengers demand fluctuations in transit operation and get the optimized timetable. The results showed that the hybrid vehicle sizes model excels the other two models both in the total time and total cost. But the capacity of the vehicles is neglected in their study.

In this study, we work on optimizing the scheduling timetable with hybrid bus sizes. Based on the analysis of passenger flow and a specific constraint of load factor, the different sizes of buses are chosen to be sent out according to the timetable. An optimization model is constructed to minimum the operation cost and maximize the passenger volume under some public welfare constraints. The operation cost is concerned with the size of bus. As for passenger volume, we obtain the origin-destination (O-D) flow matrices based on the historical IC data firstly, and then the passenger volume is determined according to a classical passenger assignment model.

The rest of this paper is organized as follows. Section 2 describes the problem and proposes the mathematic model. In Sect. 3, a genetic algorithm with self-crossover operation is designed to solve the proposed model. Section 4 presents a case study to illustrate the efficiency of our model. Finally, Sect. 5 is the conclusion.

2 Problem Description and Model

In this section, we establish the optimization model for creating the bus timetables and sizing the buses simultaneously. The hybrid-sizes bus timetable optimization problem can be described as: heterogeneous buses with different vehicle capacity are configured, and the number of each size of buses is fixed. The bus carrier needs to determine the buses' departure timetable and the corresponding hybrid sizes combination plan. In the decision-making process, the following aspects must be considered simultaneously: (i) the operation cost of each size of buses; (ii) the practical operation requirement of departure time interval; (iii) the demand of passenger flow; (iv) the passengers' comfort level on the bus, e.g. the buses' load factors. The goal of the bus carrier is to minimum the total operation cost and maximum the passenger volume. The operation cost includes energy cost, personnel wage cost, maintenance cost and so on. For different sizes of buses, the corresponding operation costs are different.

2.1 Assumptions

Without loss of generality, we study a direction of a bus route and the optimization method of the other one direction is similar. In order to formulate this problem briefly, the following assumptions are made:

1. The operation parameters (i.e., speed, acceleration, etc.) are assumed to be equal for all vehicle size buses in the study;
2. The passengers' demand will not be affected by the frequency of the buses;
3. No quantity restrictions in the use of any size buses.

2.2 Notations

To describe the model conveniently, the notations we used are listed in Table 1.

Table 1. List of notations

Notations	
Sets	
I	The set of bus stops, $i, j \in I$
K	The set of bus trips, $k \in K$
L	The set of time intervals, $l \in L$
V	The set of bus sizes, $v \in V$
Parameters	
T_i^{i+1}	The travel time from stop i to stop $i+1$ in time interval l
P_{lij}	The total number of passengers from stop i to stop j in time interval l
m_{lij}	The number of common buses from stop i to stop j in time interval l
$\overline{C}_v$	The maximum allowing number of passengers of a v-th size bus
α_0	The standard load factor
Δt_{min}	The minimum departure time interval
Δt_{max}	The maximum departure time interval
c_v	The operation cost per trip of a v-th size bus
Decision variables	
δ_v^k	1, if a v-th size bus is used for the k-th trip; 0, otherwise
t_{k1}	The departure time for the k-th bus trip

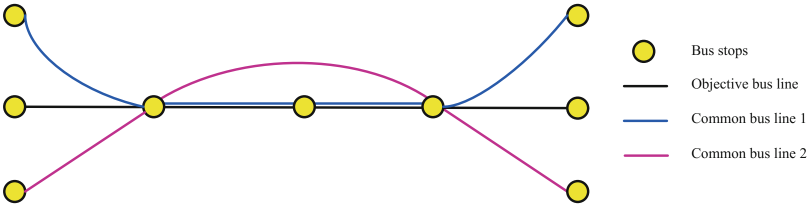


Fig. 1. Common bus lines

2.3 Objectives

- **Operation Cost** C

The operation cost consists of lots of parts, such as fuel, administrative and maintenance costs that vary with usage and the wages of the crew that operates the vehicle (Hurdle 1973). Since we are not focused on the concrete calculation, we simply the operation cost as follows

$$C = \sum_{k=1}^{K} \sum_{v=1}^{V} c_v \delta_v^k, \tag{1}$$

where c_v is the operation cost per trip of the v-th size bus, and δ_v^k is a 0–1 decision variable to represent if the v-th size bus is used for the k-th trip.

- **Passenger Volume** P

Since every bus line is not isolated but in a transport network, we forecast the passenger volume of the researched bus line in a transit network. The bus line must be influenced by its common bus lines. Common bus lines are those routes sharing common sections and the passengers must select which he probably takes (Chriqui and Robillard 1975), as shown in Fig. 1.

Niu and Zhou (2013) proposed that the multi-phase timetable which divides a day into several time blocks and applies the even vehicle-departing interval for each period, may accommodate peak-hour demand while maintaining a certain level of service for passengers boarding at off-peak hours. In this study, we divide the total operation time T into several time intervals $[T_{ll}, T_{lr}), l = 1, 2, \cdots, L$ with equal-length. The number of buses n_{li} arriving at stop i in time interval l can be calculated by

$$n_{li} = \sum_{k=1}^{K} sgn\{(T_{ll} - t_{ki} - \epsilon)(t_{ki} + \epsilon - T_{lr})\}, \tag{2}$$

in which $sgn(x) = \begin{cases} 1, x > 0 \\ 0, x < 0 \end{cases}$ is a sign function, and ϵ is a very small positive number, which can ensure that $T_{ll} - t_{ki} - \epsilon$ and $t_{ki} + \epsilon - T_{lr}$ won't be equal to zero.

A classical passenger assignment model on a transport network is studied by many researchers (Nguyen et al. 2001; Cominetti and Correa 2001; Cepeda et al. 2006), as follows

$$y_i = \sum_{j \in S} y_j \frac{f_i}{\sum_{j \in S} f_j}$$

where s is the set of strategies which can be chosen by passenger, $i \in s$ donates bus line in a transport network, f_i represents a differentiable effective frequency function of line i. And the probability of buses' arriving of line i is $f_i / \sum_{j \in S} f_j$. In this study, the passenger assignment model is used to forecast passenger volume on a bus line. Based on the historical data, we can obtain the passenger volume P_{lij} from stop i to stop j in time interval l. Then the total passenger volume can be calculated as

$$P = \sum_{l=1}^{L} \sum_{i=1}^{I-1} \sum_{j=i+1}^{I} \frac{n_{li}}{m_{lij} + n_{li}} P_{lij}. \tag{3}$$

2.4 Load Factor Constraint

We cannot only pursue the maximum of profits but have to seek for the balance between the interests of bus carrier and passengers for two reasons: (i) the public transit has the property of the basic social public welfare and the load factor must conform with regulations prescribed by the government; (ii) the passengers' safety and comfort influence the service quality of a bus line. The more passengers in a vehicle, the less service quality is. In the proposed model, we limit the load factor less than the prescribed standard.

For the k-th trip, the number of staying-on passengers at stop $i' \in I$ is

$$S_{ki'} = \sum_{i=1}^{i=i'-1} \sum_{j=i'+1}^{j=I} \frac{1}{m_{lij} + n_{li}} P_{lij}, \quad l = \left\lfloor \frac{t_{ki'}}{\delta} \right\rfloor + 1. \tag{4}$$

The number of boarding passengers at stop $i' \in I$ is

$$B_{ki'} = \sum_{j=i'+1}^{j=I} \frac{1}{m_{li'j} + n_{li'}} P_{li'j}, \quad l = \left\lfloor \frac{t_{ki'}}{\delta} \right\rfloor + 1. \tag{5}$$

Therefore, the section passenger volume between stop i' and stop $i'+1$ for the k-th trip is

$$q_{ki'} = S_{ki'} + B_{ki'} = \sum_{i=1}^{i=i'} \sum_{j=i'+1}^{j=I} \frac{1}{m_{lij} + n_{li}} P_{lij}, \quad l = \left\lfloor \frac{t_{ki'}}{\delta} \right\rfloor + 1. \tag{6}$$

Finally, we formally propose the model as follows

$$\min\ C \tag{7}$$

$$\max\ P \tag{8}$$

$$\text{s. t.}\ \sum_{v=1}^{V} \delta_v^k = 1, \quad \forall k \tag{9}$$

$$q_{ki}\delta_v^k \leq \alpha_0 * \overline{C}_v, \quad \forall k, v, i \tag{10}$$

$$t_{k,i-1} + T_{i-1}^{i}(t_{k,i-1}) = t_{ki}, \quad \forall k, i = 2, 3, \ldots, I \tag{11}$$

$$T_{i-1}^{i}(t_{k,i-1}) = \begin{cases} t_1^{i-1,i},\ t_{k,i-1} \in [T_s, T_s + \delta) \\ t_2^{i-1,i},\ t_{k,i-1} \in [T_s + \delta, T_s + 2\delta) \\ \qquad \vdots \\ t_N^{i-1,i},\ t_{k,i-1} \in [T_s + (N-1)\delta, T_s + N\delta) \end{cases} \tag{12}$$

$$\Delta t_{min} \leq t_{k1} - t_{k-1,1} \leq \Delta t_{max}, \quad k = 2, 3, \ldots, K \tag{13}$$

$$t_{11} = T_s, \quad t_{K1} = T_e, \tag{14}$$

$$\delta_v^k \in \{0, 1\}. \tag{15}$$

The proposed model is a bi-objective optimization model, in which the first objective (7) is to minimum the total operation cost, and the second objective (8) is to maximum the passenger volume. Constraint (9) guarantees that only one type of bus can be used for each trip. Constraint (10) is load factor constraint. Constraint (11) guarantees that the arriving time at a stop is the sum of the departure time at the last stop and the travel time between these two stops. Function (12) defines the travel time between two adjacent stops in each time interval. Constraint (13) guarantees two adjacent bus trips must satisfy the maximum and minimum schedule intervals. Constraint (14) specifies the departure times of the first and last bus trips in a day. Constraint (15) specifies the 0–1 decision variables.

3 Model Transform and Solution

The proposed model (7)–(15) is a nonlinear integer programming. In order to solve the proposed model, we use the weighted method to transform the proposed bi-objective model into a single-objective one. And because the two objectives have different ranges, we firstly normalized them by introducing $P_{max}, P_{min}, C_{max}, C_{min}$. Here, P_{max} and P_{min} denote the maximum and minimum total passenger volume, and C_{max} and C_{min} denote the maximum and minimum total cost. Finally, the proposed model (7)–(15) is formulate the following single-objective optimization model:

$$\begin{aligned} \max\ & \lambda \frac{P - P_{min}}{P_{max} - P_{min}} - (1-\lambda)\frac{C - C_{min}}{C_{max} - C_{min}} \\ \text{s. t.}\ & \mathit{Constraints}\ \ (9) - (15) \\ & \lambda \in [0, 1] \end{aligned} \tag{16}$$

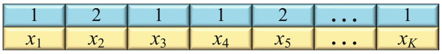
where λ denotes the preference of the decision-maker on the two objectives. If the first objective is more important than the second one, set $\lambda > 0.5$; If the two objectives are equally important, set $\lambda = 0.5$; Otherwise, set $\lambda < 0.5$.

3.1 Genetic Algorithm with Self-crossover Operation

In the section, we will design a genetic algorithm to solve the single-objective model (16). Genetic algorithm (GA) is a computational model for simulating the biological evolution process of natural selection and genetic mechanism. Since genetic algorithm first proposed by Holland (1975), it has been widely studied, experimented and applied by many researchers (Whitley 1994; Jones et al. 1997; Deb et al. 2002).

- **Representation Structure**

The chromosome v is designed as a structure of two rows, in which the first row consists the bus size of each trip, and the second row consists of the departure time of each trip. As shown in Fig. 2, K is the total bus trip times of a day, $x_1 = T_s$ and $x_K = T_e$ are respectively used to denote the departure time of the first and last bus trip in a day. There are two kinds of bus sizes, in which 1 represents the first size bus is used, and 2 represents the second size bus is used.



1	2	1	1	2	...	1
x_1	x_2	x_3	x_4	x_5	...	x_K

Fig. 2. The structure of chromosome

- **Initialization**

It is difficult to generate a feasible chromosome satisfying the load factor constraint (10). In order to obtain a feasible chromosome easilier, we use penalty function method to add constraint (10) into the objective function, as follows

$$\begin{aligned} \max \ & \lambda\frac{P-P_{min}}{P_{max}-P_{min}} - (1-\lambda)\frac{C-C_{min}}{C_{max}-C_{min}} - M\sum_{k\in K}\sum_{v\in V}\sum_{i\in I}(q_{ki}\delta_v^k - \alpha_0 * \overline{C}_v)^+ \\ \text{s. t. } & \mathit{Constraints} \quad (9),(11)-(15) \qquad\qquad (17) \\ & \lambda \in [0,1] \end{aligned}$$

in which M is a very large number, and represents the penalty coefficient in violation of the load factor constraint.

Define an integer *pop_size* as the size of population. To generate a feasible chromosome, x_1 denotes the departure time of first bus trip in a day, i.e., $x_1 = T_s$. x_K denotes the departure time of last bus trip in a day, i.e., $x_K = T_e$. Use the MATLAB function *randfixedsum* to randomly generate $K-1$ numbers u_i $(i = \{1, 2, \cdots, K-1\})$ from $[H_{min}, H_{max}]$ satisfying $\sum_{i=1}^{K-1} u_i = T_e - T_s$, and set $x_{i+1} = x_i + u_i, i = \{1, 2, \cdots, K-2\})$ to obtain a feasible chromosome. Repeat the above procedures *pop_size* times to generate the initialized population v_i, $i = 1, 2, \cdots, pop\_size$.

- **Evaluation Function**

Evaluation function assigns each chromosome a probability of reproduction so that its likelihood of being selected is proportional to its fitness relative to the other chromosomes in the population. That is, the chromosomes with higher fitness will have more chance to produce offspring.

For this programming problem, a chromosome with a larger objective value is better. Rearrange the *pop_size* chromosomes from good to bad based on their objective values. Then the fitness of the ith chromosome v_i can be calculated based on the evaluation function, defined as follows

$$Eval(v_i) = \alpha(1-\alpha)^{i-1}, \quad i = 1, 2, \ldots, pop\_size,$$

in which α is a real number in $(0, 1)$.

- **Selection Process**

The method of spinning the roulette wheel is used here to select chromosomes which breed a new generation. First, calculate the reproduction probability q_i for each chromosome $v_i, i = 1, 2, \cdots, pop\_size$, as follows

$$q_0 = 0, \quad q_i = \sum_{j=1}^{i} Eval(v_j), \quad i = 1, 2, \ldots, pop\_size.$$

Second, generate a random number r in $(0, q_{pop\_size}]$, and select the chromosome v_i such that $q_{i-1} < r \leq q_i$. Repeat the second step *pop_size* times and obtain *pop_size* chromosomes. It is obvious that the chromosomes with larger fitness are typically more likely to be selected.

- **Self-crossover Process**

Crossover is one of the mainly used operations for generating a second population. Here, we propose a self-crossover operation. Randomly select a chromosome v as parent. The probability of crossover is P_c. Generate a random number r from $[0, 1]$, and if $r \leq P_c$, conduct the following crossover operation. Firstly, randomly select two disjoint segments with length of one hour from the operation time $[T_s, T_e]$. Secondly, for the selected chromosome, judge the corresponding sections in the two segments based on the second rows. Thirdly, exchange the two corresponding sections and regenerate the corresponding second rows subject to the trip number and maximum and minimum headway constraints. A child chromosome will be finally obtained, as shown in Fig. 3. Repeat the above process $\frac{1}{2} * pop\_size$ times. The self-crossover can make the child chromosome satisfy the number of bus trips, that is, won't damage the structure of chromosome.

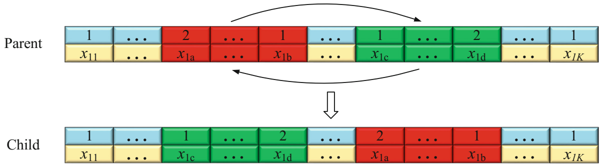

Parent
1 … 2 … 1 … 1 … 2 … 1
x_{11} … x_{1a} … x_{1b} … x_{1c} … x_{1d} … x_{IK}
Child
1 … 1 … 2 … 2 … 1 … 1
x_{11} … x_{1c} … x_{1d} … x_{1a} … x_{1b} … x_{IK}

Fig. 3. Self-crossover operation

- **Mutation Process**

For each chromosome, the probability of mutation is P_m. Randomly generate a real number r from $[0, 1]$. If $r < P_m$, the chromosome will conduct the mutation operation. Firstly, randomly select N time periods from the operation time $[T_s, T_e]$. Secondly, judge and determine the corresponding sections in the periods. Thirdly, regenerate the corresponding sections. The corresponding first rows are randomly regenerated a series of numbers from set $\{1, 2\}$. The corresponding second rows are regenerated subject to the trip number and maximum and minimum headway constraints. A new chromosome will be obtained, as shown in Fig. 4.

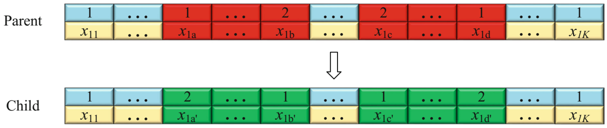

Parent
1 … 1 … 2 … 2 … 1 … 1
x_{11} … x_{1a} … x_{1b} … x_{1c} … x_{1d} … x_{IK}
Child
1 … 2 … 1 … 1 … 2 … 1
x_{11} … $x_{1a'}$ … $x_{1b'}$ … $x_{1c'}$ … $x_{1d'}$ … x_{IK}

Fig. 4. Mutation operation

- **General Procedure**

Following selection, crossover and mutation operations, a new population is generated. Genetic algorithm will terminate after a given number of cyclic iterations of the above steps. The general procedure for genetic algorithm are summarized in Algorithm 1.

Algorithm 1.	Genetic Algorithm
Step 1	Randomly initialize *pop_size* chromosomes
Step 2	Calculate the objective values for all chromosomes
Step 3	Evaluate the fitness of each chromosome via the objective values
Step 4	Select the chromosomes by spinning the roulette wheel
Step 5	Update the chromosomes by using crossover and mutation
Step 6	Repeat Step 2 to Step 5 for a given number of cycles G
Step 7	Report the best found chromosome as the optimal solution

Remark 1. Consider that the self-crossover operation is an internal crossover, we can properly increase the probability of mutation P_m in order to generate more new chromosomes in the practical application.

4 Case Study

In this section, we validate the proposed model based on Yuntong 128 bus line in Beijing, China. Yuntong 128 bus line runs from Beijing Business School Station to Laiguangying North Station. It is 21.44 km long with 31 bus stops. The operation time per day is from 5:30 to 22:00. According to the requirement of working strength of the employees (including drivers, conductors, dispatchers, etc.), the bus line must finish 74 trips per direction every day.

The field data about Yuntong 128 bus line were obtained from Beijing Transportation Information Center, including the IC card data and GPS data. We partition the whole day (5:30–24:00) into 74 time segments with equal length of 15 min. After statistically processing the field data, we can obtain passengers' O-D matrix $[Q_{lij}]_{74\times31\times31}$, running time matrix $T = [t_{li}]_{74\times31}$ and common line bus number matrix $[m_{lij}]_{74\times31\times31}$, which means the passenger number from stop i to stop j in the l-th time segment, the running time from stop i to stop $i+1$ in the l-th time segment, and the common line bus number from stop i to stop j in the l-th time segment, respectively. The other related parameters in the model are obtained from the bus carrier, as follows $\Delta t_{min} = 8, \Delta t_{max} = 20, \alpha_0 = 0.7$.

Consider that there are two different sizes of buses available. The maximum allowing number of passengers of the big ones is 200, and the operation cost per trip is 130, while the maximum allowing number of passengers of the small ones is 100, and the operation cost per trip is 80. Based on the bus carrier' preference and

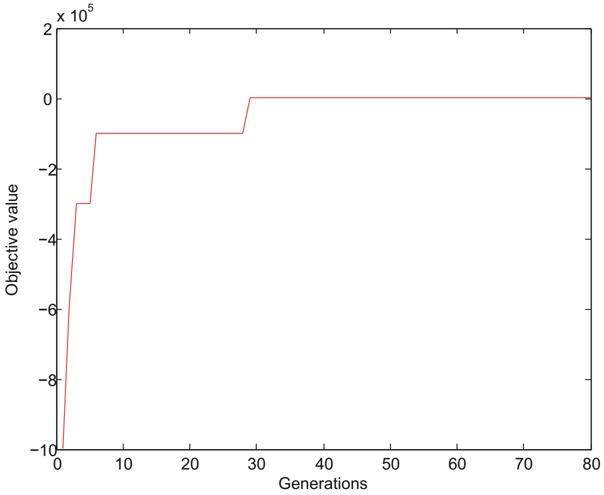


Fig. 5. The convergence of GA

experiences, we set the preference on the operation cost objective and passenger volume objective $\lambda = 0.7$. A large number of experiments were carried out to determine the parameters in the GA. The following parameters with better effect were obtained as follows: $G = 100, pop\_size = 200, P_c = 0.8$, and $P_m = 0.7$. The penalty coefficient M in violation of the load factor constraint was set as 100000.

We run the proposed GA with the parameters set above and obtain the optimal timetable. The total passenger volume Q is 3829 and the total operation cost is 6820. The convergence of objective value is shown in Fig. 5 which indicates the proposed GA is effective to solve the proposed model.

5 Conclusion

In this study, a bi-objective optimization model was proposed for creating the bus timetables and sizing the buses, simultaneously. The first objective is to minimum the total operation cost, and the second objective is to maximum the passenger volume. A genetic algorithm with self-crossover operation was designed to solve the proposed model. Finally, a real case study was presented to show the efficiency of the proposed model and algorithm.

References

Yan, S.Y., Chi, C.-J., Tang, C.H.: Inter-city bus routing and timetable setting under stochastic demands. Transp. Res. Part A **40**, 572–586 (2006)

Yan, Y.D., Meng, Q., Wang, S.A., Guo, X.C.: Robust optimization model of schedule design for a fixed bus route. Transp. Res. Part C **25**, 113–121 (2012)

Vissat, L.L., Clark, A., Gilmore, S.: Finding optimal timetables for Edinburgh bus routes. Electron. Notes Theoret. Comput. Sci. **310**(310), 179–199 (2015)

Wong, S.C., Tong, C.O.: A stochastic transit assignment model using a dynamic schedule-based network. Transp. Res. Part B **33**, 107–121 (1999)

Ceder, A., Golany, B., Tal, O.: Creating bus timetables with maximal synchronization. Transp. Res. Part A **35**, 913–928 (2001)

Yan, S.Y., Chen, H.L.: A scheduling model and a solution algorithm for inter-city bus carriers. Transp. Res. Part A **36**, 805–825 (2002)

Wu, Y.H., Yang, H., Tang, J.F., Yu, Y.: Multi-objective re-synchronizing of bus timetable: model, complexity and solution. Transp. Res. Part C **67**, 149–168 (2016)

Ceder, A., Hassold, S., Dunlop, C., Chen, I.: Improving urban public transport service using new timetabling strategies with different vehicle sizes. Int. J. Urban Sci. **17**(2), 239–258 (2013)

Sun, D., Xu, Y., Peng, Z.R.: Timetable optimization for single bus line based on hybrid vehicle size model. J. Traffic Transp. Eng. **2**(3), 179–186 (2015)

Hurdle, V.F.: Minimum cost schedules for a public transportation route. Transp. Sci. **7**(2), 109–137 (1973)

Chriqui, C., Robillard, P.: Common bus lines. Transp. Sci. **9**(2), 115–121 (1975)

Niu, H.M., Zhou, X.S.: Optimizing urban rail timetable under time-dependent demand and oversaturated conditions. Transp. Res. Part C **36**, 212–230 (2013)

Nguyen, S., Pallottino, A., Malucelli, F.: A modeling framework for passenger assignment on a transport network with timetables. Transp. Sci. **35**(3), 238–249 (2001)

Cominetti, R., Correa, J.: Common-lines and passenger assignment in congested. Transp. Sci. **35**(3), 250–267 (2001)

Cepeda, M., Cominetti, R., Florian, M.: A frequency-based assignment model for congested transit networks with strict capacity constraints: characterization and computation of equilibria. Transp. Res. Part B: Methodol. **40**(6), 437–459 (2006)

Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. University of Michigan Press, Oxford (1975)

Whitley, D.: A genetic algorithm tutorial. Stat. Comput. **4**(2), 65–85 (1994)

Jones, G., Willett, P., Glen, R.C., Leach, A.R., Taylor, R.: Development and validation of a genetic algorithm for flexible docking. J. Mol. Biol. **267**(3), 727–748 (1997)

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)

Supplier's Information Strategy in the Presence of a Dominant Retailer

Ye Wang, Wansheng Tang(✉), and Ruiqing Zhao

Institute of Systems Engineering, Tianjin University, Tianjin 300072, China
{yewang,tang,zhao}@tju.edu.cn

Abstract. Speedy development of the large-sized retail outlets empowers the emergence of dominant retailers, as a result of power transformation from suppliers to retailers. In this paper, we model a market comprised of a dominant entrant retailer, a weaker incumbent counterpart, and a common supplier from which both retailers source products. The retailers are quantity-competing, and the dominant retailer is entitled to determine the wholesale price it purchases, while the incumbent retailer accepts the price offered by the supplier. Besides, the incumbent retailer is assumed to hold private information about market demand. We investigate the collaboration strategy for the supplier which either cooperates with the dominant entrant retailer or with the vulnerable incumbent counterpart. Our result reveals that the supplier's strategy depends on subtle considerations of multiple factors such as terminal market demand state, the demand fluctuation, the expected market demand and the dominant retailer's wholesale price.

Keywords: Game theory · Information leakage · Information concealment · Dominant retailer

1 Introduction

With popularization and application of big data and the cloud, enterprises in supply chains are confronted with a complicated environment where unprecedented progress and enormous challenges co-exist. Competition between retailers for original resources is updated to the fight for information. More accurate information brings the advantage of significant profit potential and helps win competitiveness. From another front, rapidly growing dominance of large retailers has relocated the traditional power between supply and demand. The consumer demand-oriented market entitles the retailers more governance in supply chain structures. Despite that information dissemination in the presence of dominant retailers captures increasing attention, relative studies are still sparse. This motivates our study to fill in the research gap.

2 Problem Description

Following the above-mentioned reality, we consider a supply chain consisting of one common upstream *supplier* (he) and two differentiated downstream retailers.

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_30

One retailer is considered to be a dominant *entrant* retailer (she), while the other is a weak *incumbent* retailer (it). Specifically, there are two significant differences between retailers: first, the incumbent has exact acquaintance with the market demand due to his long-term immediate and continuous contact with consumers, whereas the entrant only knows the distribution of the demand information as her common knowledge (and so does the supplier), see [1,2]. Second, despite being a new comer in the supply chain, the entrant is endowed with the ability to dictate the wholesale price to the supplier, however, the incumbent's wholesale price is formulated by the supplier, see [3]. All participants are profit-maximizers.

Taking into consideration the factors influencing the terminal market, we reasonably model the market price to follow the inverse demand function [4], which is linear and downward sloping such that $P = A - B(q_i + q_e)$, where P represents the market's clearing price. The intercept A is assumed to be random, and takes two possible values of A_H or A_L ($A_H > A_L > 0$): the high type A_H occurs with probability $p \in (0, 1)$, and the low one A_L with the probability $1-p$. We define δ as the difference between the two of market demand realizations so that $\delta = A_H - A_L$; $-B$ is the demand function's slope. Without loss of generation, we assume $B = 1$. We denote q_i and q_e as the incumbent's and entrant's order quantities, respectively. Also, the expected market demand is defined as $\mu = pA_H + (1-p)A_L$. The supplier provides the incumbent a wholesale price w_i, whereas the entrant has the power to set the wholesale price w_e by herself. For simplicity, the entrant's wholesale price is assumed to be exogenous. We also normalize the marginal supply cost and other fixed costs to zero, which will not change our results qualitatively.

Initially, the accurate terminal market demand state is the incumbent's private information. The supplier merely has a sense of the demand's distribution and offers the incumbent a wholesale price contract. Once the incumbent places its order quantity, the private information about the actual market demand is faithfully leaked to the supplier. In presence of the two differentiated retailers, the supplier is confronted with collaboration choices about its cooperation partner. Specifically, the potential advantages when collaborating with the dominant entrant may prompt the supplier to choose "hug her close" policy. Under such condition, the supplier would leak the incumbent's demand state to the entrant voluntarily. Consequently, the terminal market demand information of this supply chain is completely transparent to all participants. From another front, the entrant's governance superiority on bargaining power may provide incentive for the supplier to stand in line with the weak incumbent, i.e., collaboration with the vulnerable incumbent retailer. Accordingly, supplier would conceal the actual market demand state to the entrant. Without information leaked from the supplier, the entrant has to place her order on the basis of her estimated market demand.

In the following, Sect. 3 analyzes the scenario of information leakage, where the supplier collaborates with the dominant entrant retailer. The alternative choice of cooperating with the vulnerable incumbent retailer is investigated in Sect. 4. Section 5 compares these two modalities of cooperation and derives all participants' equilibrium strategies. Section 6 gives the conclusion to close this paper.

3 Information Leakage

In this section, we derive equilibrium strategies for all the supply chain members under the information leakage scenario where the supplier chooses to collaborate with the dominant entrant retailer.

At the outset, the incumbent observes the actual terminal market demand information, which takes either the high type A_H or the low one A_L. The supplier provides the wholesale price contract w_{iH}^Y or w_{iL}^Y to the incumbent, where the superscript "Y" expresses the scenario that the supplier says "Yes" to information leakage. Afterwards, the incumbent responds by appointing its order quantity q_{iH}^Y or q_{iL}^Y. In this way, the supplier acquires the demand information from the incumbent, and then leaks to the entrant. Sequentially, the entrant places her order quantity q_{eH}^Y or q_{eL}^Y on account of this information. Finally, two retailers sell products to end consumers when demand uncertainty is resolved and market gets clear.

Note that the analysis processes are similarly formulated under the scenarios of high and low market uncertainty realizations, we thereafter elaborate on the circumstance when the incumbent observes a high market potential. Under such case, the supplier offers wholesale price w_{iH}^Y to the incumbent, and the incumbent places order quantity q_{iH}^Y with the supplier. Signing a deal with the entrant, the supplier may leak the actual demand information to its partner. As a result, the market demand state is no longer the incumbent's private information but rather a piece of transparent information to all participants. Afterwards, the entrant determines her order q_{eH}^Y according to this accurate information. Therefore, when the supplier's choice is collaboration with the dominant entrant retailer under that the terminal market demand state is high, the profit of the incumbent and the entrant is respectively given by

$$\pi_{iH}^Y = \left(A_H - q_{iH}^Y - q_{eH}^Y\right) q_{iH}^Y - w_{iH}^Y q_{iH}^Y, \tag{1}$$

$$\pi_{eH}^Y = \left(A_H - q_{iH}^Y - q_{eH}^Y\right) q_{eH}^Y - w_e q_{eH}^Y. \tag{2}$$

The Cournot competition between two retailers with complete information for the demand system is described as

$$\begin{cases} \max\limits_{q_{iH}^Y} & \pi_{iH}^Y \\ \max\limits_{q_{eH}^Y} & \pi_{eH}^Y. \end{cases} \tag{3}$$

Apart from providing products to both retailers, the supplier plays a dual role in the supply chain, that is, he serves as the wholesale price setter for the incumbent and also a information transmitter for the entrant. Therefore, his profit consists of two sources:

$$\pi_{sH}^Y = w_{iH}^Y q_{iH}^Y + w_e q_{eH}^Y. \tag{4}$$

On account of the sequential moves between the supplier and incumbent, they play a standard Stackelberg game with complete information, where the supplier is the leader and the incumbent is the follower.

The following proposition demonstrates the unique pure strategy equilibrium and the corresponding participants' profits for the scenario of collaboration with the dominant entrant retailer.

Proposition 1. *When the market demand state is high and the supplier chooses to cooperate with the dominant entrant:*

(i) *The supplier offers the incumbent the wholesale price* $w_{iH}^{*Y} = \frac{A_H + 2w_e}{4}$.

(ii) *The incumbent orders* $q_{iH}^{*Y} = \frac{A_H}{6}$. *The incumbent's corresponding profit is* $\pi_{iH}^{Y} = \left(\frac{A_H}{6}\right)^2$.

(iii) *The entrant orders* $q_{eH}^{*Y} = \frac{5A_H - 6w_e}{12}$. *The entrant's corresponding profit is* $\pi_{eH}^{Y} = \left(\frac{5A_H - 6w_e}{12}\right)^2$.

4 Information Concealment

In this section, we look for equilibrium strategies under the information concealment scenario where the supplier chooses to collaborate with the vulnerable incumbent retailer.

Unlike collaboration with the dominant entrant retailer, when the supplier chooses to align with the weak incumbent, he should conceal the incumbent's private information, i.e., the terminal market demand state from the entrant. Thus, the entrant only places her order quantity according to her expected market demand.

At first, the incumbent observes the actual terminal market demand information, either to be high or low. The supplier provides the wholesale price contract w_{iH}^N or w_{iL}^N to the incumbent, where the superscript "N" expresses the scenario that the supplier says "No" to information leakage. Afterwards, the incumbent responds by appointing its order quantity q_{iH}^N or q_{iL}^N. In this way, the supplier acquires the demand information from the incumbent, and keep this information secret. Sequentially, the entrant places her order quantity q_{eH}^N or q_{eL}^N on account of the expected market demand. Finally, both retailers sell their products to the terminal market where demand uncertainty is realized.

We follow the similar assumption as in Sect. 3 that the demand uncertainty is realized to be high. Therefore, when the supplier's choice is to collaborate with the vulnerable incumbent retailer and the terminal market demand state is high, the incumbent and entrant seek to maximize the respective profit

$$\max_{q_{iH}^N} \quad \pi_{iH}^N = \left(A_H - q_{iH}^N - q_e^N\right) q_{iH}^N - w_{iH}^N q_{iH}^N, \tag{5}$$

$$\max_{q_e^N} \quad \pi_e^N = \left[p\left(A_H - q_{iH}^N - q_e^N\right) + (1-p)\left(A_L - q_{iL}^N - q_e^N\right)\right] q_e^N - w_e q_e^N. \tag{6}$$

Different from the analysis in Sect. 3, the supplier acts as a price setter and a secret keeper for the incumbent in the no leakage case. His profit is thus formulated as

$$\pi_{sH}^{N} = w_{iH}^{N} q_{iH}^{N} + w_e q_e^{N}. \tag{7}$$

The following proposition demonstrates the unique pure strategy equilibrium and the corresponding participants' profits for the scenario that collaboration with the dominant entrant retailer.

Proposition 2. *When the market demand state is high and the supplier chooses to cooperate with the vulnerable incumbent:*

(i) *The supplier offers the incumbent the wholesale price*

$$\begin{aligned} w_{iH}^{*N} = \big[&(48A_H - 20A_L + 24w_e) - (24A_H - 21A_L - 30w_e)p - (A_H + 6w_e)p^2 \\ &+ (A_H - A_L)p^3\big]/6(16 + p - p^2). \end{aligned}$$

(ii) *The incumbent orders*

$$q_{iH}^{*N} = \frac{(12A_H - 5A_L + 6w_e) - (3A_H - 4A_L + 3w_e)p - (A_H - A_L + 3w_e)p^2}{3(16 + p - p^2)}.$$

The incumbent's corresponding profit is

$$\pi_{iH}^{N} = \left[\frac{(12A_H - 5A_L + 6w_e) - (3A_H - 4A_L + 3w_e)p - (A_H - A_L + 3w_e)p^2}{3(16 + p - p^2)}\right]^2.$$

(iii) *The entrant orders*

$$\begin{aligned} q_e^{*N} = \big[&(40A_L - 48w_e) + (42A_H - 37A_L - 18w_e)p - (A_H + 4A_L - 18w_e)p^2 \\ &- (A_H - A_L)p^3\big]/6(16 + p - p^2). \end{aligned}$$

The entrant's corresponding profit is

$$\begin{aligned} \pi_e^{N} = \big[&(40A_L - 48w_e) + (42A_H - 37A_L - 18w_e)p - (A_H + 4A_L - 18w_e)p^2 \\ &- (A_H - A_L)p^3\big]^2/36(16 + p - p^2)^2. \end{aligned}$$

5 The Supplier's Equilibrium Strategy

This section, we calculate the supplier's profits under the above two mechanisms and show the supplier's equilibrium strategy.

Corollary 1. *Comparing the supplier' profit under the collaboration scenario with the vulnerable incumbent retailer and that with the dominant entrant retailer, there is a threshold δ_H such that*

(a) *if $\delta < \delta_H$, then $\pi_{sH}^{N} < \pi_{sH}^{Y}$;*
(b) *if $\delta \geq \delta_H$, then $\pi_{sH}^{N} \geq \pi_{sH}^{Y}$;*

Corollary 1 reveals that the supplier's cooperation choice strategically alters due to different terminal market demand state. Specifically, when the market is prosperous, the information is less valuable so that the supplier prefers to collaborate with the entrant to gain greater revenue. Conversely, when the market tolerates great fluctuations, the actual demand information becomes much valuable, and hence the supplier is willing to align with the incumbent to keep the information away from the entrant.

6 Conclusions

With the tremendous development of information technology and widespread application of big data, information dissemination has attracted attention from a growing number of experts and scholars. When the dominant retailer takes over the leadership of setting wholesale price, the supplier loses his governance in the supply chain management and needs to respond strategically in terms of choosing its supply chain partner. Our work contributes to give such managerial hints to the supplier. If the market demand is upbeat, to work with the dominant retailer could bring the supplier more profit. On the contrary, when the terminal market demand is fluctuating greatly, the supplier is suggested to huddle together with the vulnerable retailer for warmth.

References

1. Anand, K.S., Goyal, M.: Strategic information management under leakage in a supply chain. Manag. Sci. **55**(3), 438–452 (2009)
2. Kong, G., Rajagopalan, S., Zhang, H.: Revenue sharing and information leakage in a supply chain. Manag. Sci. **59**(3), 556–572 (2013)
3. Geylani, T., Dukes, A.J., Srinivasan, K.: Strategic manufacturer response to a dominant retailer. Mark. Sci. **26**(2), 164–178 (2007)
4. Liao, C.N., Chen, Y.J., Tang, C.S.: Heterogeneous farmers and market selection. Manuf. Serv. Oper. Manag. (2017, in press)

Optimization Allocation Between Multiple Logistic Tasks and Logistic Resources Considered Demand Uncertainty

Xiaofeng Xu(✉) and Jing Liu

China University of Petroleum, Qingdao 266580, Shandong, China
xuxiaofeng@upc.edu.cn

Abstract. Making an allocation scheme which can achieve the optimal overall efficiency that matching multiple logistics tasks and resources under the environment that the tasks' demands are uncertain is difficult. In this paper, we build a mathematical model to describe the problem and try to solve it by the genetic algorithm. We also consider the daily usage amount of each resource should be as equilibrious as possible. The result of the case simulation proves the effectiveness of the model and the algorithm. As well as, we analyze the impact that the size of the uncertainty's degree on the allocation result.

Keywords: Resource allocation · Uncertainty · Resource equalization · Genetic algorithm

1 Introduction

In the collaborative logistic network, TRA (Task-Resource Assignment) is an important work content. Gattorna and Jones studied the allocation problem of 1-1 TRA between independent tasks and alternative resources in the supply chain [1]. Sung studied a 1-N TRA problem that multiple hospitals in the region admit the victims [2]. Yi and Ozdamar proposed a model related to a N-1 TRA problem which treats vehicles as integer commodity flows rather than binary variables [3]. Zhou et al. established an optimal allocation model, which is a N-N TRA problem involving multi-start, multi-end and multi-stage [4].

The research about uncertainty in the resources allocation is one of the major concerns. Santoso and Lee studied the impact that uncertain parameters on network planning of the supply chain standing in the strategic level [5, 6]. Hu and Liu thought about the demand from the affected areas is uncertain and the cost of allocating the relief resources is also imprecise in the bi-objective robust optimization model they proposed [7]. Liu et al. took the logistics service supply chain which has three echelons as the case to study, considering the demand is uncertain [8]. Xu et al. considered the uncertainty of time to establish a bi-level programming model with random constraints [9].

Integrating the documents, combining the N-N TRA with the uncertainty becomes a hot study object. When solving planning scheme, the genetic algorithm is used widely, such [10].

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_31

2 Problem Description

In the logistics service supply chain of the collaborative logistic network, the participators include logistics service integrator, clients and logistics resource providers. In the system, the types of tasks include procurement, transportation, warehousing, packaging and so on. So the logistics resource providers include suppliers, transport companies, warehouses, packhouses and so on. The demand of each task is uncertain and each task can select one or more resources. While, each resource can serve for one or more tasks basing on its daily capacity. During the operational process, multiple tasks run in parallel and form relations of divergent selection with multiple resources. The relations are shown as Fig. 1. The following questions need to be solved: (1) selecting resources: the whole tasks need to select the most suitable resources considering the execution time, quality of service, and cost of each resource. (2) selecting tasks: the whole resources need to select the most suitable tasks considering the daily capacity of themselves, the begin time and demand of each task.

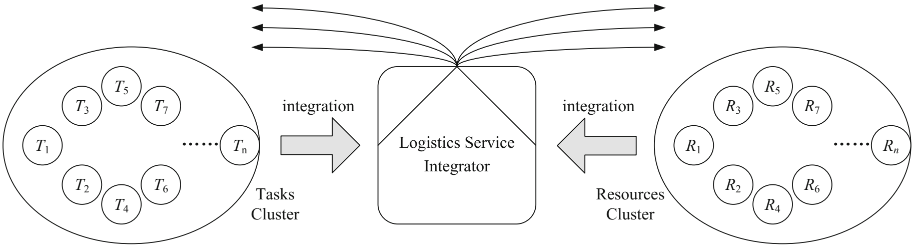


Fig. 1. The relations of divergent selection between tasks and resources

To simplify the model and keep its generality, the assumptions are made as follows:

(1) These logistics tasks are independent of each other;
(2) Each resource provides each task service for just one time when they receive the task request. During the period that one resource serve one task, its daily usage amount is the service amount the resource serve the task;
(3) Each task involves one kind of goods and different tasks could involve different kinds of goods;
(4) The service capacity of each resource is fixed during one cycle.

3 Model Building

3.1 Mathematical Description

The types of tasks k, the begin times of tasks t_{ki}^{s} and the kinds of goods p are not all the same. The amount of tasks' types is K and goods' kinds is P. The amount of tasks belonging to k type is n_k. The goods' kind of the task i belonging to k type is p_{ki} and the task's demand is d_{ki}, which obeys the normal distribution $d_{ki} \sim N(\mu_{ki}, \sigma_{ki}^{2})$. The total demand of tasks belonging to k type is D_{kp}. The types of resources is k' and the amount of resources' types is K$'$. If the numbers belonging to k and k' are same, the types of the task and the resource are the same. The amount of resources belonging to k' type is $N_{k'}$. The service capacity of the resource j belonging to k' type is $s_{k'jp}$ and $\sum_{j=1}^{N_{k'}} s_{k'jp} \geq D_{kp}(k = k')$. The service amount that the resource j belonging to k' type provide for the task i belonging to k type $(k = k')$ is $x_{ki}^{k'j}$, the unit cost is $c_{ki}^{k'j}$, the evaluation value for the quality of service is $q_{ki}^{k'j}$ and the execution time is $t_{ki}^{k'j}$. The unit property for the goods belonging to kind p taking up the resource belonging to k' type is $\gamma_{k'p}$. For example, the $\gamma_{k'p}$ to the transport company is truck/one goods, the $\gamma_{k'p}$ to the warehouse is m^3/one goods. When $k = k' = 1$, the type is transportation and the distances between transport company j and its task i are l_{1i}^{1j}. While, the trucks' average speed of this transport company is v^{1j}. Considering the uncertainty of demand, the chance constraint is made as $Prob\left(\sum_{j=1}^{N_{k'}} x_{ki}^{k'j} \geq d_{ki}\right) = \alpha$, where α shows the confidence level. For example, if $\alpha = 0.95$, it means the probability of completing the task is greater than 95%.

If the kinds of goods p_{ki} are same, all these tasks can be classified into a cluster G.

3.2 Mathematical Model

$$\min C = \sum_{k=1}^{\mathrm{K}} \sum_{i=1}^{n_k} \sum_{j=1}^{N_{k'}} \left(x_{ki}^{k'j} \cdot c_{ki}^{k'j}\right) \tag{1}$$

$$\min T = \sum_{k=1}^{\mathrm{K}} \sum_{i=1}^{n_k} t_{ki} \tag{2}$$

$$\min Q = \sum_{k=1}^{\mathrm{K}} \sum_{i=1}^{n_k} \sum_{j=1}^{N_{k'}} \left(x_{ki}^{k'j} \cdot q_{ki}^{k'j}\right) \tag{3}$$

$$\min B = \sum_{k'=1}^{\mathrm{K}'} \sum_{j=1}^{N_{k'}} \sum_{p=1}^{p=\mathrm{P}} \left(\sum_{t=t_{k'j}^{sp}}^{t=t_{k'j}^{fp}} \left(\sum_{i \in G} x_{k'ji}^{tp} \cdot \gamma_{k'p} - \bar{R}_{k'jp} \right) \right) \tag{4}$$

$$t_{ki} = \max(t_{ki1}, t_{ki2}, \cdots, t_{kiN_{k'}}) \tag{5}$$

$$c_{kij} \leq c_{ki}^{m}, k = 1, 2, \cdots, K; i = 1, 2, \cdots, n_k; j = 1, 2, \cdots N_{k'} \tag{6}$$

$$t_{kij} \leq t_{ki}^{m}, k = 1, 2, \cdots, K; i = 1, 2, \cdots, n_k; j = 1, 2, \cdots N_{k'} \tag{7}$$

$$q_{kij} \leq q_{ki}^{m}, k = 1, 2, \cdots, K; i = 1, 2, \cdots, n_k; j = 1, 2, \cdots N_{k'} \tag{8}$$

$$\sum_{i \in G} x_{kij}^{t} \cdot \gamma_{k'p} \leq s_{k'jp}, k = 1, 2, \cdots, K; i = 1, 2, \cdots, n_k; j = 1, 2, \cdots N_{k'} \tag{9}$$

$$Prob\left(\sum_{j=1}^{N_{k'}} x_{kij} \geq d_{ki}\right) = \alpha, k = 1, 2, \cdots, K; i = 1, 2, \cdots, n_k \tag{10}$$

$$x_{kij} \geq 0, c_{kij} \geq 0, q_{kij} \geq 0, k = 1, 2, \cdots, K; i = 1, 2, \cdots, n_k; j = 1, 2, \cdots N_{k'} \tag{11}$$

(1) indicates that the total execution cost is lowest. (2) shows the total execution time is shortest. (3) indicates the overall quality of service is best, where the evaluation value is smaller, the result is better. (4) means the daily usage amounts of all the resources should be most equilibrious. The daily service amount that the resource j belonging to k' type provides for the goods belonging to p kind is $\sum_{i \in P} x_{k'ji}^{tp} \cdot \gamma_{k'p}$. The daily average service amount that the resource j belonging to k' type provides for the goods belonging to p kind is $\overline{R_{k'jp}}$. (5) means that the execution time of completing a task is represented by the longest execution time among all times executed by each resource. (6), (7) and (8) ensure that the execution cost, execution time and service quality to complete each task cannot exceed their limits. (9) illustrates that the daily usage amount of each resource cannot exceed theirs daily service capacity. (10) is a chance constraint, which indicates the probability of completing each task is greater than a certain value. (11) requires that decision variables and other parameters are nonnegative.

4 Model Algorithm

(a) Considering $d_{ki} \sim N(\mu_{ki}, \sigma_{ki}^2)$, The constraint (10) can be transformed into the Eq. (12).

$$\sum_{j=1}^{N_{k'}} x_{ki}^{k'j} = \mu_{ki} + \phi^{-1}(\alpha)\sigma_{ki} \tag{12}$$

(b) Code design

Each individual in the chromosome string corresponds to the decision variables, the length of chromosome equals to number of task types multiply the number of tasks belonging to each type.

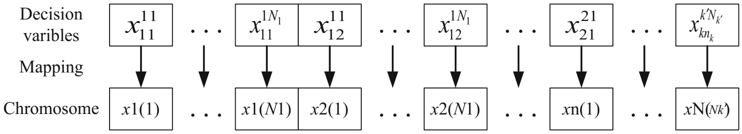


Fig. 2. Code structure

(c) Fitness function and initial population

In this paper, there are four objective functions, which need to be transformed into single target to be solved. The fitness function of this paper is as follows:

$$f = \lambda_1 \cdot \sum_{k=1}^{\mathrm{K}} \sum_{i=1}^{n_k} \sum_{j=1}^{N_{k'}} \left(x_{ki}^{k'j} \cdot c_{ki}^{k'j} \right) / c_{\max} + \lambda_2 \cdot \sum_{k=1}^{\mathrm{K}} \sum_{i=1}^{n_k} t_{ki} / t_{\max} + \lambda_3 \cdot \sum_{k=1}^{\mathrm{K}} \sum_{i=1}^{n_k} \sum_{j=1}^{N_{k'}} \left(x_{ki}^{k'j} \cdot q_{ki}^{k'j} \right) / q_{\max}$$

$$+ \lambda_4 \cdot \sum_{k'=1}^{\mathrm{K}'} \sum_{j=1}^{N_{k'}} \sum_{p=1}^{p=\mathrm{P}} \left(\sum_{t=t_{k'j}^{sp}}^{t=t_{k'j}^{fp}} \left(\sum_{i \in P} x_{k'ji}^{tp} \cdot \gamma_{k'p} - \bar{R}_{k'jp} \right) \right) / b_{\max} \tag{13}$$

(d) Selection, crossover and mutation

In this paper, the roulette selection is selected to choose individuals. pi is the selection probability, and the formula is as follows:

$$p_i = fitness(x_i) / sum(fitness)$$

The cross probability is usually in the range of [0.4, 0.99]. In the crossover process, a random number between [0, 1] will be generated for each individual in the population. When the crossover probability is not less than the random number, the individual will be made as a parent individual.

Similar to the crossover operation, a random number between [0, 1] is generated for each individual in the population. When the mutation probability is not less than the random number, the individual will be mutated.

(e) Termination

In this paper, the iteration of the genetic algorithm is set. When reaching the iteration, the operation is terminated.

5 Case Simulation

5.1 Case Description

There will be seven separate logistics tasks from March 11th to March 13th. The specific information about tasks, resources, distances and goods are shown in Tables 1, 2, 3 and 4. Seven logistics resources suppliers could be selected by the logistics service integrator to match tasks in order to attain the lowest cost, the shortest time and the best service quality of the entire system. At the same time, the matching scheme should ensure that the daily usage amount of the whole resources is as equilibrious as possible.

5.2 Case Solving

After setting all parameters, the corresponding genetic algorithm program is designed. Then, use Matlab2014 to solve the case. The parameters in the genetic algorithm are set as follows: Population size is 50; the number of iterations is 100; the crossover probability is 0.6 and the mutation probability is 0.01. The four weight coefficients in the objective function are 0.3, 0.2, 0.2 and 0.3.

Table 1. Task information.

Tasks	Goods' kind	Demand	Confidence level (α)	Start time	Time limit	Cost limit	Quality limit
t_{11}	p_1	N(2000, 16)	0.9	3.11	12	5	4
t_{12}	p_2	N(800, 16)	0.9	3.12	10	4	4
t_{21}	p_3	N(1000, 16)	0.9	3.11	1	3	4
t_{22}	p_4	N(1500, 16)	0.9	3.11	2	3	3
t_{23}	p_4	N(800, 16)	0.9	3.12	2	2	5
t_{31}	p_5	N(1500, 16)	0.9	3.11	4	1	7
t_{32}	p_6	N(1200, 16)	0.9	3.13	3	1	6

Table 2. Resource information

Resources	Tasks	Unit cost	Execution time	Service quality	Average speed	Service capacity	
y1	t_{11}	5	/	4	5	Number of available trucks	15
	t_{12}	4	/	4			
y2	t_{11}	5	/	4	4		10
	t_{12}	4	/	3			
y3	t_{11}	4	/	4	5		15
	t_{12}	4	/	4			
g1	t_{21}	2	1	5	/	Number of available goods	1200
	t_{22}	2	2	3			1500
	t_{23}	2	2	5			
g2	t_{21}	3	1	4	/		1100
	t_{22}	3	2	3			1200
	t_{23}	2	1	3			
c1	t_{31}	1	/	7	/	Number of available volume	500
	t_{32}	1	/	6			
c2	t_{31}	1	/	6	/		200
	t_{32}	2	/	6			

Table 3. Distances between tasks and resources

Tasks	Distances by y1	Distances by y2	Distances by y3
t_{11}	50	35	70
t_{12}	40	65	40

Table 4. Goods Information

Goods' kind	Truck/one goods	Volume/one goods
p_1	0.01	/
p_2	0.02	/
p_5	/	0.2
p_6	/	0.3

Run the program on Matlab2014 and get the simulation results shown in Fig. 2. The resulting chromosomes are shown as follows:

[1147.44, 857.7, 0, 59.15, 0, 745.99, 0, 1005.14, 398.27, 1106.87, 751.89, 53.25, 592.29, 912.85, 1205.14, 0].

6 Conclusion

Based on the logistics service integrator's perspective, aiming at the overall optimal efficiency of the executions of all tasks, and considering the usage balance of logistics resources, this paper designs mathematical model. The model involves a variety of logistics tasks and goods. A single task can be executed by multiple resources, and a single resource can also serve a number of tasks. The logistics service integrator makes decisions based on the corresponding resource set, making the overall efficiency the best. The results of the case simulation verify the validity of the model and the algorithm. Through the further analysis of the example, it is concluded that the greater the uncertainty of the task demand, the more difficult the decision results reach the overall optimal efficiency.

Acknowledgements. This research was supported by Shandong Provincial Natural Science Foundation, China (Grant No. ZR2015GQ006), and the Fundamental Research Funds for the Central Universities, China (Grant No. 17CX04023B).

References

1. Gattorna, J., Jones, T.: Strategic Supply Chain Alignment: Best Practice in Supply Chain Management. Gower Publishing, UK (1998). **170**(5), 325–329
2. Sung, I.: Optimal allocation of emergency medical resources in a mass casualty incident: patient prioritization by column generation. Eur. J. Oper. Res. **252**(2), 623–634 (2016)
3. Yi, W., Ozdamar, L.: A dynamic logistics coordination model for evacuation and support in disaster response activities. Eur. J. Oper. Res. **179**(3), 1177–1193 (2007)
4. Zhou, L., Wang, X., Lin, Y., Jing, Y.: Integrated multi-task scheduling for spatially distributed small-batch logistics. Comput. Integr. Manuf. Syst. **22**(3), 822–832 (2016)
5. Santoso, T., Ahmed, S., Goetschalckx, M., et al.: A stochastic programming approach for supply chain network design under uncertainty. Eur. J. Oper. Res. **167**(1), 96–115 (2005)
6. Lee, D.-H., Dong, M.: Dynamic network design for reverse logistics operations under uncertainty. Transp. Res. Part E: Logist. Transp. Rev. **45**(1), 61–71 (2009)
7. Hu, C.L., Liu, X.: A bi-objective robust model for emergency resource allocation under uncertainty. Int. J. Prod. Res. **54**(24), 7421–7438 (2016)
8. Liu, W., Qu, S., Zhong, S.: Order allocation in three-echelon logistics service supply chain under stochastic environments. Comput. Integr. Manuf. Syst. **18**(2), 381–388 (2012)
9. Xu, X., Chang, W., Liu, J.: Resource allocation optimization model of collaborative logistics network based on bi-level programming. Sci. Program. (2017)
10. Li, X., Lo, H.K.: An energy-efficient scheduling and speed control approach for metro rail operations. Transp. Res. Part B: Methodol. **64**, 73–89 (2014)

Two-Stage Heuristic Algorithm for a New Model of Hazardous Material Multi-depot Vehicle Routing Problem

Wenyan Yuan[1], Jian Wang[1], Jian Li[2], Bailu Yan[1], and Jun Wu[3](✉)

[1] School of Science, Beijing University of Chemical Technology,
Beijing 100029, China
[2] Research Base of Beijing Modern Manufacturing Development,
College of Economics and Management, Beijing University of Technology,
Beijing 100124, China
[3] School of Economics and Management,
Beijing University of Chemical Technology, Beijing 100029, China
wujun@mail.buct.edu.cn

Abstract. Vehicle routing problem (VRP) plays a vital role in logistics management. Among which, the transportation of hazardous material attracts much attention especially in China. The hazardous material multi-depot vehicle routing problem (HMDVRP) considers the transportation of hazardous material and multiple depots based on VRP. This paper develops a new HMDVRP bi-objective optimization model. Some new decision variables are introduced to the model to describe the sequence of customers and simplify the model expression. Moreover, the risk measurement of the model considers the change of the loading, which reflects the nature of hazardous material transportation. HMD-VRP is NP-hard, and the heuristic algorithms are the main method used for solving it. This paper proposes a two-stage heuristic algorithm to solve the new HMDVRP model. Numerical experiments show that the two-stage heuristic algorithm can solve the HMDVRP model effectively and efficiently.

Keywords: Hazardous material transportation · Multi-depot vehicle routing problem · Bi-objective optimization · Heuristic algorithm

1 Introduction

Hazardous materials, which defined by The Pipeline and Hazardous Materials Safety Administration of the U.S. Department of Transportation, may be flammable, explosive, toxic, radioactive, corrosive, or may have other characteristics that can harm people, facilities, environment or other living organisms. Hazardous materials are extremely dangerous, and most of them need to be transported to a specific place. During transportation, once an accident occurs, the damage may be huge. So it is very important to study the transportation problem and risk assessment of hazardous materials.

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7_32

Vehicle routing problem (VRP) is one of the most important combinatorial optimization problems for the research of real world application, and it is also one of the important contents for the research on modern logistics management. It was first proposed by Dantzig and Ramser [1], they described a route optimization problem that transports the hazardous materials gasoline from depot to each service station, and the first mathematical programming model and algorithm for VRP are also proposed in that article. Over the next several decades, the problem of VRP is extensively expanded and developed, Toth and Vigo [2] made an excellent review on the methods and applications of VRP in 2014. VRP has been classified into many different variants such as Capacitated VRP (CVRP) [3], Periodic VRP (PVRP) [4], VRP with time windows (VRPTW) [5], Dynamic VRP (DVRP) [6], Split Delivery VRP (SDVRP) [7], etc.

All the literature mentioned above consider only one depot. However, there exists a situation which is more close to real life called Multiple Depots VRP (MDVRP). Crevier et al. [8] considered the MDVRP problem in which the distributions of goods is done from several depots to customers. Research related to the MDVRP appears first in the literature by Kulkarni and Bhave [9]. Since then a lot of scholars have studied the problem of MDVRP and its variants. An overview of academic works was proposed by Liu et al. [10]. Compared with the single-depot VRP, MDVRP is more sophisticated and challenging. If customers are evidently gathered around each depot, the MDVRP can be split into multiple single-depot VRPs. Otherwise customers will be served from any of the depots, and therefore a multi-depot-based approach has to be used.

In hazardous materials transportation context, MDVRP is extended to the Hazardous materials Multiple Depots Vehicle Routing Problem (HMDVRP), and objective of minimizing risk is added to the objective function in addition to the conventional objective of minimizing the cost. Such multi-objective nature further increases the complexity of the HMDVRP. HMDVRP is a very important problem in real life. However, as far as we know, it is seldom found in literature, which is the motivation of our paper. The key difference between MDVRP and HMDVRP is that HMDVRP stresses risk issues, so it is very important to measure the risk in the process of transporting the hazardous materials. Pradhananga et al. [11,12] considered the hazardous materials VRP and used the loading of the hazardous materials and the size of exposed population on the route to measure the risk. Because of the low probability and high risk characteristics of hazardous materials transportation accident, Erkut and Ingolfsson [13] proposed three risk measurement models based on big disaster circumvention, which were maximum number of exposed population, minimum expected loss variance and minimum expected effect on transport routes, to measure the risk. Kang et al. [14] used Value at Risk (VaR) to assess the risk of hazardous materials transportation, they educed that the route selection is relevant to the risk tolerance function of decision maker. Moreover, they controlled VaR value in the required confidence level and highly dispersed the risk at the same time. Revelle et al. [15] combined the exposure population with vehicle loading of hazardous materials to measure the transportation risk.

Although existing research have considered the influence of loading on risk in the transport of hazardous materials, the loading is only treated as a constant in the process of analysis and evaluation. However, unloading during the actual process of hazardous materials transportation happens sometimes. This means that the vehicle loading is not a constant one but dynamic. However, previous literature has not taken it into account. In this paper, we proposed a risk measurement method that considers the dynamic vehicle loading and proposed a new bi-objective optimization model for HMDVRP. This new model reduces the number of decision variables and constraints, which makes the model representation more concise. Meanwhile, the expression of the decision variable in the new model is intuitive, which is helpful for the decision maker to make the decision quickly. The comparison of different risk measurement in our paper shows that the risk measurement proposed in this paper reflects the transportation risk of HMDVRP more accurately, and identifies the optimal route more effectively.

2 Transportation Model

In the following, mathematical model to the HMDVRP is given:

$$\min Z_1 = \sum_{(i,j)\in A} \sum_n \sum_k \sum_m sg_m^{nkr_j^{nk}} \cdot sn_{ij}^{nk} \cdot p_{ij} \cdot \rho_{ij} \cdot d_m / cap\_v + \sum_n \sum_k R_1^{nk} \tag{1}$$

$$\min Z_2 = \sum_n \sum_k \left(\sum_{(i,j)\in A} sn_{ij}^{nk} \cdot c_{ij} + C_1^{nk} + C_{last}^{nk} \right) \tag{2}$$

$$\sum_m sign(r_m^{nk}) \cdot d_m \le cap\_v \qquad n = 1, \ldots, N; k = 1, ..., v_n \tag{3}$$

$$\sum_k \sum_m sign(r_m^{nk}) \cdot d_m \le cap\_d \qquad n = 1, \ldots, N \tag{4}$$

$$\sum_n \sum_k sign(r_m^{nk}) = 1 \qquad m = 1, \ldots, M \tag{5}$$

Equations (1) and (2) are objective functions for minimizing the total risk and total cost of the transportation process, respectively. Constraint (3) and (4) indicate that the total demand of each route and each depot should not exceed the vehicle capacity and depot capacity respectively, and Eq. (5) enforces each customer to be serviced by a unique vehicle and arc.

In the model, if we remove the formula (1) and (4), it is the same problem with Kulkarni and Bhave [9]. For the same size of a problem (N depots, M customers and V vehicles), the number of decision variables in the model of Kulkarni and Bhave are (M+N)(M+N)V, and the number of constraints are 2M+V(M2+N+4); however, the number of decision variables and constraints in the model of this paper are MV and V+M respectively, both of them are much less than theirs, which makes the description of the model more concise.

3 Numerical Experiments

In this section, a case study is carried out to compare the algorithm with random generation for classification and the algorithm with the MDIHA for classification. The performance of the algorithms is evaluated using a randomly generated example: 50-customer HMDVRP, in which there are two depots available. Two stage heuristic algorithm to solve the HMDVRP model is used.

Figure 1 shows the information of the non-dominated solutions obtained by the algorithm with the two classification methods. From Fig. 1 we can see that the algorithm with MDIHCA can obtain better non-dominated solutions which have smaller risk and cost than the one with random generation for classification, and Pareto frontier clearly gives the optimal risk under different transportation costs, which can help decision makers make decisions and develop emergency measures quickly.

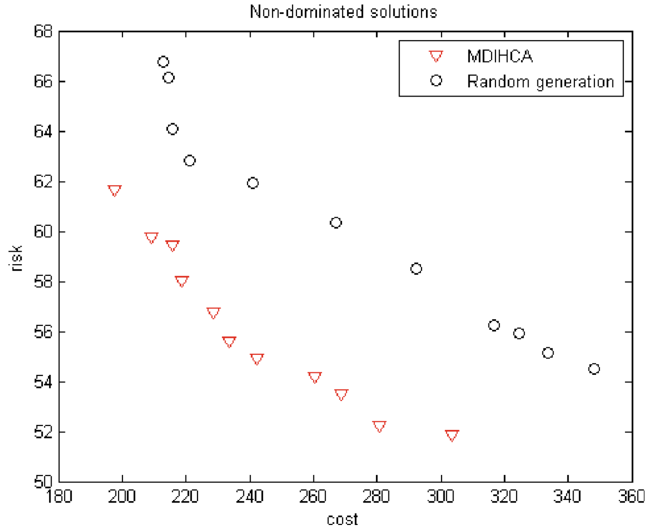


Fig. 1. Information of the non-dominated solutions

4 Conclusion

The VRP problem is of great importance both in theory and practice in transportation and logistics areas. In this paper, a two-stage heuristic algorithm was proposed to solve the HMDVRP model. Numerical experiments show that the proposed two-stage heuristic algorithm is an efficient algorithm and can solve the new model effectively.

Acknowledgments. This research was supported by the National Natural Science Foundation of China under Grant numbers (71171011, 71571010, 71372195), BUCT Fund for (JD1724).

References

1. Dantzig, G.B., Ramser, J.M.: The truck dispatching problem. Manag. Sci. **6**(1), 80–91 (1959)
2. Toth, P., Vigo, D.: Vehicle Routing: Problems, Methods, and Applications. SIAM, Philadelphia (2014)
3. Baldacci, R., Toth, P., Vigo, D.: Exact algorithms for routing problems under vehicle capacity constraints. Ann. Oper. Res. **175**, 213–245 (2010)
4. Mourgaya, M., Vanderbeck, F.: The periodic vehicle routing problem: classification and heuristic for tactical planning. RAIRO - Oper. Res. **40**, 169–194 (2006)
5. Qureshi, A.G., Taniguchi, E., Yamada, T.: An analysis of exact VRPTW solutions on ITS data-based logistics instances. Int. J. Intell. Transp. Syst. Res. **10**(1), 34–46 (2012)
6. Psaraftis, H.: Dynamic vehicle routing: status and prospects. Ann. Oper. Res. **61**(1), 143–164 (1995)
7. Archetti, C., Speranza, M.G.: The split delivery vehicle routing problem: a survey. In: The Vehicle Routing Problem: Latest Advances and New Challenges. Operations Research/Computer Science Interfaces, Part I, vol. 43, pp. 103–122 (2008)
8. Crevier, B., Cordeau, J.F., Laporte, G.: The multi-depot vehicle routing problem with inter-depot routes. Eur. J. Oper. Res. **176**(2), 756–773 (2007)
9. Kulkarni, R.V., Bhave, P.R.: Integer programming formulations of vehicle routing problems. Eur. J. Oper. Res. **20**(1), 58–67 (1985)
10. Liu, T., Jiang, Z., Liu, R., Liu, S.: A review of the multi-depot vehicle routing problem. Energy Procedia **13**, 3381–3389 (2011). Proceedings of ESEP 2011 Conference, Singapore, 9–10 December 2011
11. Pradhananga, R., Taniguchi, E., Yamada, T.: Ant colony system based routing and scheduling forhazardous material transportation. Procedia-Soc. Behav. Sci. **2**(3), 6097–6108 (2010)
12. Pradhananga, R., Taniguchi, E., Yamada, T., Qureshi, A.G.: Bi-objective decision support system for routing and scheduling of hazardous materials. Soc.-Econ. Plan. Sci. **48**(2), 135–148 (2014)
13. Erkut, E., Ingolfsson, A.: Catastrophe avoidance models for hazardous materials route planning. Transp. Sci. **34**(2), 165–179 (2000)
14. Kang, Y., Batta, R., Kwon, C.: Value-at-risk model for hazardous material transportation. Ann. Oper. Res. **222**(1), 361–387 (2014)
15. Revelle, C., Cohon, J., Shobrys, D.: Simultaneous siting and routing in the disposal of hazardous wastes. Transp. Sci. **25**, 262–271 (1991)

Author Index

F. Chao et al. (eds.), *Advances in Computational Intelligence Systems*,
Advances in Intelligent Systems and Computing 650, DOI 10.1007/978-3-319-66939-7