Xiaochun Yang
Chang-Dong Wang
Md. Saiful Islam
Zheng Zhang (Eds.)

LNAI 12447

Advanced Data Mining and Applications

16th International Conference, ADMA 2020
Foshan, China, November 12–14, 2020
Proceedings

ADMA
2020

Springer

MOREMEDIA

Lecture Notes in Artificial Intelligence 12447

Subseries of Lecture Notes in Computer Science

More information about this subseries at http://www.springer.com/series/1244

Xiaochun Yang · Chang-Dong Wang ·
Md. Saiful Islam · Zheng Zhang (Eds.)

Advanced Data Mining and Applications

16th International Conference, ADMA 2020
Foshan, China, November 12–14, 2020
Proceedings

Springer

Editors

Xiaochun Yang
Northeastern University
Shenyang, China

Md. Saiful Islam
Griffith University
Southport, QLD, Australia

Chang-Dong Wang
School of Data and Computer Science
Guangzhou, China

Zheng Zhang
School of Computer Science
and Technology
Shenzhen, China

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-3-030-65389-7 ISBN 978-3-030-65390-3 (eBook)
https://doi.org/10.1007/978-3-030-65390-3

LNCS Sublibrary: SL7 – Artificial Intelligence

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

Preface

On November 12–14, 2020, the 16th International Conference on Advanced Data Mining and Applications (ADMA 2020) was held in Foshan, China. Researchers and practitioners, ranging from young to experienced, from around the world came to the leading international forum to share innovative ideas, original research findings, case study results, and experienced-based insights into Advanced Data Mining and Applications. With its ever-growing importance in these data-rich times, ADMA has become a flagship conference in this field.

ADMA 2020 received 96 submissions in total. After a rigorous review, 35 regular papers and 14 short research papers (8 or more pages) were selected for presentation at the conference and publications in the proceeding. This corresponds to a full paper acceptance rate of 36.5%. The Program Committee (PC) members composed of international experts in relevant fields did a thorough and professional job of reviewing the papers submitted to ADMA 2020. Each paper went through a review process receiving at least three reviews provided by the PC. With the growing importance of data in this digital age, papers accepted in ADMA 2020 could cover a wide range of research topics in the fields of data mining, including machine learning, text mining, graph mining, predictive data analytics, recommender systems, query processing, analytics-based applications, privacy, and security analytics. The ADMA 2020 conference program was complemented by six outstanding keynote presentations giving by Prof. Chengqi Zhang, Prof. Phoebe Chen, Prof. Mohammed Bennamoun, Prof. X. Sean Wang, Prof. Shuqiang Jiang, and Prof. Yudong Zhang.

We would like to express our gratitude to all individuals, institutions, and sponsors that supported ADMA 2020. We also thank the PC members for completing the review process and providing valuable comments under a tight schedule constraint. This high-quality program would not have been possible without the expertise and dedication of our PC members.

We are grateful for the guidance of the general co-chairs Prof. Xue Li and Prof. Vincent Tseng, the time and effort of the program co-chairs Prof. Xiaochun Yang and A/Prof. Chang-Dong Wang, the publication chairs Dr. Md. Saiful Islam and Dr. Zheng Zhang, the publicity co-chairs Dr. Chi Yang, Dr. Guibing Guo, and Dr. Zheng Xiao, the tutorial co-chairs Dr. Dong Huang and Dr. Xiangmin Zhou, the workshop chair Prof. Zhao Zhang, the web chair Dr. Kang Su, and the local organization chair Dr. Xianlu Luo. We also would like to acknowledge the support of the members of the conference Steering Committee. All of them helped to make ADMA 2020 a success. Finally, we would like to take this valuable opportunity to thank all authors who submitted technical papers for contributing to the tradition of excellence in Advanced

Data Mining Applications. We firmly believe that many colleagues will find the papers in this proceeding exciting and beneficial for advancing their research.

November 2020

Xiaochun Yang
Chang-Dong Wang
Md. Saiful Islam
Zheng Zhang

Organization

General Co-chairs

Xue Li	Neusoft Education, China
Vincent Tseng	National Chiao Tung University, Taiwan

Program Committee Co-chairs

Xiaochun Yang	Northeastern University, China
Changdong Wang	Sun Yat-sen University, China

Tutorial Co-chairs

Dong Huang	South China Agricultural University, China
Xiangmin Zhou	RMIT University, Australia

Proceedings (Publication) Chairs

Md. Saiful Islam	Griffith University, Australia
Zheng Zhang	Harbin Institute of Technology, China

Publicity Co-chairs

Chi Yang	Huazhong University of Science and Technology, China
Guibing Guo	Northeastern University, China
Zheng Xiao	Neusoft Institute Guangdong, China

Workshop Chair

Zhao Zhang	Hefei University of Technology, China

Local Chair

Xianlu Luo	Neusoft Institute Guangdong, China

Web Chair

Kang Su	Neusoft Institute Guangdong, China

Program Committee

Noha Alduaiji	Majmaah University, Saudi Arabia
Md Musfique Anwar	Jahangirnagar University, Bangladesh
Tarique Anwar	Macquarie University, Australia
Taotao Cai	Deakin University, Australia
Yi Cai	South China University of Technology, China
Hongxu Chen	The University of Queensland, Australia
Lu Chen	Swinburne University of Technology, Australia
Tong Chen	The University of Queensland, Australia
Yurong Cheng	Beijing Institute of Technology, China
Farhana Choudhury	The University of Melbourne, Australia
Ningning Cui	Northeastern University, China
Lunke Fei	Guangdong University of Technology, Australia
Yunjun Gao	Zhejiang University, China
Yanhui Gu	Nanjing Normal University, China
Bin Guo	Institute Telecom SudParis, France
Dong Huang	South China Agricultural University, China
Guangyan Huang	Deakin University, Australia
Md. Saiful Islam	Griffith University, Australia
S. M. Riazul Islam	Sejong University, South Korea
Jing Jiang	University of Technology Sydney, Australia
Peiquan Jin	University of Science and Technology of China, China
Xiangjie Kong	Zhejiang University of Technology, China
Indika Kumara	Jheronimus Academy of Data Science, The Netherlands
Bohan Li	Nanjing University of Aeronautics and Astronautics, China
Lei Li	The University of Queensland, Australia
Xueping Peng	University of Technology Sydney, Australia
Dechang Pi	Nanjing University of Aeronautics and Astronautics, China
Muhammad Rahman	National Institutes of Health, USA
Iqbal H. Sarker	Chittagong University of Engineering and Technology, Bangladesh
Shaoxu Song	Tsinghua University, China
Eiji Uchino	Yamaguchi University, Japan
Sayan Unankard	Maejo University, Thailand
Can Wang	CSIRO, Australia
Chang-Dong Wang	Sun Yat-sen University, China
Hongzhi Wang	Harbin Institute of Technology, China
Junhu Wang	Griffith University, Australia
Weiqing Wang	Monash University, Australia
Xianzhi Wang	University of Technology Sydney, Australia
Jie Wen	Harbin Institute of Technology, China
Feng Xia	Federation University, Australia

Junchang Xin	Northeastern University, China
Jiajie Xu	Soochow University, China
Shan Xue	Macquarie University, Australia
Xiaochun Yang	Northeastern University, USA
Yajun Yang	Tianjin University, China
Hongzhi Yin	The University of Queensland, Australia
Peng Yuwei	Wuhan University, China
Xiaowang Zhang	Tianjin University, China
Zheng Zhang	Harbin Institute of Technology, China
Bin Zhao	Nanjing Normal University, Australia
Rui Zhou	Swinburne University of Technology, Australia
Xiangmin Zhou	RMIT University, Australia
Ye Zhu	Deakin University, Australia

Additional Reviewers

Aziz, Abdul
Dai, Chenglong
Han, Yuqiang
Hao, Bowen
He, Chengkun
Hou, Minglliang
Kayesh, Humayun
Li, Yicong
Liu, Jiaying
Lumbantoruan, Rosni
Shehzad, Ahsan
Sun, Xiangguo
Sun, Zhenchao
Wang, Connie
Wu, Peiyun
Ye, Guanhua
Yu, Junliang
Zhang, Limeng

Contents

Machine Learning

Text Mining

Graph Mining

Predictive Analytics

Recommender Systems

Privacy and Security

Query Processing

Data Mining Applications

Machine Learning

Subspace-Weighted Consensus Clustering for High-Dimensional Data

Xiaosha Cai[1,2] and Dong Huang[1,2](✉)

[1] College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China
xiaoshacai@hotmail.com

[2] Guangzhou Key Laboratory of Smart Agriculture, Guangzhou, China
huangdonghere@gmail.com

Abstract. Consensus clustering aims to combine multiple base clusters into a probably better and more robust clustering result. Despite the significant progress in recent years, the existing consensus clustering approaches are mostly designed for general-purpose scenarios, yet often lack the ability to effectively and efficiently deal with high-dimensional data. To this end, this paper proposes a subspace-weighted consensus clustering approach, which is based on two key observations in high-dimensional data. First, the cluster structures often lie in different subspaces in high-dimensional feature space. Second, the features in high-dimensional data may be of different importance and should be treated differently. Specifically, we utilize the Laplacian score to estimate the importance of different features. Then the weighted random sampling is performed repeatedly to produce a set of diverse random subspaces, in which multiple base clusters can thereby be generated. Further, the reliability of each base clustering is evaluated and weighted by considering the reliability of the features in the corresponding subspace, after which a subspace-weighted bipartite graph can be constructed and efficiently partitioned to obtain the final consensus clustering result. Experimental results on ten real-world high-dimensional datasets demonstrate the effectiveness and efficiency of the proposed approach.

Keywords: Consensus clustering · Ensemble clustering · Weighting mechanism · Laplacian score · High-dimensional data

1 Introduction

Data clustering is an unsupervised learning task whose purpose is to partition a set of data instances into multiple groups, each referred to as a cluster, such that the instances in the same cluster share high similarity while those in different clusters are as dissimilar as possible [16]. Different from the conventional clustering techniques that mostly use a single model to achieve a single clustering result, the consensus clustering technique has been drawing increasing attention

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 3–16, 2020.
https://doi.org/10.1007/978-3-030-65390-3_1

in recent years due to its ability to combine multiple base clusters into a probably better and more robust clustering result.

Consensus clustering is also known as ensemble clustering. It has proved to be a powerful tool for dealing with complex and noisy datasets [25]. In the past decade, many successful consensus clustering methods have been developed [4–6,9–11,13–15,28]. Among the existing consensus clustering methods, an important category of them is the pair-wise co-occurrence based methods [6,9]. Typically, Fred and Jain [6] proposed the evidence accumulation clustering (EAC) method, which constructs a co-association matrix by considering the pair-wise co-occurrence relationship, i.e., how many times two instances occur in the same cluster among the multiple base clusters. Then this co-association matrix is treated as the new similarity matrix for the data instances, and the final consensus clustering result can be built by exploiting agglomerative clustering [6]. To extend the EAC method, Huang et al. [9] constructed a new cluster-wise similarity matrix by exploring the higher-order relationship among the base clusters via random walks, which is then mapped from the cluster-level to the instance-level to obtain an enhanced co-association (ECA) matrix for the consensus clustering. Besides these co-occurrence based methods, another category of consensus clustering is the median partition based methods, which generally treat the consensus clustering problem as an optimization problem. Its purpose is to find a median partition (or median clustering) such that the similarity between this clustering and the base clusters is maximized [5,15,26]. Franek and Jiang [5] cast the consensus clustering problem into an Euclidean median problem, which can be solved by the Weiszfeld algorithm [24] and then mapped back to the clustering domain. Huang et al. [15] cast the consensus clustering problem into a binary linear programming problem, and solved it based on the factor graph model [2]. Furthermore, the graph partitioning based methods are also a main category of consensus clustering, which typically formulate the ensemble of base clusters into some graph structure, and then partition this graph to obtain the clustering result [4,11]. Fern and Brodley [4] built a bipartite graph by treating both instances and clusters as graph nodes, and partitioned it by the METIS algorithm. Huang et al. [11] devised an ultra-scalable spectral clustering (U-SPEC) method by means of hybrid representative selection and fast K-nearest representative approximation, and then combined multiple U-SPEC clusterers into an ultra-scalable ensemble clustering (U-SENC) framework, so as to robustly and efficiently obtain the consensus clustering result for very large-scale datasets.

These methods deal with the consensus clustering problem from different technical perspectives [4–6,9–11,13,15]. Despite significant success, the existing consensus clustering methods are mostly designed for low-dimensional datasets, yet lack the ability to effectively handle high-dimensional applications. In high-dimensional data, it is often recognized that the cluster structures can be revealed in various subspaces [1,3]. To this end, some efforts have been made to incorporate the feature sampling technique into the consensus clustering framework. Yu et al. [27] utilized random sampling to produce a set of random subspaces, and presented the incremental semi-supervised clustering ensemble algorithm. Jing et al. [17] proposed three new consensus clustering algorithms by exploiting stratified feature sampling. Though multiple subspaces are explored in these methods [17,27], they generally performed feature sampling with all features treated equally, which neglect the potentially different importance of different features in high-dimensional data. Furthermore, after the multiple subspaces are obtained, they also lack the ability to evaluate the reliability of these subspaces and weight them accordingly.

To address the above problems, in this paper, we propose a subspace-weighted consensus clustering (SWCC) approach for high-dimensional data. Specifically, the Laplacian score [8] is first exploited to estimate the importance of different features in the high-dimensional space, based on which the weighted random sampling can be performed to obtained a set of random subspaces. Then the reliability of each generated subspace is evaluated according to the reliability of the features inside it, and thus the subspace weighting mechanism can be designed. With the multiple subspaces with weights, multiple base clusters are further generated, where the weights of the subspaces are used for the base clusters. Finally, a subspace-weighted bipartite graph is constructed for the multiple base clusters and efficiently partitioned to achieve the consensus clustering result. Extensive experiments are conducted on ten real-world high-dimensional datasets, which have shown the effectiveness and efficiency of our SWCC approach.

The remainder of the paper is organized as follows. The proposed consensus clustering approach is described in Sect. 2. The experimental results are reported and analyzed in Sect. 3. Finally, the paper is concluded in Sect. 4.

2 Subspace-Weighted Consensus Clustering

In this section, we describe the details of the proposed SWCC approach. The formulation of the consensus clustering problem is given in Sect. 2.1. The ensemble generation with weighted random sampling is presented in Sect. 2.2. Finally, the subspace-weighted consensus function with bipartite graph is designed in Sect. 2.3.

2.1 The Consensus Clustering Problem

Let $X = (x_1, x_2, \cdots, x_n)$ denote a dataset with n instances, where x_i is the i-th instance. Let $F = (f_1, f_2, \cdots, f_d)^T$ denote the set of data features, where d is the dimension and f_i is the i-th feature.

By running multiple clustering algorithms or the same algorithm repeatedly with different initializations and parameters, we can obtain a set of diverse base clusters, denoted as

$$\Pi = \{\pi^1, \pi^2, \cdots, \pi^M\}, \tag{1}$$

where M is the number of base clusters and π^i is the i-th base clustering in Π. Each base clustering consists of a certain number of clusters, which can be denoted as

$$\pi^i = \{C_1^i, C_2^i, \cdots, C_{n^i}^i\}, \tag{2}$$

where n^i denotes the number of clusters and C_j^i denotes the j-th cluster in π^i. For convenience, we can denote the set of all clusters in the multiple base clusters as follows

$$\mathcal{C} = \{C_1, C_2, \cdots, C_{n^c}\}, \tag{3}$$

where n^c is the total number of clusters in Π. It is obvious that $n^c = n^1 + n^2 + \cdots + n^M$. The purpose of consensus clustering is to fuse the information of the multiple base clusters in Π and construct a better clustering result π^*.

2.2 Feature-Weighted Ensemble Generation

In this section, we first estimate the importance of each feature by Laplacian score, then perform weighted random sampling to obtain a set of subspaces, and finally generate the ensemble of base clusters from these subspaces.

The Laplacian score is able to evaluate the importance of each feature by considering its locality-preserving power [8,12,19]. To reflect the local structure of data, a K-nearest neighbor (K-NN) graph is constructed as follows:

$$G = \{V, E\}, \tag{4}$$

where $V = X$ is the node set, $E = \{e_{ij}\}_{n\times n}$ is the adjacent matrix, and e_{ij} is the edge weight between node i and node j, which can be computed as

$$e_{ij} = \begin{cases} \exp\left(-\frac{d(x_i - x_j)}{2*\sigma}\right), & \text{if } x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i), \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $d(x_i - x_j)$ denotes the Euclidean distance between x_i and x_j, σ is the kernel parameter which is computed by taking average of all distances, and $KNN(x_i)$ represents the set of K-nearest neighbors of x_i.

Let D denote the degree matrix, that is

$$D = \{\hat{d}_{ij}\}_{n\times n}, \tag{6}$$

$$\hat{d}_{ij} = \begin{cases} \sum_{h=1}^{n} e_{ih}, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Then we have the graph Laplacian $L = D - E$. Further, the Laplacian score of the i-th feature, i.e., f_i, can be computed as

$$y_i = \frac{\tilde{f_i}^T L \tilde{f_i}}{\tilde{f_i} D \tilde{f_i}} \tag{8}$$

with

$$\tilde{f}_i = f_i - \frac{f_i^T D\mathbf{1}}{\mathbf{1}^T D\mathbf{1}}\mathbf{1} \tag{9}$$

$$\mathbf{1} = (1, 1, \cdots, 1)^T \tag{10}$$

Thereby, the Laplacian scores of all features can be denoted as

$$Y = (y_1, y_2, \cdots, y_d)^T. \tag{11}$$

Note that a lower Laplacian score indicates that this feature may have better locality-preserving power and higher importance [8]. Based on the Laplacian scores, the feature weights can thereby be computed as follows

$$W = \mathbf{1} - Y = (w_1, w_2, \cdots, w_d)^T, \tag{12}$$

where w_i is the weight of feature f_i. The feature weights are further normalized as

$$\bar{w_i} = \frac{w_i}{\sum_{j=1}^{d} w_j}, \tag{13}$$

Thus we have the normalized weights of features as $\bar{W} = (\bar{w_1}, \bar{w_2}, \cdots, \bar{w_d})^T$.

With the feature weights obtained, the weighted random sampling can be performed to generate a set of random subspaces with a certain sampling ratio r, where the features with greater weights will have higher probability to get sampled. Formally, the set of subspaces is denoted as $\mathcal{S} = \{S_1, S_2, \cdots, S_M\}$, where S_i denotes the i-th subspace. In each subspace, a base clustering is then generated by the k-means algorithm with random initialization and a random number of clusters. Therefore, the ensemble of M base clusters, i.e., Π, can be constructed.

2.3 Subspace-Weighted Consensus Function with Bipartite Graph

In this section, we proceed to fuse the generated base clusters into a final consensus clustering result. Note that each base clustering is generated in a subspace, and it is intuitive that the reliability of a base clustering can be estimated according to the subspace that it results from. Each subspace is a set of data features, and its reliability can be evaluated and weighted by considering the Laplacian scores of the features inside it. Specifically, the weight of a subspace, say, S_i, can be computed as

$$W_i^S = \frac{\sum_{f_j \in S_i} w_j}{|S_i|}, \tag{14}$$

where $|S_i|$ denotes the number of features in the subspace S_i. Then, the weight of the subspace S_i will be used as the weight of the i-th base clustering, i.e.,

π^i. Thereby, with both clusters and instances treated as nodes, the subspace-weighted bipartite graph can be defined as

$$\tilde{G} = \{\tilde{V}, \tilde{E}\} \tag{15}$$

where $\tilde{V} = \{v_1, v_2, \cdots, v_{n+n^c}\} = X \cup \mathcal{C}$ is the node set, and $\tilde{E}$ is the adjacent matrix. Note that $\tilde{G}$ is a bipartite graph, where a non-zero edge between two nodes exists if and only if one of the nodes is a data instance and the other is the cluster that contains it. For each edge between a cluster and an instance, its edge weight is determined by the weight of the base clustering (corresponding to the weight of its associated subspace) that contains this cluster. Formally, the edge weight between node i and node j is computed as

$$\tilde{e}_{ij} = \begin{cases} W_p^S, & \text{if } v_i \in X, v_j \in \mathcal{C}, v_i \in v_j \text{ and } v_j \in \pi^p, \\ W_q^S, & \text{if } v_j \in X, v_i \in \mathcal{C}, v_j \in v_i \text{ and } v_i \in \pi^q, \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

With the subspace-weighted bipartite graph constructed, the transfer cut [18] can then be utilized to efficiently partition the graph to obtain the final consensus clustering result.

3 Experiments

In this section, we experimentally evaluate the proposed SWCC approach against several baseline consensus clustering approaches on multiple real-world high-dimensional datasets. All the experiments are carried out in Matlab R2016b on a computer with an i5-8400 CPU and 64 GB of RAM.

Table 1. Dataset Description

Dataset	Abbr.	#Instance	Dimension	#Class
Dyrskjot-2003	*DS-1*	40	1203	3
Golub-1999-v2	*DS-2*	72	1868	3
Pomeroy-2002-v2	*DS-3*	42	1379	5
Yeoh-2002-v2	*DS-4*	248	2526	6
Bredel-2005	*DS-5*	50	1739	3
Khan-2001	*DS-6*	83	1069	4
Risinger-2003	*DS-7*	42	1771	4
WebKB-Cornell	*DS-8*	827	4134	7
WebKB-Washington	*DS-9*	1166	4165	7
WebKB-Wisconsin	*DS-10*	1210	4189	6

Table 2. Average NMI (%) over 100 runs by different methods on the benchmark datasets. The best score in each row is in bold.

Dataset	SWCC	LWGP	LWEA	SEC	PTGP	PTA	KCC
DS-1	$\mathbf{56.46}_{\pm 7.82}$	$46.18_{\pm 3.90}$	$50.36_{\pm 6.70}$	$46.83_{\pm 7.84}$	$49.91_{\pm 6.38}$	$46.65_{\pm 9.16}$	$50.58_{\pm 6.59}$
DS-2	$\mathbf{68.38}_{\pm 8.30}$	$67.28_{\pm 9.17}$	$62.21_{\pm 9.29}$	$56.04_{\pm 14.13}$	$66.91_{\pm 7.65}$	$60.22_{\pm 14.68}$	$60.63_{\pm 10.82}$
DS-3	$\mathbf{64.18}_{\pm 4.73}$	$53.95_{\pm 2.72}$	$54.00_{\pm 3.80}$	$51.85_{\pm 6.86}$	$55.17_{\pm 3.45}$	$55.44_{\pm 3.84}$	$55.50_{\pm 5.60}$
DS-4	$\mathbf{41.56}_{\pm 5.40}$	$30.55_{\pm 3.51}$	$27.25_{\pm 3.11}$	$35.24_{\pm 6.25}$	$34.66_{\pm 4.03}$	$34.23_{\pm 4.11}$	$36.77_{\pm 5.67}$
DS-5	$\mathbf{37.49}_{\pm 3.78}$	$34.19_{\pm 3.92}$	$34.77_{\pm 6.57}$	$34.68_{\pm 5.78}$	$31.04_{\pm 5.56}$	$33.30_{\pm 8.18}$	$32.35_{\pm 5.23}$
DS-6	$\mathbf{63.64}_{\pm 10.04}$	$56.00_{\pm 3.07}$	$57.80_{\pm 6.65}$	$46.16_{\pm 13.26}$	$55.32_{\pm 3.07}$	$48.10_{\pm 7.62}$	$52.38_{\pm 13.99}$
DS-7	$\mathbf{31.13}_{\pm 4.18}$	$17.36_{\pm 4.66}$	$19.27_{\pm 4.61}$	$27.10_{\pm 6.30}$	$29.85_{\pm 3.65}$	$29.90_{\pm 3.68}$	$29.12_{\pm 3.94}$
DS-8	$\mathbf{24.78}_{\pm 1.45}$	$5.38_{\pm 0.73}$	$2.26_{\pm 0.74}$	$6.90_{\pm 2.28}$	$8.14_{\pm 1.30}$	$8.24_{\pm 1.47}$	$8.57_{\pm 2.15}$
DS-9	$\mathbf{20.85}_{\pm 2.46}$	$6.49_{\pm 1.87}$	$1.78_{\pm 0.78}$	$10.23_{\pm 1.81}$	$11.82_{\pm 1.20}$	$12.98_{\pm 1.65}$	$10.34_{\pm 1.46}$
DS-10	$\mathbf{21.28}_{\pm 1.46}$	$8.03_{\pm 0.75}$	$4.01_{\pm 1.08}$	$10.88_{\pm 2.71}$	$12.27_{\pm 2.23}$	$12.97_{\pm 2.41}$	$11.27_{\pm 1.80}$
Avg. score	**42.98**	32.54	31.37	32.59	35.51	34.20	34.75
Avg. rank	**1.00**	5.30	5.00	5.20	3.90	4.00	3.60

Table 3. Average ARI (%) over 100 runs by different methods on the benchmark datasets. The best score in each row is in bold.

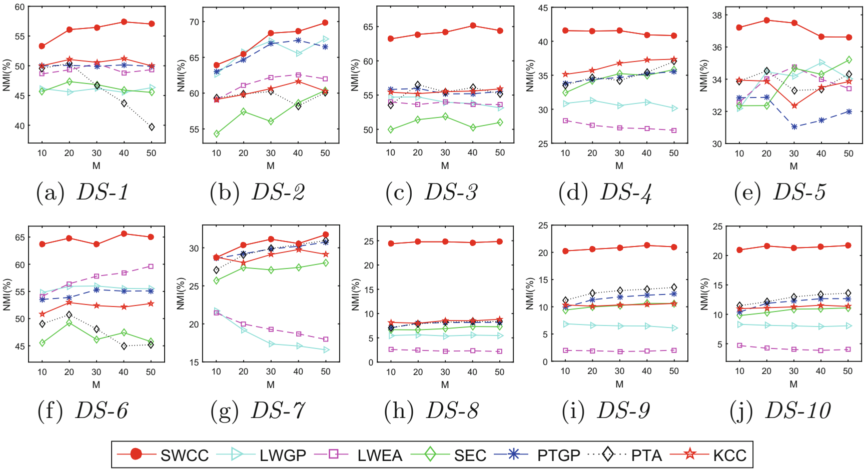
Dataset	SWCC	LWGP	LWEA	SEC	PTGP	PTA	KCC
DS-1	$\mathbf{61.57}_{\pm 8.40}$	$51.67_{\pm 4.26}$	$55.26_{\pm 6.38}$	$47.99_{\pm 11.60}$	$54.85_{\pm 5.15}$	$45.11_{\pm 14.28}$	$54.54_{\pm 7.90}$
DS-2	$\mathbf{71.29}_{\pm 9.71}$	$68.38_{\pm 9.77}$	$64.20_{\pm 11.19}$	$56.28_{\pm 17.89}$	$70.20_{\pm 8.93}$	$58.08_{\pm 20.79}$	$62.69_{\pm 15.37}$
DS-3	$\mathbf{54.15}_{\pm 6.11}$	$42.26_{\pm 3.47}$	$41.25_{\pm 5.28}$	$39.34_{\pm 8.41}$	$41.48_{\pm 4.94}$	$42.27_{\pm 5.25}$	$43.33_{\pm 7.87}$
DS-4	$22.09_{\pm 7.24}$	$18.93_{\pm 1.62}$	$17.50_{\pm 1.27}$	$20.23_{\pm 6.72}$	$17.43_{\pm 5.28}$	$16.56_{\pm 4.81}$	$\mathbf{22.72}_{\pm 6.89}$
DS-5	$\mathbf{35.28}_{\pm 3.26}$	$29.55_{\pm 7.75}$	$31.65_{\pm 11.67}$	$34.12_{\pm 9.21}$	$30.45_{\pm 9.05}$	$32.04_{\pm 12.96}$	$29.80_{\pm 8.06}$
DS-6	$\mathbf{49.72}_{\pm 14.96}$	$35.18_{\pm 3.00}$	$36.62_{\pm 6.35}$	$31.29_{\pm 14.53}$	$35.33_{\pm 3.38}$	$30.27_{\pm 7.04}$	$36.53_{\pm 15.71}$
DS-7	$\mathbf{20.98}_{\pm 5.46}$	$-4.05_{\pm 5.12}$	$-2.53_{\pm 4.87}$	$10.42_{\pm 7.49}$	$11.40_{\pm 3.90}$	$11.93_{\pm 4.96}$	$12.30_{\pm 5.18}$
DS-8	$\mathbf{16.84}_{\pm 2.29}$	$3.63_{\pm 2.09}$	$-2.39_{\pm 2.88}$	$9.82_{\pm 4.31}$	$10.61_{\pm 2.41}$	$10.53_{\pm 2.83}$	$10.53_{\pm 4.79}$
DS-9	$17.13_{\pm 3.65}$	$12.25_{\pm 5.74}$	$-0.47_{\pm 6.16}$	$17.09_{\pm 4.68}$	$14.39_{\pm 2.73}$	$\mathbf{22.74}_{\pm 2.80}$	$14.22_{\pm 4.71}$
DS-10	$\mathbf{19.53}_{\pm 3.10}$	$3.88_{\pm 1.71}$	$-6.82_{\pm 2.83}$	$9.76_{\pm 5.28}$	$14.39_{\pm 2.73}$	$14.24_{\pm 3.40}$	$10.53_{\pm 3.67}$
Avg. score	**36.86**	26.17	23.43	27.63	30.05	28.38	29.72
Avg. rank	**1.20**	5.30	5.00	4.90	3.70	4.30	3.50

3.1 Datasets and Evaluation Measures

We used ten high-dimensional datasets in the experiments, namely, *Dyrskjot-2003* [22], *Golub-1999-v2* [22], *Pomeroy-2002-v2* [22], *Yeoh-2002-v2* [22], *Bredel-2005* [22], *Khan-2001* [22], *Risinger-2003* [22], *WebKB-Cornell* [7], *WebKB-Washington* [7] and *WebKB-Wisconsin* [7]. These ten datasets are abbreviated as *DS-1* to *DS-10* for simplicity. Details of the datasets are presented in Table 1.

In our experiments, we use the ensemble size $M = 30$, the number of nearest neighbors $K = 5$, and the sampling ratio $r = 0.1$ on all the benchmark datasets. The performance of our approach w.r.t. varying parameters M, K, and r will also be evaluated in Sects. 3.3, 3.4, and 3.5, respectively.

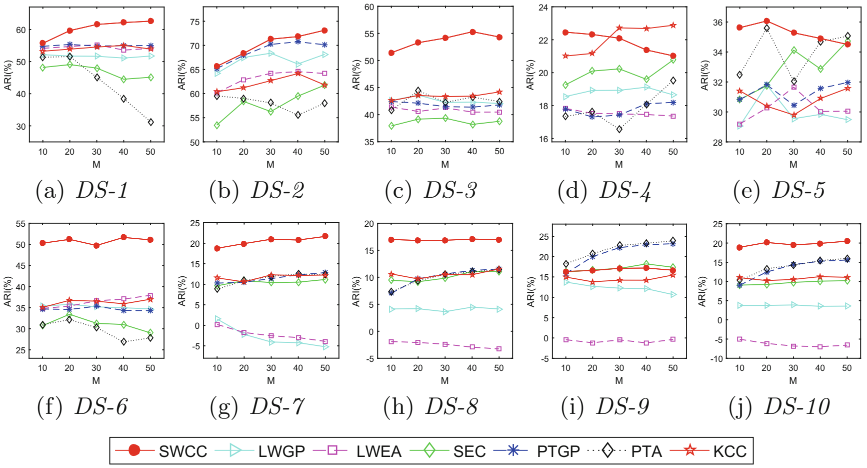
To compare the clustering results of different consensus clustering approaches, two widely-used evaluation measures are adopted, namely, normalized mutual information (NMI) [20] and adjusted Rand index (ARI) [23]. The larger the NMI and ARI values are, the more consistent the clustering result is with the ground-truth.



(a) *DS-1* (b) *DS-2* (c) *DS-3* (d) *DS-4* (e) *DS-5*

(f) *DS-6* (g) *DS-7* (h) *DS-8* (i) *DS-9* (j) *DS-10*

Fig. 1. Average NMI (%) over 100 runs by different methods with varying ensemble size M.



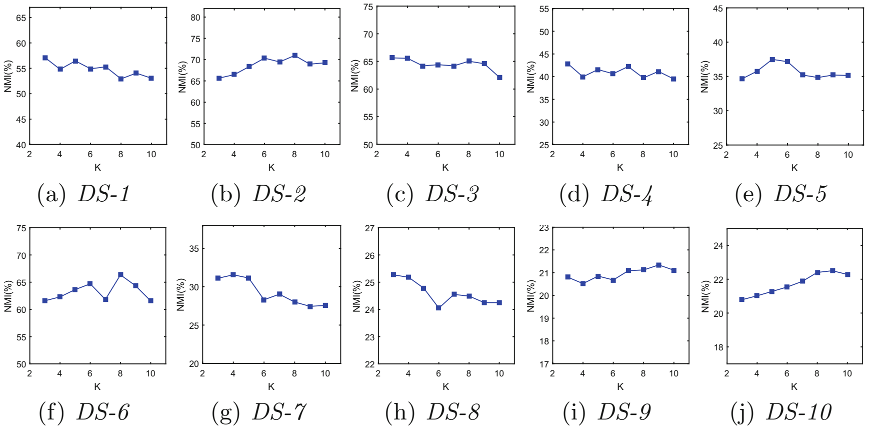
(a) *DS-1* (b) *DS-2* (c) *DS-3* (d) *DS-4* (e) *DS-5*

(f) *DS-6* (g) *DS-7* (h) *DS-8* (i) *DS-9* (j) *DS-10*

Fig. 2. Average ARI (%) over 100 runs by different methods with varying ensemble size M.

3.2 Comparison with Other Consensus Clustering Algorithms

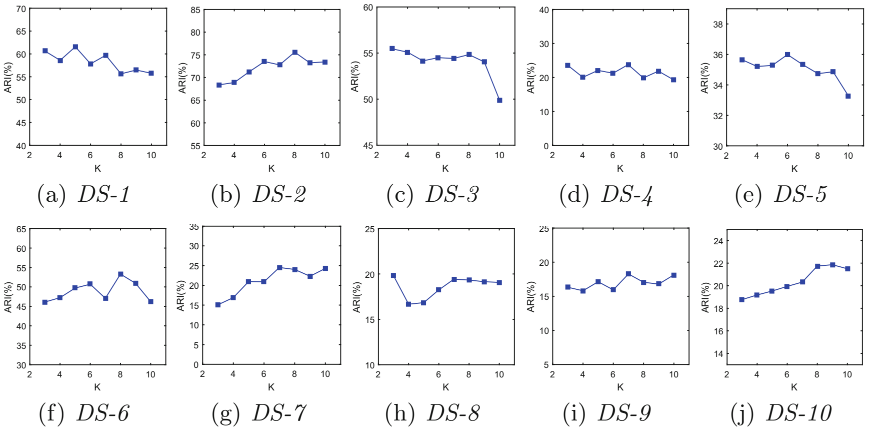
In this section, we compare the proposed SWCC algorithm with six baseline consensus clustering algorithms, namely, locally weighted graph partitioning (LWGP) [10], locally weighted evidence accumulation (LWEA) [10], spectral ensemble clustering (SEC) [21], probability trajectory based graph partitioning (PTGP) [13], probability trajectory accumulation (PTA) [13] and k-means based consensus clustering (KCC) [25].



(a) *DS-1* (b) *DS-2* (c) *DS-3* (d) *DS-4* (e) *DS-5*

(f) *DS-6* (g) *DS-7* (h) *DS-8* (i) *DS-9* (j) *DS-10*

Fig. 3. Average NMI (%) by SWCC with varying number of nearest neighbors K.

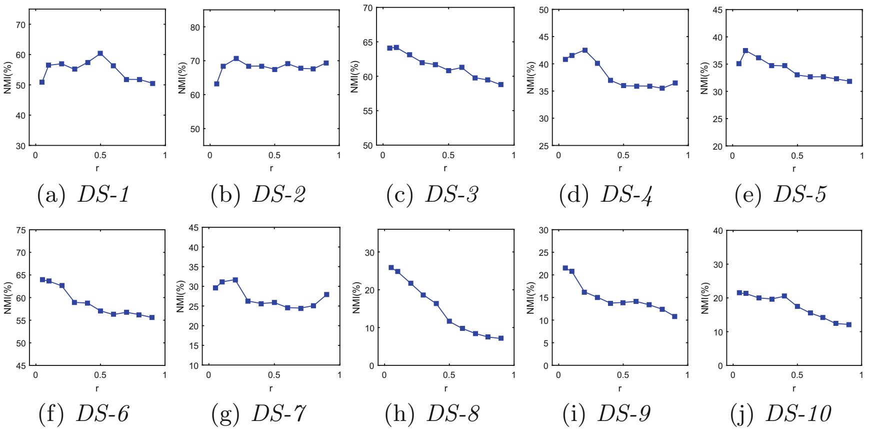


(a) *DS-1* (b) *DS-2* (c) *DS-3* (d) *DS-4* (e) *DS-5*

(f) *DS-6* (g) *DS-7* (h) *DS-8* (i) *DS-9* (j) *DS-10*

Fig. 4. Average ARI (%) by SWCC with varying number of nearest neighbors K.

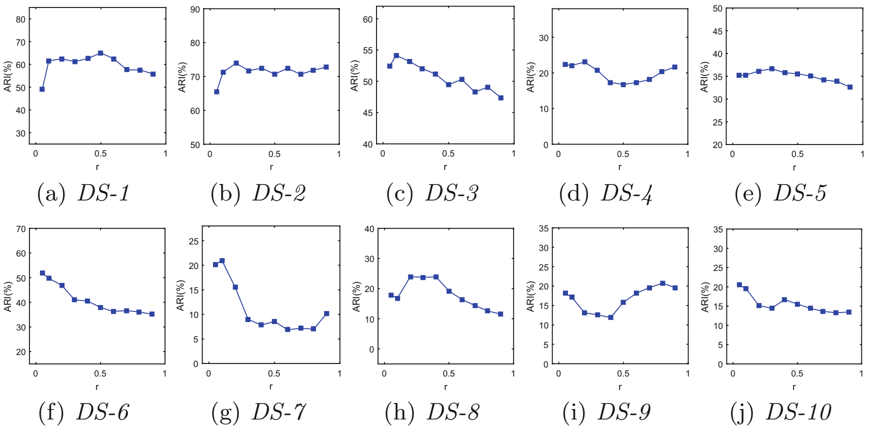
Tables 2 and 3 respectively report the NMI and ARI scores (over 100 runs) of the SWCC algorithm and the baseline algorithms. Specifically, in terms of NMI, our SWCC algorithm outperforms all the baseline algorithms in all of the ten benchmark datasets. In terms of ARI, SWCC also achieves the highest score in eight out of the ten datasets. Moreover, the average scores of NMI (%) and ARI (%) of SWCC (across the ten datasets) are 42.98 and 36.86, respectively, while that of the second best algorithm, i.e., PTGP, are 35.51 and 30.05, respectively. In terms of average rank, our SWCC algorithm also achieves the best average ranks of 1.00 and 1.20 w.r.t. NMI (%) and ARI (%), respectively.



(a) *DS-1* (b) *DS-2* (c) *DS-3* (d) *DS-4* (e) *DS-5*

(f) *DS-6* (g) *DS-7* (h) *DS-8* (i) *DS-9* (j) *DS-10*

Fig. 5. Average NMI (%) by SWCC with varying sampling ratio r.



(a) *DS-1* (b) *DS-2* (c) *DS-3* (d) *DS-4* (e) *DS-5*

(f) *DS-6* (g) *DS-7* (h) *DS-8* (i) *DS-9* (j) *DS-10*

Fig. 6. Average ARI (%) by SWCC with varying sampling ratio r.

3.3 Robustness to Ensemble Size

Consensus clustering fuses multiple base clusters into a probably better clustering result. The number of base clusters is also called ensemble size. In this section, we compare the performances of different consensus clustering algorithms with varying ensemble sizes.

As shown in Figs. 1 and 2, our SWCC algorithm exhibits consistently high performance (w.r.t. both NMI and ARI) as the ensemble size goes from 10 to 50. When compared to other consensus clustering algorithms, SWCC outperforms most of the baseline algorithms with different ensemble sizes. The advantages of our SWCC algorithm over the baseline algorithms are particularly significant on the *DS-1*, *DS-3*, *DS-6*, *DS-8*, and *DS-10* datasets (as can be seen in Figs. 1 and 2).

3.4 Influence of the Number of Nearest Neighbors K

In this section, we evaluate the influence of the number of nearest neighbors K in the proposed SWCC algorithm. As shown in Figs. 3 and 4, the performance of SWCC is overall stable with different numbers of nearest neighbors in terms of NMI and ARI. Empirically, a moderate value of K, e.g., in the range of [4,8], is preferred. In our experiments, $K = 5$ is used on all benchmark datasets.

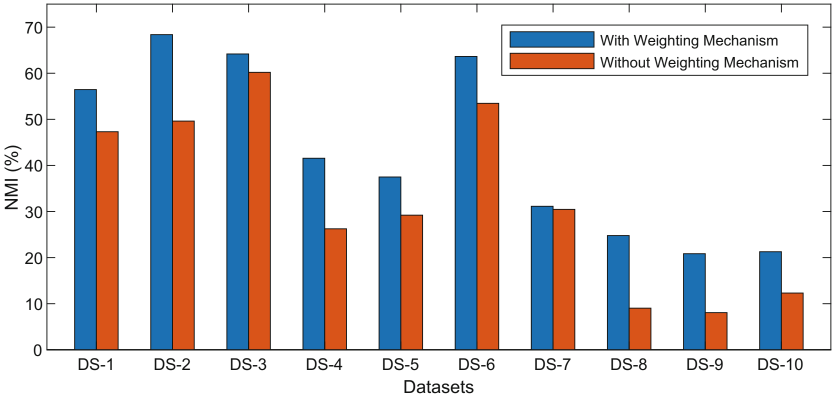


Fig. 7. Average NMI (%) by SWCC with and without using the weighting mechanism.

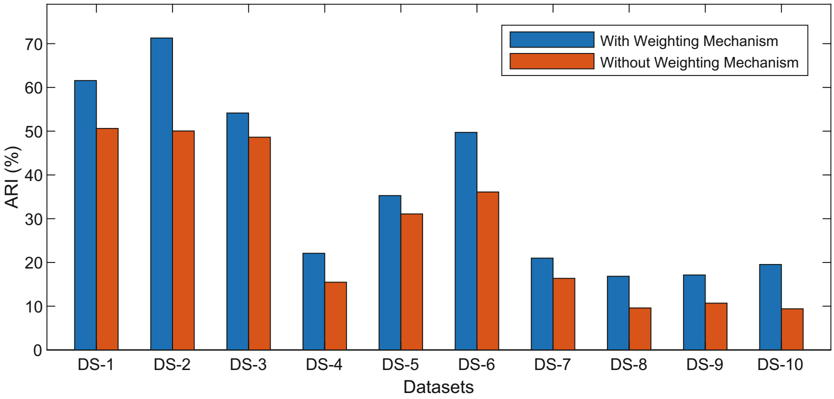


Fig. 8. Average ARI (%) by SWCC with and without using the weighting mechanism.

3.5 Influence of the Sampling Ratio r

In this section, we evaluate the influence of the sampling ratio r in the proposed SWCC algorithm. As shown in Figs. 5 and 6, a relatively small value of r is

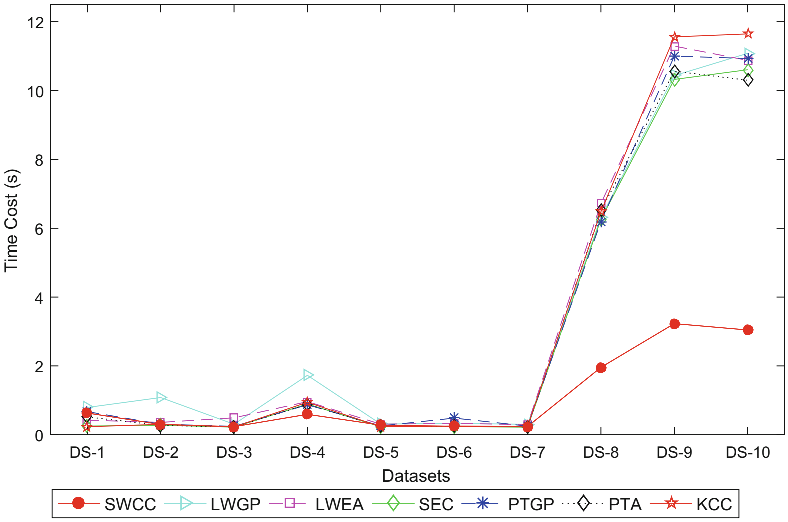


Fig. 9. Average time cost(s) by different consensus clustering methods on the benchmark datasets.

usually beneficial to the performance of SWCC, probably due to the fact that a smaller sampling ratio may bring in more diversity in the ensemble generation phase. Empirically, the sampling ratio r is suggested to be set in the range of $[0.1, 0.3]$. In our experiments, $r = 0.1$ is used on all benchmark datasets.

3.6 Influence of the Weighting Mechanism

In the section, we test the performance of the proposed SWCC algorithm *with* and *without* using the weighting mechanism. As shown in Figs. 7 and 8, the incorporation of the weighting mechanism brings in improvements in the NMI and ARI scores on all of the ten datasets. Especially, rather obvious improvements have been achieved on the *DS-1*, *DS-2*, *DS-4*, *DS-6*, *DS-8*, *DS-9*, and *DS-10* datasets.

3.7 Time Efficiency

In this section, we compare the time costs (in seconds) of different consensus clustering algorithms on the benchmark datasets. As can be seen in Fig. 9, on the first seven small datasets, our SWCC algorithm shows comparable time efficiency to the baseline algorithms. As the data size increases, on the last three larger datasets, i.e., *DS-8*, *DS-9*, and *DS-10*, our SWCC algorithm exhibits a much clearer advantage over the baseline algorithms in the time efficiency.

To summarize, the overall experimental results in Tables 2 and 3 and Figs. 1, 2 and 9 have shown that, the proposed SWCC algorithm is able to produce much better and more robust consensus clustering results than the baseline algorithms while maintaining high efficiency.

4 Conclusion

In this paper, we present a novel consensus clustering approach termed SWCC for high-dimensional data. Specifically, we first estimate the importance of the features by the Laplacian score, which takes the locality-preserving power of the features into consideration. Then we perform the weighted random sampling to obtain a set of subspaces, based on which an ensemble of base clusters can be built. Thereafter, the reliability of each subspace is evaluated by considering the reliability of the features inside it, which is further used for evaluating and weighting the base clusters in the consensus function. Finally, a subspace-weighted bipartite graph is constructed and partitioned to achieve the consensus clustering result. Experiments on ten benchmark datasets have demonstrated the superiority of the proposed approach in terms of both effectiveness and efficiency.

Acknowledgments. This work was supported by NSFC (61976097).

References

1. Cai, X., Huang, D., Wang, C.D., Kwoh, C.K.: Spectral clustering by subspace randomization and graph fusion for high-dimensional data. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), pp. 330–342 (2020)
2. Dueck, D.: Affinity Propagation: Clustering Data by Passing Messages. Ph.D. thesis, University of Toronto (2009)
3. Elhamifar, E., Vidal, R.: Sparse subspace clustering: algorithm, theory, and applications. IEEE Trans. Pattern Anal. Mach. Intell. **35**(11), 2765–2781 (2013)
4. Fern, X.Z., Brodley, C.E.: Solving cluster ensemble problems by bipartite graph partitioning. In: Proceedings of the International Conference on Machine Learning (ICML) (2004)
5. Franek, L., Jiang, X.: Ensemble clustering by means of clustering embedding in vector spaces. Pattern Recogn. **47**(2), 833–842 (2014)
6. Fred, A.L.N., Jain, A.K.: Combining multiple clusterings using evidence accumulation. IEEE Trans. Pattern Anal. Mach. Intell. **27**(6), 835–850 (2005)
7. Gu, Q., Zhou, J.: Subspace maximum margin clustering. In: Proceedings of the ACM Conference on Information and Knowledge Management (CIKM), pp. 1337–1346 (2009)
8. He, X., Deng, C., Niyogi, P.: Laplacian score for feature selection. In: Advances in Neural Information Processing Systems (2005)
9. Huang, D., Wang, C., Peng, H., Lai, J., Kwoh, C.: Enhanced ensemble clustering via fast propagation of cluster-wise similarities. IEEE Trans. Syst. Man. Cybern. Syst. (2018, in press). https://doi.org/10.1109/TSMC.2018.2876202
10. Huang, D., Wang, C.D., Lai, J.H.: Locally weighted ensemble clustering. IEEE Trans. Cybern **48**(5), 1460–1473 (2018)
11. Huang, D., Wang, C.D., Wu, J.S., Lai, J.H., Kwoh, C.K.: Ultra-scalable spectral clustering and ensemble clustering. IEEE Trans. Knowl. Data Eng. **32**(6), 1212–1226 (2020)
12. Huang, D., Cai, X., Wang, C.D.: Unsupervised feature selection with multi-subspace randomization and collaboration. Knowl.-Based Syst. **182**, 104856 (2019)

13. Huang, D., Lai, J.H., Wang, C.D.: Robust ensemble clustering using probability trajectories. IEEE Trans. Knowl. Data Eng. **28**(5), 1312–1326 (2016)
14. Huang, D., Lai, J.H., Wang, C.D., Yuen, P.C.: Ensembling over-segmentations: from weak evidence to strong segmentation. Neurocomputing **207**, 416–427 (2016)
15. Huang, D., Lai, J., Wang, C.D.: Ensemble clustering using factor graph. Pattern Recogn. **50**, 131–142 (2016)
16. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recogn. Lett. **31**(8), 651–666 (2010)
17. Jing, L., Tian, K., Huang, J.Z.: Stratified feature sampling method for ensemble clustering of high dimensional data. Pattern Recogn. **48**(11), 3688–3702 (2015)
18. Li, Z., Wu, X.M., Chang, S.F.: Segmentation using superpixels: a bipartite graph partitioning approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
19. Liang, J., Huang, D.: Laplacian-weighted random forest for high-dimensional data classification. In: Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI) pp. 748–753 (2019)
20. Liang, Y., Huang, D., Wang, C.D.: Consistency meets inconsistency: a unified graph learning framework for multi-view clustering. In: Proceedings of the of IEEE International Conference on Data Mining (ICDM), pp. 1204–1209 (2019)
21. Liu, H., Wu, J., Liu, T., Tao, D., Fu, Y.: Spectral ensemble clustering via weighted k-means: theoretical and practical evidence. IEEE Trans. Knowl. Data Eng. **29**(5), 1129–1143 (2017)
22. de Souto, M.C., Costa, I.G., de Araujo, D.S., Ludermir, T.B., Schliep, A.: Clustering cancer gene expression data: a comparative study. BMC Bioinform. **9**(1), 497 (2008)
23. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J. Mach. Learn. Res. **11**(11), 2837–2854 (2010)
24. Weiszfeld, E., Plastria, F.: On the point for which the sum of the distances to n given points is minimum. Ann. Oper. Res. **167**(1), 7–41 (2009)
25. Wu, J., Liu, H., Xiong, H., Cao, J., Chen, J.: K-means-based consensus clustering: a unified view. IEEE Trans. Knowl. Data Eng. **27**(1), 155–169 (2015)
26. Xu, Y., Zhang, Z., Lu, G., Yang, J.: Approximately symmetrical face images for image preprocessing in face recognition and sparse representation based classification. Pattern Recogn. **54**, 68–82 (2016)
27. Yu, Z., et al.: Incremental semi-supervised clustering ensemble for high dimensional data clustering. IEEE Trans. Knowl. Data Eng. **28**(3), 701–714 (2016)
28. Zhang, Z., Liu, L., Shen, F., Shen, H.T., Shao, L.: Binary multi-view clustering. IEEE Trans. Pattern Anal. Mach. Intell. **41**(7), 1774–1782 (2018)

NOV-RSI: A Novel Optimization Algorithm for Mining Rare Significance Itemsets

Huan Phan[1,2,4](✉) and Bac Le[3,4]

[1] Division of IT, VNUHCM-University of Social Sciences and Humanities,
Ho Chi Minh City, Vietnam
huanphan@hcmussh.edu.vn

[2] Faculty of Mathematics and Computer Science, VNUHCM-University of Science,
Ho Chi Minh City, Vietnam

[3] Faculty of IT, VNUHCM-University of Science, Ho Chi Minh City, Vietnam
lhbac@fithcmus.edu.vn

[4] Vietnam National University, Ho Chi Minh City, Vietnam

Abstract. Rare itemsets mining is an important task for potential applications such as the detection of computer attacks, fraudulent transactions in financial institutions, bioinformatics and medicine. In the traditional data mining on transaction databases, such items have no weight (equal weight, as equal to 1). However, in real world application, each item often has a different weight (the importance/significance of each item). Therefore, we need to mine weighted frequent/rare itemsets on transaction databases. In this paper, we propose an algorithm for mining rare significance itemsets based on NOT satisfy the downward closure property. We propose an efficient algorithm called NOV-RSI. The experimental results show that the proposed algorithm performs faster than other existing algorithms on both real-life datasets of UCI and synthetic datasets generated by IBM Almaden.

Keywords: Data mining · NOT satisfy the downward closure property · NOV-RSI algorithm · Rare significance itemset

1 Introduction

For more than two decades, most of the researches are for mining frequent itemsets with the weights/significance of all items are the same (*equal weight, as equal to 1*), the algorithmic approaches based on Apriori [1] and FP-Tree [2]. In addition, to speed up the execution of mining frequent itemsets, Phan et al. proposed NOV-FI [3] algorithm based on the Kernel_COOC array. Besides, rare itemsets mining is an important task for potential applications such as the detection of computer attacks, fraudulent transactions in financial institutions, bioinformatics and medicine. Algorithms such as Apriori-Inverse [4] and Rarity [5] implement an *Apriori-like* approach. Thereafter to speed up the execution of mining minimal rare itemsets, Szathmary et al. proposed Walky-G [6] algorithm based on the IT-Tree structure. But in real-world applications,

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 17–29, 2020.
https://doi.org/10.1007/978-3-030-65390-3_2

items can have different significance/importance in databases, and such databases are called weighted databases. Most algorithms for frequent weighted/significance itemsets mining are based on *satisfying the downward closure property* such as algorithms [7–9]. However, Huai et al. [10] proposed *Apriori-like* algorithms based on approach NOT *satisfy the downward closure property* (*very rare proposed algorithms following this approach*). This is a great challenge.

In this paper, we propose a novel algorithm called NOV-RSI for mining rare *significance* itemsets based on *NOT satisfying the downward closure property*. Furthermore, the proposed algorithm is easily expanded on parallel computing systems. The paper has algorithms as follows:

- *Algorithm 1:* Computing Kernel_LOOC array of co-occurrences/occurrences of kernel item in at least one transaction;
- *Algorithm 2:* Building list nLOOC_Tree based on Kernel_COOC array;
- *Algorithm 3:* NOV-RSI algorithm mining all rare significance itemset based on list of nLOOC-Tree.

This paper is organized as follows: in Sect. 2, we describe the basic concepts for mining frequent itemsets, rare itemsets (*the weights/significance of all items are the same or different*) and data structure for datasets. Some theoretical aspects of our approach relies, are given in Sect. 3. Besides, we describe our NOV-RSI algorithm to mine rare significance itemsets based on *Algorithm 1* and *Algorithm 2*. Details of implementation and experiment are discussed in Sect. 4. Finally, we conclude with a summary of our approach, perspectives and extensions of this future work.

2 Background

In this section, we present the basic concepts for mining frequent itemsets, rare itemsets (*the weights/significance of all items are the same or different*) and efficient data structure for dataset.

2.1 Mining Weighted/Significance Frequent, Rare Itemset

Let I = {i_1, i_2, …, i_m} be a set of *m* distinct items. A set of items X = {i_1, i_2, …, i_k}, $\forall i_j \in$ I ($1 \leq j \leq k$) is called an *itemset*, an itemset with *k* items is called a *k-itemset*. $\mathbf{D}$ be a dataset containing *n* transaction, a set of transaction T = {t_1, t_2, …, t_n} and each transaction t_j = {i_{k1}, i_{k2}, …, i_{kl}}, $\forall i_{kl} \in$ I and a set of weight/significance SIG = {sig_{i1}, sig_{i2}, …, sig_{im}}, $\forall sig_{ik} \in [0, 1]$ respective to each item.

Definition 1. The count of an itemset *X* is the number of transaction in which occurs as a subset, denoted *count*(*X*). The support of an itemset *X* computes:

$$\sup(X) = count(X)/\mathrm{n} \tag{1}$$

Definition 2. Let X = {i_1, i_2, …, i_k}, $\forall i_j \in$ I ($1 \leq j \leq k$), significance of itemset X to compute *sig*(X) = *max*(*sig*$_{i1}$, *sig*$_{i2}$, …, *sig*$_{ik}$).

The *significance support* of itemset *X* to computes as follow:

$$\textit{sigsup}(\mathrm{X}) = \textit{sig}(\mathrm{X}) \times \textit{sup}(\mathrm{X}) \tag{2}$$

Definition 3. Let *maxsigsup* be the threshold maximum significance support value specified by user. An itemset *X* is a rare significance itemset if *sigsup*(X) < *maxsigsup*, denoted ***RSI*** is the set of all the rare significance itemset.

See an Example transaction database $\mathcal{D}$ in Tables 1 and 2.

Table 1. The transaction database $\mathcal{D}$ used as our running example

TID	Items							TID	Items						
t1	A	C		E	F			t6					E		
t2	A	C				G		t7	A	B	C		E		
t3				E			H	t8	A		C	D			
t4	A	C	D		F	G		t9	A	B	C		E		G
t5	A	C		E		G		t10	A		C		E	F	G

Table 2. Items significance of $\mathcal{D}$

Item	A	B	C	D	E	F	G	H
Significance	0.55	0.70	0.50	0.65	0.40	0.60	0.30	0.80

Example 1. See Table 1 and 2. There are eight different items I = {A, B, C, D, E, F, G, H} and ten transactions T = {t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}}. And see Table 2 and *maxsigsup* = 0.20. Consider item X = {G}, *sup*(G) = 0.50, *sig*(G) = 0.30, *sigsup*(G) = 0. 15 < *maxsigsup*, we have itemset X = {G}∈ ***RSI***. However, itemset Y = {G, A}, *sup*(GA) = 0.30, *sigsup*(GA) = *max*(sig_G, sig_A) = 0.70, *sigsup*(GA) = *sig*(GA) × *sup*(GA) = 0.70 × 0.50 = 0.35 ≥ *maxsigsup*, we have item G ∉ ***RSI*** (DOES NOT *satisfy the downward closure property*).

Property 1. $\forall i_k \in \mathrm{I}$, *sigsup*($\mathrm{i_k}$) < *maxsigsup*: $i_k \in$ ***RSI***.

2.2 Data Structure for Transaction Database

The binary matrix is an efficient data structure for mining frequent itemsets [3]. The process begins with the transaction database transformed into a binary matrix BiM, in which each row corresponds to a transaction and each column corresponds to an item. Each element in the binary matrix BiM contains 1 if the item is presented in the current transaction; otherwise it contains 0.

3 The Proposed Algorithms

3.1 Generating Array Contain Co-occurrence Items of Kernel Item

In this section, we describe the framework of the algorithm that generates co-occurrence items of items in transaction database.

Definition 4. [3] Project set of item i_k on database $\mathcal{D}$: $\pi(i_k) = \{t_j \in \mathcal{D} \mid i_k \subseteq t_j\}$ is set of transaction contain item i_k. According to Definition 1

$$count(i_k) = |\pi(i_k)| \tag{3}$$

Definition 5. [3] Project set of itemset $X = \{i_1, i_2, \ldots, i_k\}$, $\forall i_j \in I$ $(1 \le j \le k)$: $\pi(X) = \pi(i_1) \cap \pi(i_2) \ldots \pi(i_k)$.

$$count(X) = |\pi(X)| \tag{4}$$

Definition 6. (Reduce search space) Let $\forall i_k \in I$ $(i_1 \succ i_2 \succ \ldots \succ i_m)$ items are ordered in *significance descending*, i_k is called a kernel item. Itemset $X_{lexcooc} \subseteq I$ is called co-occurrence items with the kernel item i_k, as to satisfy $\pi(i_k) \equiv \pi(i_k \cup i_j)$, $i_k \prec i_j$, $\forall i_j \in X_{lexcooc}$. Denoted as $lexcooc(i_k) = X_{lexcooc}$.

Definition 7. (Reduce search space) Let $\forall i_k \in I$ $(i_1 \succ i_2 \succ \ldots \succ i_m)$ items are ordered in *significance descending*, i_k is called a kernel item. Itemset $Y_{lexlooc} \subseteq I$ is called occurrence items with item i_k in as least one transaction, but not co-occurrence items, so that $1 \le |\pi(i_k \cup i_j)| < |\pi(i_k)|$, $\forall i_j \in Y_{lexlooc}$. Denoted as $lexlooc(i_k) = Y_{lexlooc}$.

Algorithm Generating Array of Co-occurrence Items

This algorithm is generating co-occurrence items of items in transaction database and archived into the *Kernel_COOC* array and each element has 4 fields:

- Kernel_COOC[k].*item*: kernel item *k*;
- Kernel_COOC[k].*sup*: support of kernel item *k*;
- Kernel_COOC[k].*cooc*: co-occurrence items with kernel item *k*;
- Kernel_COOC[k].*looc*: occurrence items kernel item *k* in at least one transaction.

The framework of **Algorithm 1** is as follows:

Algorithm 1. Generating Array of Co-occurrence Items

```
Input   : Dataset Ɗ
Output: Kernel_COOC array, matrix BiM
foreach Kernel_COOC[k] do
    Kernel_COOC[k].item = $i_k$
    Kernel_COOC[k].sup = 0
    Kernel_COOC[k].cooc = $2^m - 1$
    Kernel_COOC[k].looc = 0
foreach $t_j \in T$ do
    foreach $i_k \in t_j$ do
        Kernel_COOC[k].sup ++
        Kernel_COOC[k].cooc = Kernel_COOC[k].cooc AND vectorbit($t_j$)
        Kernel_COOC[k].looc = Kernel_COOC[k].looc OR vectorbit($t_j$)
sort Kernel_COOC array in descending by significance
foreach $i_k \in t_j$ do
    Kernel_COOC[k].cooc = lexcooc($i_k$)
    Kernel_COOC[k].looc = lexlooc($i_k$)
```

We illustrate **Algorithm 1** on Example database in Table 1.
Initialization of the Kernel_COOC array, number items in database m = 8;

Item	A	B	C	D	E	F	G	H
sup	0	0	0	0	0	0	0	0
cooc	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
looc	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

Read once of each transaction from t_1 to t_{10}
Transaction t_1 = {A, C, E, F} has vector bit representation **10101100**;

Item	A	B	C	D	E	F	G	H
sup	1	0	1	0	1	1	0	0
cooc	**10101100**	11111111	**10101100**	11111111	**10101100**	**10101100**	11111111	11111111
looc	**10101100**	00000000	**10101100**	00000000	**10101100**	**10101100**	00000000	00000000

The same, transaction t_{10} = {A, C, E, F, G} has vector bit representation **10101110**;

Item	A	B	C	D	E	F	G	H
sup	8	2	8	2	7	3	5	1
cooc	**10100000**	11101000	**10100000**	10110000	**00001000**	**10100100**	**10100010**	00001001
looc	**11111110**	11101010	**11111110**	10110110	**11101111**	**10111110**	**11111110**	00001001

After the processing of **Algorithm 1**, the Kernel_COOC array is as follows (Table 3):

Table 3. Kernel_COOC array are ordered in support ascending order (line 1 to 11)

Item	H	B	D	F	G	E	A	C
sup	0.10	0.20	0.20	0.30	0.50	0.70	0.80	0.80
cooc	E	A, C, E	A, C	A, C	A, C		C	A
looc		G	F, G	D, E, G	B, D, E, F	A, B, C, F, G, H	B, D, E, F, G	B, D, E, F, G

Execute command line 12, 13 and 14 in **Algorithm 1**:

We added the *sig* field to demonstrate items the ordered by *significance descending*. We have *looc*(G) = {B, D, E, F}, where B ≻ D ≻ F ≻ E ≻ G, so *lexlooc*(G) = {∅} and result on Table 4.

Table 4. Kernel_COOC array are co-occurrence items ordered in significance descending

Item	H	B	D	F	A	C	E	G
sig	0.80	0.70	0.65	0.60	0.55	0.50	0.40	0.30
sup	0.10	0.20	0.20	0.30	0.80	0.80	0.70	0.50
cooc	E	A, C, E	A, C	A,C	C	∅	∅	∅
looc	∅	G	F, G	E, G	E, G	E, G	G	∅

3.2 Generating List NLOOC-Tree

In this section, we describe the algorithm generating list *nLOOC-Tree* based on *Kernel_LOOC* array. Each node within the *nLOOC_Tree*, 2 main fields:

- nLOOC_Tree[k].*item*: kernel item *k*;
- nLOOC_Tree[k].*sup*: support of item *k*;

The framework of **Algorithm 2** is as follows:

Algorithm 2. Generating list nLOOC-Tree (*self-reduced search space*)

```
Input  : Kernel_COOC array, matrix BiM
Output: List nLOOC-Tree
foreach Kernel_COOC[k] do
    nLOOC_Tree[k].item = Kernel_COOC[k].item
    nLOOC_Tree[k].sup = Kernel_COOC[k].sup
foreach $i_j \in t_l$ do
    foreach $i_j \in$ Kernel_COOC[k].looc do
        if $i_j \notin$ child node of nLOOC_Tree[k] then
            Add child node $i_j$ to nLOOC_Tree[k]
        else
            Update support of child node $i_j$ on nLOOC_Tree[k]
return list nLOOC_Tree
```

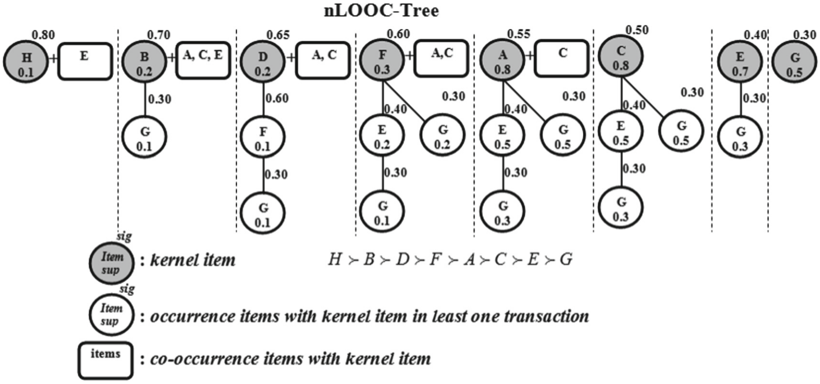


Fig. 1. List nLOOC-Tree based on Kernel_COOC array

Each **nLOOC-Tree** has the characteristics following Fig. 1:

- The height of the tree is less than or equal to the number of items that occur at least in one transaction with the kernel item (*items are ordered in significance support ascending order*).
- Single-path is an *ordered pattern* from the root node (*kernel item*) to the leaf node and the support of the pattern is the support of the leaf node ($i_k \rightarrow i_{k+1} \rightarrow \ldots \rightarrow i_\ell$).
- Sub-single-path is part of *single-path* from the root node to any node in an ordered pattern and the *sub-single-path* support is the support of the child node at the end of the *sub-single-path.*

Example 2. Consider *kernel item F*, we observe **nLOOC-Tree**(F) generating *single-path* $\{\underline{\mathbf{F}} \rightarrow \mathrm{E} \rightarrow \mathrm{G}\}$, *sup*($\underline{\mathrm{F}}$EG) = 0.10 and *sigsup*($\underline{\mathbf{F}}$EG) = 0.06; *sub-single-path* $\{\underline{\mathbf{F}} \rightarrow \mathrm{E}\}$, *sup*($\underline{\mathbf{F}}$E) = 0.20 and *sigsup*($\underline{\mathbf{F}}$E) = *sig*($\underline{\mathbf{F}}$E) × *sup*($\underline{\mathbf{F}}$E) = 0.60 × 0.20 = 0.12.

3.3 Algorithm Generating All Rare Significance Itemsets

In this section, we describe the framework of the algorithm generating all *rare significance itemsets* based on the nLOOC-Tree and Kernel_COOC.

The power set of any itemset X is the set of all subsets of X, including the empty set and X itself, variously denoted as $\wp(X)$*. The set of subsets of X of cardinality greater than or equal to k is sometimes denoted by* $\wp_{\geq k}(X)$*.*

Lemma 1. (*Generating rare significance itemset from co-occurrence items*) $\forall \mathrm{i_k} \in I$, if $sigsup(\mathrm{i_k}) < maxsigsup$ and itemset $X_{lexcooc}$ is set of for all element of $lexcooc(\mathrm{i_k})$ then $sup(\mathrm{i_k} \cup x_{lexcooc}) < maxsigsup$, $\forall x_{lexcooc} \in \wp_{\geq 1}(X_{lexcooc})$ and itemset $\{\mathrm{i_k} \cup x_{lexcooc}\} \in$ **RSI**, $\forall x_{lexcooc} \in \wp_{\geq 1}(X_{lexcooc})$.

Proof. According to Definition 6, (1), (2) and (3): itemset $X_{lexcooc}$ is set of co-occurrence items with the kernel item $\mathrm{i_k}$, as to satisfy $\pi(\mathrm{i_k}) \equiv \pi(\mathrm{i_k} \cup x_{lexcooc})$, $\forall x_{lexcooc} \in \wp_{\geq 1}(X_{lexcooc})$. Therefore, we have $sup(\mathrm{i_k}) = sup\ (\mathrm{i_k} \cup x_{lexcooc})$, $sigsup(\mathrm{i_k}) = sigsup(\mathrm{i_k} \cup x_{lexcooc}) = sig(\mathrm{i_k} \cup X_{lexcooc}) \times sup(\mathrm{i_k}) = sig(\mathrm{i_k}) \times sup(\mathrm{i_k}) < maxsigsup$ and according to Definition 7: itemsets$\{\mathrm{i_k} \cup x_{lexcooc}\} \in$ **RSI**, $\forall x_{lexcooc} \in \wp_{\geq 1}(X_{lexcooc})$■.

Example 3. See Table 4. Consider the item D as kernel item ($maxsigsup = 0.15$), we detect co-occurrence items with the item D as $lexcooc(\mathrm{D}) = \{\mathrm{A, C}\}$ then $\wp_{\geq 1}(\{\mathrm{A, C}\}) = \{\mathrm{A, C, AC}\}$, $sigsup(\underline{\mathrm{D}}\mathrm{A}) = sigsup(\underline{\mathrm{D}}\mathrm{C}) = sigsup(\underline{\mathrm{D}}\mathrm{AC}) = sig(\mathrm{D}) \times sup(\mathrm{D}) = 0.65 \times 0.20 = 0.13 < maxsigsup$ and itemsets $\{DA, DC, DAC\}$are rare significance itemset.

Lemma 2. (*Generating rare significance itemset from occurrence items with kernel item k in at least one transaction*) $\forall \mathrm{i_k} \in I$, $sigsup(\mathrm{i_k}) < maxsigsup$, $X_{lexcooc} = lexcooc(\mathrm{i_k}) \wedge \forall sp_j \in$ nLOOC-Tree($\mathrm{i_k}$), if $sigsup(sp_j) < maxsigsup$ then $\{\mathrm{i_k} \cup ssp_\ell\} \in$ **RSI**, $\forall ssp_\ell \in sp_j$ and $\{\mathrm{i_k} \cup ssp_j \cup x_{lexcooc}\} \in$ **RSI**, $\forall x_{lexcooc} \in \wp_{\geq 1}(X_{lexcooc})$.

Proof. According to Definition 6, 7 and Lemma 1: we have $|\pi(\mathrm{i_k} \cup \mathrm{y_{lexlooc}})| < |\pi(\mathrm{i_k})| \equiv |\pi(\mathrm{i_k} \cup X_{lexcooc})|$, $\mathrm{y_{lexlooc}} \equiv sp_j \in$ nLOOC-Tree($\mathrm{i_k}$) contain of single-paths/sub-single-paths, and $sigsup(\mathrm{i_k} \cup sp_j) < maxsigsup$, $\{\mathrm{i_k} \cup sp_j\} \in$ **RSI**. Therefore, we have $sigsup(\mathrm{i_k} \cup sp_j \cup x_{lexcooc}) < maxsigsup$ and $\{\mathrm{i_k} \cup sp_j \cup X_{lexcooc}\} \in$ **RSI**, $x_{lexcooc} \in \wp_{\geq 1}(X_{lexcooc})$■.

Example 4. See Table 4 and Fig. 1. Consider the item D as kernel item ($maxsigsup = 0.15$) with $sigsup(\mathrm{D}) = 0.13 < maxsigsup$, we detect *occurrence items with kernel item D in at least one transaction* as $Y_{lexlooc} = lexlooc(\mathrm{D}) = \{\mathrm{F, G}\}$; we observe **nLOOC-Tree**(D) generating *single-path* $\{\underline{\mathbf{D}} \rightarrow \mathrm{F} \rightarrow \mathrm{G}\}$, $sup(\underline{\mathbf{D}}\mathrm{FG}) = 0.10$ and $sigsup(\underline{\mathbf{D}}\mathrm{FG}) = 0.65 \times 0.10 = 0.065 < maxsigsup$ then itemsets $\{\underline{D}F, \underline{D}G, \underline{D}FG\}$ are rare significance itemset and itemsets $\{\underline{D}AF, \underline{D}CF, \underline{D}ACF, \underline{D}AG, \underline{D}CG, \underline{D}ACG, \underline{D}AFG, \underline{D}CFG, \underline{D}ACFG\} \in$ **RSI**.

Property 2. $\forall sp_j \in$ nLOOC-Tree($\mathrm{i_k}$) $\wedge\ sig(\mathrm{i_k}) \times minsup\_leafnode(sp_j) \geq maxsigsup$: $\{\mathrm{i_k} \cup sp_j\} \notin$ **RSI** (*minsup_leafnode* is minimum support value of each leaf node on single-paths in nLOOC-Tree($\mathrm{i_k}$)).

The framework of **Algorithm 3** is presented as follows:

Algorithm 3. Generating all rare significance itemsets satisfy *maxsigsup*

```
        Input  : $maxsigsup$, $Kernel\_COOC$ $array$, nLOOC_Tree
        Output: RSI consists all rare significance itemsets
1:      foreach Kernel_COOC[k] do
2:          if $sigsup(Kernel\_COOC[k].item) < maxsigsup$ then
3:              RSI[k] $=\cup\{i_k \cup$ Powerset$(Kernel\_COOC[k].cooc)\}$//lem1
4:              $SSP$ = GenPath(nLOOC_Tree($Kernel\_COOC[k].item$))
5:              RSI[k] $=\cup\{i_k \cup$ Powerset$(Kernel\_COOC[k].cooc) \cup ssp_j\}, \forall\ ssp_j \in SSP$ //lem2
6:          else
7:              if $(Kernel\_COOC[k].looc \neq \emptyset) \vee (sig(i_k) \times min\_sup$(leaf node)$< maxsigsup)$ then
8:                  $SSP$=GenPath(nLOOC_Tree($Kernel\_COOC[k].item$))
9:                  foreach $ssp_j \in SSP$ do
10:                     if $sigsup(i_k \cup ssp_j) < maxsigsup$ then
11:                         RSI[k] $=\cup\{ i_k \cup ssp_j\}$
12:                         RSI[k] $=\cup\{ i_k \cup ssp_j \cup$ Powerset$(Kernel\_COOC[k].cooc)\}$
13:     return RSI
```

3.4 The Algorithm Diagram NOV-RSI

In this section, we represent the diagram of **NOV-RSI** algorithm for high-performance mining *rare significance itemsets*, as follows Fig. 2:

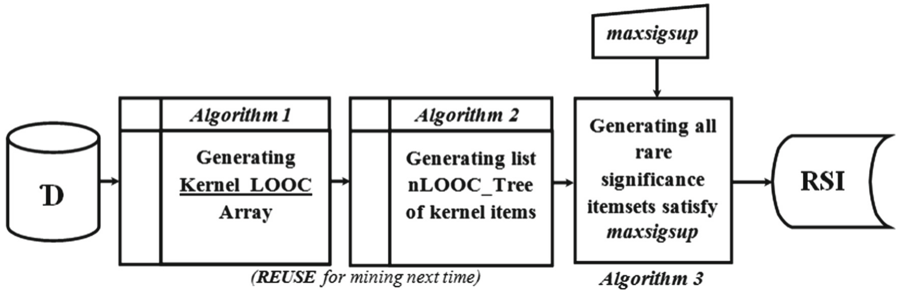

maxsigsup
D
Algorithm 1
Generating Kernel_LOOC Array
Algorithm 2
Generating list nLOOC_Tree of kernel items
Generating all rare significance itemsets satisfy maxsigsup
RSI
(REUSE for mining next time)
Algorithm 3

Fig. 2. The diagram algorithm for NOV-RSI.

We illustrate **Algorithm 3** on Example database in Table 1, 2 and *maxsigsup* = 0.10. After the processing **Algorithm 1** result the *Kernel_COOC* array in Table 4 and **Algorithm 2** presented the list **nLOOC_Tree** in Fig. 1.

Consider *kernel item H*, *sigsup*(H) = 0.80 × 0.10 = 0.08 < *maxsigsup* (*Lemma 1 - line 3*) generating rare significance itemset of *kernel item H* as $\mathrm{RSI}_{[H]}$ = {($\underline{\mathbf{H}}$; 0.08), ($\underline{\mathbf{H}}$E; 0.08)};

Consider *kernel item D*, *sigsup*(D) = 0.65 × 0.20 = 0.13 > *maxsigsup*, *lexcooc*(D) = {A, C} have $\wp_{\geq 1}$({A, C}) = {A, C, AC}. We observe **nLOOC-Tree**(D) have single-path/sub-single-path {$\underline{\mathbf{D}}$ → F → G}, {$\underline{\mathbf{D}}$ → F} and {$\underline{\mathbf{D}}$ → G}: *sigsup*($\underline{\mathbf{D}}$FG) = 0.65 × 0.10 = 0.065 < *maxsigsup*; *sigsup*($\underline{\mathbf{D}}$F) = 0.65 × 0.10 = 0.065 < *maxsigsup*; *sigsup*($\underline{\mathbf{D}}$G) = 0.65 × 0.10 = 0.065 < *maxsigsup* (*Lemma* 2 *- line 5*) generating rare significance itemset of *kernel item D* as $\text{RSI}_{[D]}$ = {($\underline{\mathbf{D}}$FG, 0.065), {($\underline{\mathbf{D}}$F, 0.065), {($\underline{\mathbf{D}}$G, 0.065), ($\underline{\mathbf{D}}$AFG, 0.065), ($\underline{\mathbf{D}}$CFG, 0.065), ($\underline{\mathbf{D}}$ACFG, 0.065), ($\underline{\mathbf{D}}$AF, 0.065), ($\underline{\mathbf{D}}$CF, 0.065), ($\underline{\mathbf{D}}$ACF, 0.065), ($\underline{\mathbf{D}}$AG, 0.065), ($\underline{\mathbf{D}}$CG, 0.065), ($\underline{\mathbf{D}}$ACG, 0.065)};

Consider *kernel item B*, *sigsup*(B) = 0.70 × 0.20 = 0.14 > *maxsigsup*, *lexcooc*(B) = {A, C, E} have $\wp_{\geq 1}$({A, C, E}) = {A, C, E, AC, AE, CE, ACE}. We observe **nLOOC-Tree**(B) have single-path {$\underline{\mathbf{B}}$ → G}: *sigsup*($\underline{\mathbf{B}}$G) = 0.70 × 0.10 = 0.07 < *maxsigsup* (*Lemma* 2 *- line 5*) generating rare significance itemset of *kernel item B* as $\text{RSI}_{[B]}$ = {($\underline{\mathbf{B}}$G, 0.07), ($\underline{\mathbf{B}}$AG, 0.07), ($\underline{\mathbf{B}}$CG, 0.07), ($\underline{\mathbf{B}}$EG, 0.07), ($\underline{\mathbf{B}}$ACG, 0.07), ($\underline{\mathbf{B}}$AEG, 0.07), ($\underline{\mathbf{B}}$CEG, 0.07), ($\underline{\mathbf{B}}$ACEG, 0.07)};

Consider *kernel item F*, *sigsup*(F) = 0.60 × 0.30 = 0.18 > *maxsigsup*, *lexcooc*(F) = {A, C} have $\wp_{\geq 1}$({A, C}) = {A, C, AC}. We observe **nLOOC-Tree**(F) have single-path/sub-single-path {$\underline{\mathbf{F}}$ → E → G}, {$\underline{\mathbf{F}}$ → E} and {$\underline{\mathbf{F}}$ → G}: *sigsup*($\underline{\mathbf{F}}$EG) = 0.60 × 0.10 = 0.06 < *maxsigsup*; *sigsup*($\underline{\mathbf{F}}$E) = 0.60 × 0.20 = 0.12 > *maxsigsup*; *sigsup*($\underline{\mathbf{F}}$G) = 0.60 × 0.20 = 0.12 > *maxsigsup* generating rare significance itemset of *kernel item F* as $\text{RSI}_{[F]}$ = {($\underline{\mathbf{F}}$EG, 0.06), ($\underline{\mathbf{F}}$AEG, 0.06), ($\underline{\mathbf{F}}$CEG, 0.06), ($\underline{\mathbf{F}}$ACEG, 0.06)} (*Lemma* 2 *- line 5*);

Consider *kernel item E*, *sigsup*(E) = 0.40 × 0.70 = 0.28 > *maxsigsup*. We observe **nLOOC-Tree**(E) have single-path {$\underline{\mathbf{E}}$ → G}, *minsup_leafnode*({$\underline{\mathbf{E}}$ → G}) = 0.30 and sig(E) × *minsup_leafnode*({$\underline{\mathbf{E}}$ → G}) = 0.40×0.30 = 0.12 > *maxsigsup*, so $\text{RSI}_{[E]}$ = {∅} (*Property 5 - line 7*).

Consider *kernel item C* (similarly *kernel item E*), *sigsup*(C) = 0.50 × 0.80 = 0.40 ≥ *maxsigsup*. We observe **nLOOC-Tree**(C) have single-paths {$\underline{\mathbf{C}}$ → E → G}, {$\underline{\mathbf{C}}$ → G}, *minsup_leafnode*({$\underline{\mathbf{C}}$ → E → G}, {$\underline{\mathbf{C}}$ → G}) = 0.30 and sig(C) × *minsup_leafnode*{$\underline{\mathbf{C}}$ → E → G}, {$\underline{\mathbf{C}}$ → G}) = 0.50 × 0.30 = 0.15 > *maxsigsup*, so $\text{RSI}_{[C]}$ = {∅} (*Pro* 2 *- line 7*).

Consider *kernel item A* (similarly *kernel item C*), *sigsup*(A) = 0.55 × 0.80 = 0.44 ≥ *maxsigsup,* $\text{RSI}_{[A]}$ = {∅}.

Table 5 shows the rare significance itemsets at *maxsigsup = 0.10.*

Table 5. RSI satisfy *maxsigsup* = 0.10 (Example database in Table 1 and 2)

Kernel item	Rare significance itemsets – RSI (#RSI = 26)			
H	(**H**; 0.08)	(**H**E; 0.08)		
B	(**B**G; 0.07)	(**B**AG; 0.07)	(**B**CG; 0.07)	(**B**EG; 0.07)
	(**B**ACG; 0.07)	(**B**AEG; 0.07)	(**B**CEG; 0.07)	(**B**ACEG; 0.07)
D	(**D**F; 0.065)	(**D**G; 0.065)	(**D**AF; 0.065)	(**D**CF; 0.065)
	(**D**AG; 0.065)	(**D**CG; 0.065)	(**D**ACF; 0.065)	(**D**ACG; 0.065)
	(**D**FG; 0.065)	(**D**AFG; 0.065)	(**D**CFG; 0.065)	(**D**ACFG; 0.065)
F	(**F**EG; 0.06)	(**F**AEG; 0.06)	(**F**CEG; 0.06)	(**F**ACEG; 0.06)

4 Experiments

All experiments were performed on a PC with a Core Duo CPU T2500 2.0 GHz, 4 Gb main memory, running Microsoft Windows 7 Ultimate. All codes were compiled using C#, MVStudio 2010, .Net Framework 4.

We experimented on two instance types of datasets, see Table 6:

- Two real datasets are both *dense* form of UCI Machine Learning Repository [http://archive.ics.uci.edu/ml] as **Chess** and **Mushroom** datasets.
- Two synthetic *sparse* datasets are generated by software of IBM Almaden Research Center [http://www.almaden.ibm.com] as **T10I4D100K** and **T40I10D100K** datasets.

Table 6. Datasets description in experiments

Name	#Trans	#Items	#Avg. Length	Type	Density (%)
Chess	3,196	75	37	Dense	49.3
Mushroom	8,142	119	23	Dense	19.3
T10I4D100K	100,000	870	10	Sparse	1.1
T40I10D100K	100,000	942	40	Sparse	4.2

Additionally, we build one table to save the significance values of items by random real values in the range of 0 *to* 1. This is the first proposed algorithm for RSI mining based on approach DOES NOT *satisfy the downward closure property*. To evaluate the performance of the proposed algorithm, we modified (DOES NOT *satisfy the downward closure property*) the **AprioriInverse** [4] and **Rarity** [6] to mine RSI called the **WaprioriInverse** and **WRarity** algorithm. Therefore, we have compared the NOV-RSI algorithm with algorithms **WAprioriInverse** and **WRarity**.

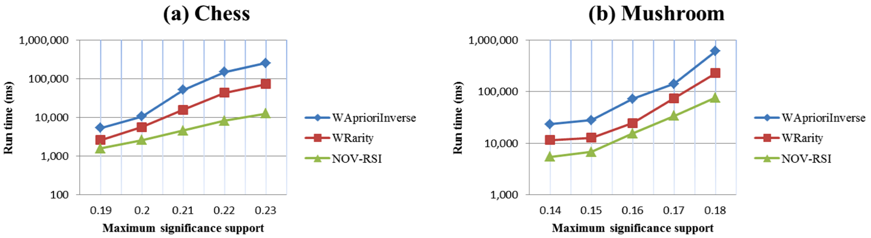


Fig. 3. Running time of the three algorithms on **Chess** and **Mushroom** datasets.

Figure 3(a) and (b) show the running time of the compared algorithms on real datasets *Chess* and *Mushroom*. The NOV-RSI algorithm runs faster than WAprioriInverse and WRarity algorithms in all maximum significance supports.

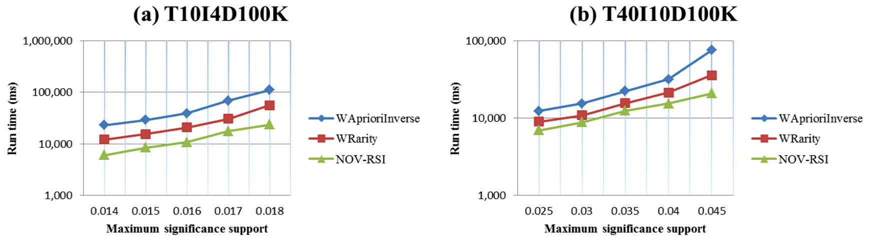


Fig. 4. Running time of the three algorithms on **T10I4D100K** and **T40I10D100K** datasets.

Figure 4(a) and (b) show the running time of the compared algorithms on synthetic datasets *T10I4KD100K* and *T40I10D100K*. The NOV-RSI algorithm runs faster than WaprioriInverse and WRarity algorithms.

In the experiment mentioned above, results suggest the following comparison of these algorithms when running time is concerned: NOV-RSI runs faster than algorithms WaprioriInverse and WRarity algorithms in all *maxsigsup* on real and synthetic datasets.

5 Conclusion

According to this paper, we presented a high-performance algorithm for mining rare significance itemsets on transaction databases, comprising three phases: *the first phase*, we quickly detect a Kernel_COOC array of co-occurrences and occurrences of kernel item in at least one transaction; *the second phase*, we build the list of nLOOC-Tree base on the Kernel_COOC and a binary matrix of dataset (*self-reduced search space*); *the last phase*, the algorithm is proposed for fast mining RSI based on nLOOC-Tree. Besides, when using mining RSI with *other maxsigsup value*, the proposed algorithm only performs mining RSI based on the nLOOC-Tree that is calculated previously (*the second phase* - *Algorithm* 2), there by reducing the significant processing time. The experiment's results show that the proposed algorithms perform better than other existing algorithms.

The results from the algorithm proposed: In the future, we will expand the NOV-RSI algorithm to be able to mine *rare significance itemsets* on Multi-Cores, Many-CPUs, GPU and distributed computing systems such as Hadoop, Spark.

Acknowledgements. This work was supported by the following institutions VNUHCM-University of Social Sciences and Humanities; VNUHCM-University of Science, Vietnam National University, Ho Chi Minh City, Vietnam.

References

1. Agrawal, R., Imilienski, T., Swami, A.: Mining association rules between sets of large databases. In: ACM SIGMOD International Conference on Management of Data, pp. 207–216 (1993)
2. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: a frequent pattern tree approach. Data Min. Knowl. Disc. **8**(1), 53–87 (2004)
3. Phan, H., Le, B.: A novel algorithm for frequent itemsets mining in transactional databases. In: Ganji, M., Rashidi, L., Fung, B.C.M., Wang, C. (eds.) PAKDD 2018. LNCS (LNAI), vol. 11154, pp. 243–255. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04503-6_25
4. Koh, Y.S., Rountree, N.: Finding sporadic rules using Apriori-Inverse. In: Ho, T.B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 97–106. Springer, Heidelberg (2005). https://doi.org/10.1007/11430919_13
5. Troiano, L., Birtolo, C.: A fast algorithm for mining rare itemsets. In: IEEE 19th International Conference on Intelligent Systems Design and Applications, pp. 1149–1155 (2009)
6. Szathmary, L., Valtchev, P., Napoli, A., Godin, R.: Efficient vertical mining of mRI. In: 19th International Conference on Concept Lattices and Their Applications, pp. 269–280 (2012)
7. Lan, G.C., Hong, T.P., Lee, H.Y., Lin, C.W.: Tightening upper bounds for mining weighted frequent itemsets. Intell. Data Anal. **19**(2), 413–429 (2015)
8. Kiran, R.U., Kotni, A., Reddy, P.K., Toyoda, M., Bhall, S., Kitsuregawa, M.: Efficient discovery of weighted frequent itemsets in very large transactional databases: a re-visit. In: Proceedings of the IEEE International Conference on Big Data (Big Data), pp. 723–732 (2018)
9. Yun, U., Shin, H., Ryu, K.H., Yoon, E.: An efficient mining algorithm for maximal weighted frequent patterns in transactional databases. Knowl.-Based Syst. **33**, 53–64 (2012)
10. Huai, Z., Huang, M.: A weighted frequent itemsets Incremental Updating Algorithm base on hash Table. In: 3rd International Conference on Communication Software and Networks (ICCSN), pp. 201–204. IEEE (2011)

MSPP: A Highly Efficient and Scalable Algorithm for Mining Similar Pairs of Points

Subrata Saha[1], Ahmed Soliman[2], and Sanguthevar Rajasekaran[2](✉)

[1] Healthcare and Life Sciences Division, IBM Research, Yorktown Heights, NY 10598, USA
subrata.saha@uconn.edu

[2] Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269, USA
{ahmed.soliman,sanguthevar.rajasekaran}@uconn.edu

Abstract. The closest pair of points problem or closest pair problem (CPP) is an important problem in computational geometry where we have to find a pair of points from a set of points in a metric space with the smallest distance between them. This problem arises in a number of applications, such as but not limited to clustering, graph partitioning, image processing, patterns identification, and intrusion detection. Numerous algorithms have been presented for solving the CPP. The algorithms that are employed in practice have a worst case quadratic run time complexity. In this article, we present an elegant approximation algorithm for the CPP called "MSPP: **M**ining **S**imilar **P**airs of **P**oints." It is faster than currently best known algorithms while maintaining a very good accuracy. The proposed algorithm also detects a set of closely similar pairs of points in Euclidean and Pearson's metric spaces, and can be adapted in numerous real world applications, such as clustering, dimension reduction, constructing and analyzing gene/transcript co-expression network, intrusion detection, and so forth.

Keywords: Closest Pair Problem (CPP) · Mining Similar Pairs of Points (MSPP) · Time Series Motif Mining (TSMM)

1 Introduction

Given a set of n points in any metric space, the problem of finding the closest pair of points is known as the Closest Pair Problem (CPP) and has been well studied. Rabin [13] proposed a randomized algorithm with an expected run time of $\mathcal{O}(n)$ where the expectation is in the space of all possible outcomes of coin flips made in the algorithm. Rabin's algorithm used the floor function as a basic operation. In 1979, Fortune and Hopcroft [5] presented a deterministic algorithm with a run time of $\mathcal{O}(n \log\log n)$ assuming that the floor operation takes $\mathcal{O}(1)$ time. Both of these algorithms assume a $\mathcal{O}(1)$ dimensional space. The run times of these

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 30–37, 2020.
https://doi.org/10.1007/978-3-030-65390-3_3

algorithms have an exponential dependency on the dimension. Other classical algorithms include [7,12]. Yao [16] has proven a lower bound of $\Omega(n \log n)$ on the algebraic decision tree model for a space of any dimension. This lower bound holds under the assumption that the floor function is not allowed.

Time Series Motif Mining (TSMM) is a crucial problem that can be thought of as CPP in a large dimensional space. Mueen *et al.* have presented an elegant exact algorithm called Mueen-Keogh (MK) for TSMM [11]. MK improves the performance of the brute-force algorithm with a novel application of the triangular inequality. A number of probabilistic as well as approximate algorithms are also known for solving this problem (e.g. [1,3,6,9,10,14,15]). Cai *et al.* [2] proposed a deterministic algorithm to solve the TSMM problem, called JUMP, which outperforms existing $\mathcal{O}\left(n^2\right)$ methods by a factor of up to 100. This was done by skipping unnecessary comparisons and multiplication operations. Although this algorithm performs well in practice, especially as the length of the timeseries n increases, the speed-up is dependent on the skipping fraction. Li and Lin [8] have recently presented a novel algorithm called LL to solve the TSMM problem in an expected $\mathcal{O}(n)$ time. The idea is to build and maintain a data structure called *grids*. Although the LL algorithm runs in an expected linear time, in practice, the performance significantly drops if the time series has a large number of very close pairs of points.

MSPP is an out-of-core algorithm. So, it can work with a very large dataset with a very small memory footprint. To start with the input is in the disk. There are two basic steps in the algorithm and in each step, we do one pass through the input data. In each pass, the algorithm incrementally retrieves information embedded in the dataset and after the final pass, it outputs a set of similar pairs of points. We claim that a pair of points in this set is the closest pair with a very high probability. Our algorithm works for a variety of metrics including Euclidean and Pearson's metric spaces. Next, we illustrate our proposed algorithm MSPP.

2 Methods

Our algorithm works by discretizing the continuous attribute values of the points. If two points are very similar in the Euclidean space, then they also can be expected to be very similar when we transform the points using a subset of the attributes. In the first pass, we discretize the attribute values and detect the highly similar pair of points in the transformed space. In the second pass, we compute the Euclidean distances (or, Pearson's coefficient) between every pair of points found in the first pass and output a set of similar pairs of points. The details of our algorithm are provided next.

First Pass. Let the input points be p_i where $1 \leq i \leq m$. Each input point has n attributes and let the attributes be a_j, $1 \leq j \leq n$. We assume that the points are given in column-major order in the disk. Specifically, the input points are stored as an $n \times m$ matrix M where each column i of M corresponds to a point and each row j of M corresponds to an attribute. In the first pass, we retrieve each row j of M at a time. There are two basic steps involved in the first pass.

Discretizing and Encoding Attribute Values. As stated above, row j of M has the values of the attribute a_j for all the input points. There are n iterations, one for each row of M. In iteration j we retrieve row j (i.e. a line in the file) from the dataset residing in the disk where $1 \leq j \leq n$. Let v_i^j be the value of the attribute a_j for the point p_i where $1 \leq j \leq n$ and $1 \leq i \leq m$. In iteration j we cluster the set of values v_i^j of the attribute a_j into k disjoint clusters (k being user defined). We employ `k-means++` clustering algorithm to perform this task because of its simplicity and expected linear time complexity. It initializes the cluster centers before proceeding with the standard k-means optimization iterations. With the `k-means++` initialization, the algorithm is guaranteed to find a solution that is $\mathcal{O}(\log k)$ competitive to the optimal k-means solution. After clustering, each value v_i^j of the attribute a_j falls into one of the k clusters. We encode each v_i^j with k binary bits. Only one bit will be turned "on" out of the k bits. We can think of a bit as a binary variable, having two possible values called "true" and "false" where "on" bit contains "true" value and the rest of the $k-1$ bits contain "false" values. In this scenario, the "on" bit corresponds to that cluster where the particular attribute value v_i^j of a point p_i falls into.

Mining Similar Pairs of Points in the Binary Space. Next, we randomly sample a subset of coordinates in the encoded binary space and hash the points based on this subset. Two points will be hashed into the same bucket if they have the same values for the randomly chosen coordinates. If two points fall into the same bucket in the hash table, this is a candidate pair. We keep a priority queue Q that stores the best r pairs that have been encountered thus far (r being user defined). The key used for any pair in Q will be the Hamming distance between them (across all the coordinates). The Hamming distance between each candidate pair will be computed and inserted into Q if this Hamming distance is less than the largest key in Q. We repeat this process of sampling and hashing t times (for some suitable value of t). In each stage of sampling the candidate pairs generated are used to update Q. A similar pair of points in terms of Hamming distance may not necessarily be similar in the original Euclidean space. This is the reason why we keep a priority queue Q.

Second Pass. In the second pass we compute the Euclidean distance or Pearson's correlation coefficient between every pair of points found in Q and output a set of s (s being user defined) best pairs. Please note that the original dataset always resides in the disk.

3 Results and Discussions

3.1 Datasets

We have employed both real and synthetic datasets in our experiments. Real datasets were taken from both biomedical and data mining domains. Synthetic datasets were created by randomly generating varying numbers of points and attributes. Next, we provide the details about the datasets.

Real Datasets. To demonstrate the effectiveness of MSPP, we have used 6 real microarray gene expression datasets. Each row of a gene expression dataset corresponds to an individual where each column represents the expression of a particular gene across the individuals. Consequently, in our experiment each gene is synonymous with a point and its expressions from different individuals correspond to the distinct attribute values. More details about the datasets can be found in [17]. In addition, another experiment has been carried out to evaluate the performance of MSPP algorithm by employing "individual household electric power consumption timeseries data" [4].

Synthetic Datasets. To perform rigorous simulations, we have generated numerous synthetic datasets by varying the number of points as well as attributes. To mimic the real world scenario, values of a particular attribute are randomly generated using Gaussian distribution having mean 0 and standard deviation 1.

3.2 Evaluation Metrics

We measure the effectiveness of our proposed algorithm MSPP using 4 different metrics. These metrics are defined below.

1. **A-Rank.** A-Rank means average rank. We have computed the average rank of the top pairs of points detected by algorithm of interest over 5 runs.
2. **Accuracy.** Fractions of the pairs of points correctly identified in the top 50 and top 100 pairs of points.
3. **Speed-up.** Measures the improvement in execution time of MSPP with respect to other similar algorithms of interest where both the algorithms perform the same task in an identical environment.
4. **Time.** Measures elapsed time using total number of CPU clock cycles consumed by each of the algorithms of interest.

3.3 Outcomes

Real Datasets. We have performed rigorous experimental evaluations to test the scalability, efficiency, and effectiveness of MSPP. As described above, our algorithm MSPP has been run on 6 real gene expression microarray datasets as shown in Table 1. Both CPU times and accuracy have been used as performance metrics for our evaluation. Since our algorithm MSPP may not always output the closest pair, we wanted to check the quality of output from our algorithm. To measure this quality, we have used the brute force algorithm to identify the top pairs (the closest, the second closest, the third closest, etc.). We used these outputs to identify the rank of the best output from MSPP. We have also computed the average rank over 5 runs. The results are reported in Table 1. For all of the cases, MSPP finds the closest pair. Please note that TSMM algorithms work on time-series data and detect only closest pair of points.

MSPP not only detects the closest pair of points but also outputs a user defined number of closely similar pairs of points. We have demonstrated it by

Table 1. Benchmark for MSPP algorithm along with accuracy on real datasets.

Dataset	Name	Points	Attributes	CPU time (s)		Accuracy		
				Brute-force	MSPP	A-Rank	Top-50	Top-100
D1.1	Colon Tumor	2,000	60	**0.49**	0.85	1	1.00	1.00
D1.2	Central Nervous System	7,129	60	20.15	**8.86**	1	1.00	1.00
D1.3	Leukemia	7,129	72	23.70	**10.20**	1	1.00	1.00
D1.4	Breast Cancer	24,481	97	199.75	**6.61**	1	1.00	1.00
D1.5	Mixed Lineage Leukemia	12,582	72	56.41	**38.37**	1	1.00	1.00
D1.6	Lung Cancer	12,600	203	138.79	**96.08**	1	1.00	1.00

observing the top 50 and top 100 closest pairs of points from both the brute force and MSPP algorithms. It is evident from Table 1 that MSPP identified all the top 50 and top 100 closest pairs in all the datasets. The improvement of performance in terms of execution time of MSPP becomes more significant for larger datasets such as in D1.4. Please, see Table 1 and Fig. 1(d) for runtime comparisons. In each experiment, 10 bits were used to encode each attribute value. MSPP picked 20 coordinates randomly in each of 5 stages of sampling.

Now, consider the time series dataset. The experiments were done in a similar fashion as stated above. On each run, a number of points were randomly picked from the timeseries and then, the MSPP and brute-force algorithms were executed on these points. Each point has a length of $1,000$ attributes. Results of this experiment are summarized in Table 2. For a visual comparison please see, Fig. 1(f). It is to be noted that a log scale has been used for the y-axis.

Table 2. Benchmark for MSPP on randomly picked points with $1,000$ attributes from the entire space of points from the timeseries data.

Points	Average CPU time (s)		Accuracy		
	Brute-force	MSPP	A-Rank	Top-50	Top-100
5,000	137.87	**7.52**	1	1.00	1.00
10,000	323.66	**15.59**	1	1.00	1.00
15,000	627.74	**16.74**	1	1.00	1.00
20,000	973.73	**25.73**	1	1.00	0.99
25,000	1,636.02	**29.93**	1	0.94	0.93
30,000	2,510.25	**35.19**	1	0.85	0.84

Synthetic Datasets. A set of experiments has been carried out using randomly generated data (as pointed out earlier) to evaluate the performance of MSPP. To study the effect of input sizes and attributes on the performance, a total of 20 datasets have been generated with varying numbers of points and attributes. The execution times to identify the closely similar pairs of points from these datasets are shown in Fig. 1(a). It is evident that the run time almost linearly increases

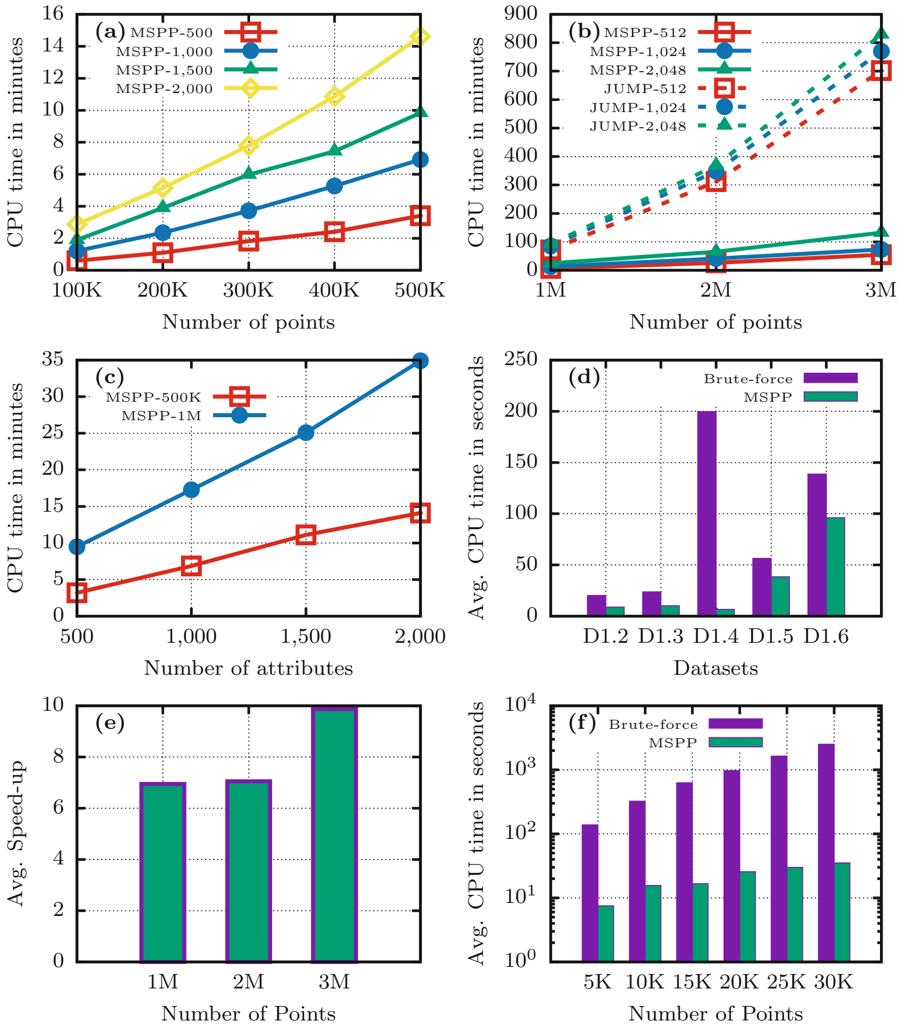


Fig. 1. Performance evaluations of MSPP algorithm: (a) CPU time consumed by MSPP with respect to varied points and dimensions. (b) CPU time consumed by MSPP and JUMP on large datasets. (c) CPU time consumed by MSPP on large dimensions. (d) CPU time consumed by MSPP and Brute-force. (e) Avg. Speed-up achieved by MSPP over JUMP on large datasets. (f) CPU time consumed by MSPP and Brute-force on randomly picked points from a timeseries.

with the number of points. To further study the effect of large dimensions on our algorithm, another set of datasets has been generated by varying the number of dimensions. In these datasets the input sizes are fixed at 500,000 and 100,000 points while the dimensions (number of attributes) are increased from 500 to 2,000 in steps of 500 increment. The execution times are illustrated in Fig. 1(c).

This set of experiments reveals the linear relationship between the execution time of MSPP and the number of dimensions.

Time series motif mining (TSMM) is a crucial problem that can be thought of as a special case of the CPP in a large dimensional space as described earlier. Since very efficient algorithms exist (e.g. [2,8]) in the literature for solving the TSMM problem, we have further conducted experiments to investigate their performances on simulated points generated by employing Gaussian distribution as stated earlier. This time the number of attributes has been fixed at 512, $1,024$ and $2,048$. Three large datasets have been generated. The execution times have been plotted in Fig. 1(b) while Fig. 1(e) shows the average speed-up ratios. These comparisons reveal that MSPP is faster than JUMP. For example, MSPP is almost 10× faster than JUMP on the dataset containing 3 million points. Please note that we also tried to include the execution times of the LL algorithm in this comparative study. However, we observed that the LL algorithm requires more than 70 h to run on $1,000,000$ points with $1,000$ attributes. So, we are not reporting the runtimes of LL. In each experiment for the synthetic datasets, 2 bits were used to encode each attribute value. MSPP picked 20 coordinates randomly in each stage of sampling and the number of stages was 5. It is to be noted that any time series motif mining algorithm will only identify the closest pair from the given set of points. On the contrary, MSPP identifies a user defined number of similar pairs of points. In the above experiments, MSPP outputs 500,000 similar pairs of points containing the closest pair with a high probability. In this respect, a direct comparison may not be appropriate between MSPP and any other time series motif miner.

4 Conclusions

In this article, we have proposed an efficient, reliable, and scalable algorithm called MSPP to detect a set of highly similar pairs of points in both of Euclidean and Pearson's metric spaces. It is an out-of-core algorithm and thus, it can work on a large high dimensional dataset. MSPP consumes less amount of physical memory. Experimental evaluations show that the algorithm is indeed effective and efficient in terms of both accuracy and execution time. MSPP can be used in a diverse set of practical applications, such as but not limited to time series motif mining, clustering, gene co-expression network, feature reduction in high dimensional space, and 2-locus problem in genome-wide association study.

References

1. Beaudoin, P., Coros, S., van de Panne, M., Poulin, P.: Motion-motif graphs. In: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 117–126 (2008)
2. Cai, X., Zhou, S., Rajasekaran, S.: JUMP: a fast deterministic algorithm to find the closest pair of subsequences. In: Proceedings of the 2018 SIAM International Conference on Data Mining, pp. 73–80. Society for Industrial and Applied Mathematics (May 2018). https://doi.org/10.1137/1.9781611975321.9. https://epubs.siam.org/doi/abs/10.1137/1.9781611975321.9

3. Chiu, B., Keogh, E., Lonardi, S.: Probabilistic discovery of time series motifs. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2003, pp. 493–498. ACM Press (2003). https://doi.org/10.1145/956750.956808
4. Dua, D., Graff, C.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml
5. Fortune, S., Hopcroft, J.: A note on Rabin's nearest-neighbor algorithm. Inf. Process. Lett. **8**(1), 20–23 (1979). https://doi.org/10.1016/0020-0190(79)90085-1
6. Guyet, T., Garbay, C., Dojat, M.: Knowledge construction from time series data using a collaborative exploration system. J. Biomed. Inf. **40**(6), 672–687 (2007). https://doi.org/10.1016/j.jbi.2007.09.006. https://linkinghub.elsevier.com/retrieve/pii/S1532046407001050
7. Khuller, S., Matias, Y.: A simple randomized sieve algorithm for the closest-pair problem. Inf. Comput. **118**(1), 34–37 (1995). https://doi.org/10.1006/inco.1995.1049
8. Li, X., Lin, J.: Linear time motif discovery in time series. In: Proceedings of the 2019 SIAM International Conference on Data Mining, pp. 136–144. Society for Industrial and Applied Mathematics (May 2019). https://doi.org/10.1137/1.9781611975673.16. https://epubs.siam.org/doi/abs/10.1137/1.9781611975673.16
9. Meng, J., Yuan, J., Hans, M., Wu, Y.: Mining motifs from human motion. In: Eurographics 2008 Short Papers, pp. 1–4 (2008)
10. Minnen, D., Isbell, C.L., Essa, I., Starner, T.: Discovering multivariate motifs using subsequence density estimation and greedy mixture learning. Proc. Nat. Conf. Artif. Intell. **1**, 615–620 (2007)
11. Mueen, A., Keogh, E., Zhu, Q., Cash, S., Westover, B.: Exact discovery of time series motifs. In: Proceedings of the 2009 SIAM International Conference on Data Mining, pp. 473–484. Society for Industrial and Applied Mathematics (September 2009). https://doi.org/10.1137/1.9781611972795.41. https://doi.org/10.1137/1.9781611972795.41
12. Preparata, F.P., Shamos, M.I.: Computational Geometry: An Introduction. Springer, New York (1985). https://doi.org/10.1007/978-1-4612-1098-6
13. Rabin, M.O.: Probabilistic algorithms. In: Traub, J.F. (ed.) Algorithms and Complexity: New Directions and Recent Results, pp. 21–39. Academic Press, New York (1976)
14. Rombo, S., Terracina, G.: Discovering representative models in large time series databases. In: Christiansen, H., Hacid, M.-S., Andreasen, T., Larsen, H.L. (eds.) FQAS 2004. LNCS (LNAI), vol. 3055, pp. 84–97. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-25957-2_8
15. Tanaka, Y., Iwamoto, K., Uehara, K.: Discovery of time-series motif from multi-dimensional data based on MDL principle. Mach. Learn. **58**(2–3), 269–300 (2005). https://doi.org/10.1007/s10994-005-5829-2
16. Yao, A.C.: Lower bounds for algebraic computation trees with integer inputs. In: 30th Annual Symposium on Foundations of Computer Science, pp. 308–313. IEEE (1989). https://doi.org/10.1109/SFCS.1989.63495. http://ieeexplore.ieee.org/document/63495/
17. Zhu, Z., Ong, Y.S., Dash, M.: Markov blanket-embedded genetic algorithm for gene selection. Pattern Recogn. **40**(11), 3236–3248 (2007). https://doi.org/10.1016/j.patcog.2007.02.007. https://linkinghub.elsevier.com/retrieve/pii/S0031320307000945

Discovering High Utility Itemsets Using Set-Based Particle Swarm Optimization

Wei Song(✉) and Junya Li

School of Information Science and Technology, North China University of Technology,
Beijing 100144, China
songwei@ncut.edu.cn

Abstract. Mining high utility itemsets (HUIs) is a hot research topic in data mining. Algorithms based on evolutionary computation are attracting increasing attention because they have the advantage of avoiding the combinatorial explosion of the HUI search space. Among evolutionary methods used for mining HUIs, particle swarm optimization (PSO) is the most popular. Existing PSO-based HUI mining (HUIM) algorithms transform positions according to the result of applying the sigmoid function to the velocity. In this paper, we propose an HUIM algorithm based on set-based PSO (S-PSO) called HUIM-SPSO, which mainly considers elements in positions whose velocities are high. We introduce the modeling of HUIM using S-PSO, and explain HUIM-SPSO in detail. To reflect the diversity of the mining results, we propose the measure of the bit edit distance. Extensive experimental results show that the HUIM-SPSO algorithm is efficient and can discover more HUIs with a high degree of diversity.

Keywords: Data mining · High utility itemset · Set-based particle swarm optimization · Bit edit distance

1 Introduction

High utility itemset mining (HUIM) [9, 12] is an extension of frequent itemset mining that is used to discover high-profit itemsets by considering both the quantity and value of a single item. Most existing HUIM algorithms are designed to discover all high utility itemsets (HUIs). Because of the combinatorial explosion incurred by the items in the search space of all HUIs, the performance of exact algorithms tends to quickly degrade with the size of the database and becomes unacceple for large databases. Furthermore, for application fields such as recommender systems, it is not necessary to use all HUIs [14].

Several HUIM approaches have been proposed that use evolutionary algorithms (EAs), such as genetic algorithms (GAs) [3], particle swarm optimization (PSO) [5, 6], and artificial bee colony (ABC) [10]. Although these algorithms can discover reasonable itemsets in an acceptable time, determining how to identify more HUIs within a limited number of iterations is challenging.

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 38–53, 2020.
https://doi.org/10.1007/978-3-030-65390-3_4

PSO is the most widely used EA in previous works [5, 6]; we also consider the HUIM problem from the perspective of PSO. Different from the binary coding scheme PSO [4] used for HUIM, set-based PSO (S-PSO) proposed by Chen et al. [1] is exploited in our algorithm. For S-PSO, the position is updated according to the structure of the cut set, which maintains positions with high speed.

Using S-PSO, we propose an efficient HUIM algorithm called HUIM-SPSO. First, we redefine the S-PSO operations for the HUIM problem. Then, we describe the proposed algorithm in detail. To show the superiority of S-PSO for the HUIM problem, we define the bit edit distance to measure the diversity of the mining results. The experimental results show that HUIM-SPSO is not only efficient but can also discover more HUIs with a high degree of diversity.

2 Preliminaries

2.1 Problem of HUIM

Let $\boldsymbol{I} = \{i_1, i_2, \ldots, i_m\}$ be a finite set of items. $X \subseteq \boldsymbol{I}$ is called an *itemset*. Let $\boldsymbol{D} = \{T_1, T_2, \ldots, T_n\}$ be a transaction database. Each transaction $T_i \in \boldsymbol{D}$, with unique identifier *tid*, is a subset of $\boldsymbol{I}$.

The *internal utility* $q(i_p, T_d)$ represents the quantity of item i_p in transaction T_d. The *external utility* $p(i_p)$ is the unit profit value of item i_p. The *utility* of item i_p in transaction T_d is defined as $u(i_p, T_d) = p(i_p) \times q(i_p, T_d)$. The utility of itemset X in transaction T_d is defined as $u(X, T_d) = \sum_{i_p \in X \wedge X \subseteq T_d} u(i_p, T_d)$. The utility of itemset X in $\boldsymbol{D}$ is defined as $u(X) = \sum_{X \subseteq T_d \wedge T_d \in D} u(X, T_d)$. The transaction utility of transaction T_d is defined as $TU(T_d) = u(T_d, T_d)$. To mine HUIs, the *minimum utility threshold* δ, which is specified by the user, is defined as a percentage of the total TU values of the database, whereas the *minimum utility value* is defined as $min\_util = \delta \times \sum_{T_d \in D} TU(T_d)$. An itemset X is called an HUI if $u(X) \geq min\_util$. Given a transaction database $\boldsymbol{D}$, the task of HUIM is to determine all itemsets that have utilities no less than *min_util*.

The *transaction-weighted utilization* (TWU) of itemset X is the sum of the transaction utilities of all the transactions containing X, which is defined as $TWU(X) = \sum_{X \subseteq T_d \wedge T_d \in D} TU(T_d)$ [9]. X is a high TWU itemset (HTWUI) if $TWU(X) \geq min\_util$. An HTWUI with k items is called a k-HTWUI. It is proved in [9] that HTWUIs satisfy the transaction-weighted downward closure property, and all HUIs are HTWUIs.

Consider the transaction database in Table 1 and the profit table in Table 2. For convenience, we write an itemset $\{C, E\}$ as CE. In the example database, the utility of item E in transaction T_2 is $u(E, T_2) = 3 \times 1 = 3$, the utility of itemset CE in transaction T_2 is $u(CE, T_2) = u(C, T_2) + u(E, T_2) = 6 + 3 = 9$, and the utility of itemset CE in the transaction database is $u(CE) = u(CE, T_2) + u(CE, T_4) = 24$. Given $min\_util = 35$, as $u(CE) < min\_util$, CE is not an HUI. The TU of T_2 is $TU(T_2) = u(ABCDE, T_2) = 18$, and the utilities of other transactions are shown in the third column of Table 1. The TWU of itemset CE is $TWU(CE) = TU(T_2) + TU(T_4) = 48$; thus, CE is an HTWUI.

Table 1. Example database

TID	Transactions	TU
T_1	(*B*, 1), (*C*, 3), (*D*, 5)	35
T_2	(*A*, 4), (*B*, 1), (*C*, 1), (*D*, 1), (*E*, 1)	18
T_3	(*A*, 4), (*C*, 2), (*D*, 5)	31
T_4	(*C*, 2), (*D*, 5), (*E*, 1)	30
T_5	(*A*, 5), (*B*, 2), (*D*, 5), (*E*, 3)	33
T_6	(*A*, 3), (*B*, 1), (*C*, 1), (*D*, 1)	14
T_7	(*D*, 1), (*E*, 1), (*F*, 2)	8

Table 2. Profit table

Item	*A*	*B*	*C*	*D*	*E*	*F*
Profit	1	2	6	3	3	1

2.2 Basic Idea of S-PSO

PSO is an EA that imitates the social behavior of bird flocking and fish schooling [4]. In the original PSO algorithm, several particles are initialized at random. Each particle moves toward the optimal value according to the following two equations:

$$Vec_i(t+1) = w \times Vec_i(t) + c_1 \times r_1 \times (PBest_i - Pos_i(t)) + c_2 \times r_2 \times (GBest - Pos_i(t)), \tag{1}$$

$$Pos_i(t+1) = Pos_i(t) + Vec_i(t+1), \tag{2}$$

where $Vec_i(t)$ and $Vec_i(t+1)$ are the velocities of the ith particle at iterations t and $t+1$, respectively; $Pos_i(t)$ and $Pos_i(t+1)$ are the positions of the ith particle at iterations t and $t+1$, respectively; $PBest_i$ is the previous best position of the ith particle; $GBest$ is the current best position of all particles; the three constants w, c_1, and c_2 are weighting coefficients; and r_1 and r_2 are random numbers in the range (0, 1).

All particles update their velocities and positions repeatedly until the best solution is found or the maximum number of iterations is reached.

Velocity Update of S-PSO. In S-PSO, a velocity is a set of probabilities that specify the possibility of each element being used in the position update [1]. Let E be a crisp set of all possible solutions. Vec defined on E is given by

$$Vec = \{p(e)\,|e \in E\}. \tag{3}$$

Using the velocity definition in Eq. 3, the update of the velocity in Eq. 1 is implemented by the following four types of calculation [1].

(1) Coefficient × Velocity: Eq. 1 is composed of three parts. The first part belongs to this type and can be defined as follows:

$$c \times Vec = \{p^*(e)\,|e \in E\}, \tag{4}$$

where

$$p^*(e) = \begin{cases} 1, \text{ if } c \times p(e) > 1 \\ c \times p(e), \text{ otherwise} \end{cases}. \tag{5}$$

(2) Position − Position: In Eq. 1, the second and third parts consist of calculating the differences between the local best position and the current position, and the global best position and the current position. Let Pos_A and Pos_B be two positions. The calculation of the position difference is defined as

$$Pos_A - Pos_B = \{e|e \in Pos_A \wedge e \notin Pos_B\}. \tag{6}$$

(3) Coefficient × (Position − Position): After subtracting the current position from both the local best position and global best position, the calculation of the last two parts is categorized into this type. Similarly, this can also be defined as

$$cE' = \{p'(e)\,|e \in E\}, \tag{7}$$

where

$$p'(e) = \begin{cases} 1, \text{ if } e \in E' \text{ and } c > 1 \\ c, \text{ if } e \in E' \text{ and } 0 \leq c \leq 1 \\ 0, \text{ if } e \notin E' \end{cases}. \tag{8}$$

(4) Sum of the set of probabilities: The final calculation is the sum of three velocities. Let $Vec_1 = \{p_1(e) \mid e \in E\}$, $Vec_2 = \{p_2(e) \mid e \in E\}$, and $Vec_3 = \{p_3(e) \mid e \in E\}$ be three sets of probabilities defined on E, where the sum of Vec_1, Vec_2, and Vec_3 is defined as

$$Vec_1 + Vec_2 + Vec_3 = \{max(p_1(e), p_2(e), p_3(e))\,|e \in E\}. \tag{9}$$

Position Update of S-PSO. After the velocity update, a random number $\alpha \in (0, 1)$ is generated for each particle. For the jth element of Vec_i, if its corresponding probability $p(e)$ is not smaller than α, then $p(e)$ is retained, that is

$$cut_\alpha(Vec_i^j) = \begin{cases} p(e), \text{ if } p(e) = Vec_i^j \wedge p(e) \geq \alpha \\ 0, \qquad \text{otherwise} \end{cases}. \tag{10}$$

The set of velocity Vec_i, with each element computed by Eq. 10, is called its *cut set*. With the cut set and previous position, the position update in S-PSO is also calculated using Eq. 2.

3 Existing Algorithms

Early HUIM algorithms follow a two-phase routine. The first phase identifies candidate HUIs using the TWU model. Then, the second phase filters the actual HUIs by scanning the original database [9, 12]. Two-phase algorithms typically generate many candidates, which leads to a huge search space and high computational cost. Therefore, one-phase algorithms without candidates were introduced. For one-phase algorithms, data structures, such as a utility list [8] and chain of accurate utility list [7], are used for the efficient discovery of HUIs.

Recently, EAs have been used to traverse immense candidate itemset spaces within an acceptable time to mine HUIs. Two HUIM algorithms, HUPE$_{\text{UMU}}$-GARM and HUPE$_{\text{WUMU}}$-GARM, based on the GA, were proposed in [3]. The difference between them is that the second algorithm does not require a minimum utility threshold. The main problem of these two algorithms is that they tend to fall into local optima easily.

Using ant colony optimization, Wu et al. proposed an HUIM algorithm that generates a routing graph before all the ants start their tours [13]. An ant might generate several candidate itemsets during a tour. Therefore, each node in the routing graph represents a specific itemset that can be evaluated to determine if it is an HUI.

Song and Huang studied the HUIM problem from the perspective of the ABC algorithm [10]. The proposed HUIM-ABC discovers HUIs by modeling the itemsets as nectar sources. For each nectar source, three types of bees are used for sequential optimization iteratively.

Among various EAs, PSO is the most widely used algorithm in HUIM. HUIM-BPSO$_{\text{sig}}$ [5] and HUIM-BPSO [6] are two PSO-based algorithms for mining HUIs. HUIM-BPSO outperforms HUIM-BPSO$_{\text{sig}}$ using an OR/NOR-tree structure.

In addition to the velocity and previous position used by the binary coding scheme PSO in existing algorithms, the cut set is also used in S-PSO to update positions. Thus, elements corresponding to high velocities tend to have more chances to be retained in the next iteration, which may generate results with high diversity. To the best of our knowledge, S-PSO has not been used in HUIM.

4 Modeling HUIM Using S-PSO

In this paper, each particle is represented by two types of vectors that represent the velocity and position.

Definition 1. Let *SN* be the population size, N_c be the number of 1-HTWUIs, and all 1-HTWUIs be sorted in a total order (e.g., lexicographic order) during the entire mining process. A *velocity vector* V_i $(1 \leq i \leq SN)$ is a vector with N_c elements, and each element V_i^j is a probability, corresponding to the velocity of the *j*th 1-HTWUI, to be used in the position update; and a position vector P_i $(1 \leq i \leq SN)$ is a binary vector with N_c elements, and each element P_i^j is either zero or one, which indicates whether the *j*th 1-HTWUI is absent or present in P_i, respectively.

For these two vectors, a velocity vector changes according to the previous positions, and a position vector then changes with respect to the velocity vector and represents a new candidate itemset. Specifically, if the *j*th position of a position vector contains a one, the item in the *j*th position, according to the total order, is present in a potential HUI; otherwise, this item is not included and cannot be in a potential HUI. For a position vector P_i, its *j*th bit is initialized by either zero or one using roulette wheel selection with the probability

$$p(P_i^j) = \frac{TWU(item_j)}{\sum_{k=1}^{N_c} TWU(item_k)}, \tag{11}$$

where N_c is the number of 1-HTWUIs.

Within each iteration of S-PSO, the fitness function is calculated to characterize the optimization problem. Let X be an itemset represented by a position vector P_i. The utility of X is used as the fitness function directly:

$$fitness(P_i) \ = u(X). \tag{12}$$

We also need to redefine the calculation of (*Position – Position*). Let P_a and P_b be two position vectors with N_c elements. We define

$$dP = P_a - P_b = \ \{dP_i |\, 1 \le i \le N_c\}, \tag{13}$$

where

$$dP_i = \begin{cases} 1, & \text{if } P_a^i = 1 \text{ and } P_b^i = 0 \\ 0, & \text{otherwise} \end{cases}. \tag{14}$$

5 HUIM-SPSO Algorithm

5.1 Bitmap Item Information Representation

We use a bitmap, which is an effective representation of item information in HUIM algorithms [11], in HUIM-SPSO. Specifically, itemsets are represented by a *bitmap cover*. In a bitmap cover, there is one bit for each transaction in the database. If item *i* appears in transaction T_j, then bit *j* of the bitmap cover for item *i* is set to one; otherwise, the bit is set to zero. This naturally extends to itemsets. Let *X* be an itemset. *Bit*(*X*) corresponds to the bitmap cover that represents the transaction set for the itemset *X*. Let *X* and *Y* be two itemsets. *Bit*(*X* ∪ *Y*) can be computed as *Bit*(*X*) ∩ *Bit*(*Y*), that is, the bitwise-AND of *Bit*(*X*) and *Bit*(*Y*). Thus, the utility values of the target itemsets can be calculated efficiently using bitwise operations.

5.2 Proposed Algorithm

According to the above discussion, the proposed HUIM algorithm based on S-PSO (HUIM-SPSO) is shown in Algorithm 1.

```
Algorithm 1   HUIM-SPSO
Input         Population size SN, minimum utility value min_util, maximum number of iterations max_iter
Output        HUIs
 1   Init( );
 2   iter = 1;
 3   while iter ≤ max_iter do
 4     for i=1 to SN do
 5       Randomly generate $r_1$ and $r_2$;
 6       Calculate $V_i$ using Eq. 1;
 7       Randomly generate $\alpha$;
 8       for j=1 to $N_c$ do
 9         if $V_i^j < \alpha$ then
10           $V_i^j = 0$;
11         end if
12       end for
13       Position_update($P_i$);
14       X = IS($P_i$);
15       if u(X) ≥ min_util and $X \notin SHUI$ then
16         $X \rightarrow SHUI$;
17       end if
18       $X_l = IS(Pbest_i)$;
19       if $u(X) > u(X_l)$ then
20         $Pbest_i = P_i$;
21       end if
22     end for
23     Find GBest among SN particles;
24     iter ++;
25   end while
26   Output all HUIs in SHUI.
```

In Algorithm 1, the initialization procedure (described in Algorithm 2) is called in Step 1. Then, the number of iterations is set to one (Step 2). The main loop (Steps 3–25) repeats the update of the velocity and position vectors until the maximum number of iterations is reached. The loop from Step 4 to Step 22 processes each particle individually. For each particle P_{i}, Step 5 generates two random numbers in the range (0, 1). It should be noted that we set values of w, c_1, and c_2 in Eq. 1 to one in our algorithm. The velocity vector is calculated in Step 6. Step 7 generates a random number α to modify the velocity vector. Then, the velocity vector V_i of the enumerating particle is updated in the loop from Steps 8–12. The position is updated by calling the procedure described in Algorithm 3. Step 14 determines the itemset that corresponds to the enumerating particle. The function *IS*() returns itemset X by unifying the items in P_i if its value is one. The newly determined itemset is stored in *SHUI* if it is an HUI and has not been discovered before. *SHUI* is the set of discovered HUIs. Steps 18–21 update the local best value of the enumerating particle. *GBest* is updated by the particle corresponding to the discovered HUI with the highest utility value in Step 23. Step 24 increments the number of iterations by one. Finally, Step 26 outputs the discovered HUIs.

```
Algorithm 2   Procedure Init( )
Input         Transaction database D, population size SN, minimum utility value
              min_util
Output        The first population of particles
 1            Scan database D once;
 2            Delete items that are not 1-HTWUIs;
 3            Represent the reorganized database as a bitmap;
 4            for $i$=1 to $SN$ do
 5              for $j$=1 to $N_c$ do
 6                Initialize $P_i^j$ with 0 or 1 using Eq. 11;
 7                if $P_i^j \neq 0$ then
 8                  $v_i^j = rand( )$;
 9                end if
10              end for
11              $PBest_i = P_i$;
12              $X = IS(P_i)$;
13              if $u(X) \geq min\_util$ then
14                $X \rightarrow SHUI$;
15              end if
16            end for
17            Find $GBest$ among $SN$ particles.
```

In Algorithm 2, the transaction database is first scanned once to determine the 1-HTWUIs (Steps 1–2). In Step 3, the bitmap representation of the pruned database is constructed. The main loop (Steps 4–16) generates the initial particles individually. The loop from Step 5 to Step 10 initializes each element of the position and velocity vectors of the enumerating particle. The function *rand*() returns a random number in the range (0, 1). Note that we only initialize the elements whose values are one in the position vector with a random velocity. P_i is also initialized as $PBest_i$ in Step 11. Step 12 determines the itemset that corresponds to the enumerating particle. If the current particle can produce an HUI *X* (Step 13), Step 14 records this itemset. Finally, *GBest* is initialized in Step 17.

```
Algorithm 3   Position_update(P)
Input         Position vector P
Output        New updated position vector P
1   Initialize new_P with all elements equals to 0;
2   Randomly generate a positive number k no higher than $N_c$;
3   if $|V| \geq k$ then
4       Generate new_P by setting k 1s within the positions whose corresponding velocities are not 0;
5   else
6       Generate new_P by setting |V| 1s in the positions whose corresponding velocities are not 0;
7       Change values of (k–|V|)-bits of new_P from 0s to 1s;
8   end if
9   P = new_P;
```

In Algorithm 3, the processed position is initialized as a vector with N_c 0 s in Step 1. Then, the new position is built in a constructive manner. Step 2 determines the number of bits to be set to one. The new position first learns from the elements in the corresponding velocity (Steps 3–4). $|V_i|$ is the number of elements whose values are not zeros in V_i. If the construction of *new_P* is not finished by only considering non-zero elements in its velocity vector, other 1-HTWUIs are used to build a new position vector (Steps 5–8). The updated new position is finally determined in Step 9.

5.3 Illustrative Example

We use the transaction database in Table 1 and profit table in Table 2 for the explanation. After the first database scan, the TWU of each item is shown in Table 3.

Table 3. TWU of each item

Item	*A*	*B*	*C*	*D*	*E*	*F*
TWU	96	100	128	169	89	8

Given $min\_util = 35$, as $TWU(F) < min\_util$, item F is deleted from transactions T_7, and the utility of F is eliminated from the TUs of T_7. The reorganized database is then represented by a bitmap, as shown in Table 4.

Assume the size of population *SN* to be 3. As the number of 1-HTWUIs is 5, there are five elements in both the velocity vector and position vector. According to Eq. 11, three position vectors are generated randomly: $P_1 = <10111>$, $P_2 = <11001>$, and $P_3 = <11010>$. Then, three velocities are also generated randomly: $V_1 = \{0.52, 0, 0.87, 0.01, 0.15\}$, $V_2 = \{0.15, 0.58, 0, 0, 0.62\}$, and $V_3 = \{0.76, 0.03, 0, 0.28, 0\}$. For the first population, $PBest_i$ is the same as P_i. Thus, $PBest_1 = <10111>$, $PBest_2 = <11001>$, and $PBest_3 = <11010>$. For the example database, we can see that the three particles represent *ACDE*, *ABE*, and *ABD*, respectively. We have $u(ACDE) = 16$, $u(ABE) = 27$,

Table 4. Bitmap representation of the reorganized database

	A	B	C	D	E
T'_1	0	1	1	1	0
T'_2	1	1	1	1	1
T'_3	1	0	1	1	0
T'_4	0	0	1	1	1
T'_5	1	1	0	1	1
T'_6	1	1	1	1	0
T'_7	0	0	0	1	1

and $u(ABD) = 41$. Among these three itemsets, only ABD is an HUI, so $SHUI = \{ABD\text{: } 41\}$, where the number after the colon denotes the utility. According to the utility value, $GBest$ is $<11010>$.

We then take particle P_1 as an example. According to Eq. 13, $PBest_1 - P_1 = <00000>$ and $GBest - P_1 = <01000>$. Suppose $r_1 = 0.15$ and $r_2 = 0.66$, according to Eq. 7, $r_1(PBest_1 - P_1) = \{0, 0, 0, 0, 0\}$ and $r_2(GBest - P_1) = \{0, 0.66, 0, 0, 0\}$. Then, using Eq. 9, we have the new velocity: $V_1 = \{0.52, 0, 0.87, 0.01, 0.15\} + \{0, 0, 0, 0, 0\} + \{0, 0.66, 0, 0, 0\} = \{0.52, 0.66, 0.87, 0.01, 0.15\}$. Let the randomly generated α be 0.04. For the current V_1, only the fourth element is lower than α, so V_1 is changed to $\{0.52, 0.66, 0.87, 0, 0.15\}$. Then k is randomly generated as one, which indicates that there is only one bit with value one in the new position. Because the first, second, third, and the last elements of V_1 are non-zero, only one of these four bits may be set to one in the new position. Suppose the first bit is selected randomly, the new position vector is $P_1 = <10000>$, which represents itemset A. Because $u(A) = 16 < min\_util$, A is not an HUI. Furthermore, because $u(A) = u(ACDE)$ and $u(A) < u(ABD)$, neither $PBest_1$ nor $GBest$ changes.

In the same iteration, the second and third particles are processed similarly. Then the next iteration starts to process each particle to discover HUIs until the maximal number of iterations is reached.

6 Performance Evaluation

We evaluate the performance of our HUIM-SPSO algorithm and compare it with the HUIM-BPSO$_{\text{sig}}$ [5] and HUIM-BPSO [6] algorithms. We downloaded the source code of the two comparison algorithms from the SPMF data mining library [2].

6.1 Test Environment and Datasets

The experiments were performed on a computer with a 4-core 3.20 GHz CPU and 4 GB memory running 64-bit Microsoft Windows 7. Our programs were written in Java. Four real datasets were used to evaluate the performance of the algorithms. The characteristics of the datasets are presented in Table 5.

Table 5. Characteristics of the datasets used for the experimental evaluations

Datasets	Avg. trans. length	No. of items	No. of trans
Chess	37	76	3,196
Mushroom	23	119	8,124
Accidents_10%	34	469	34,018
Connect	43	130	67,557

The four datasets were also downloaded from the SPMF data mining library [2]. The Chess and Connect datasets originate from game steps. The Mushroom dataset contains various species of mushrooms and their characteristics. The Accident dataset is composed of (anonymized) traffic accident data. Similar to the work of Lin et al. [5, 6], only 10% of the total dataset was used for the experiments.

For all experiments, the termination criterion was set to 10,000 iterations and the population size was set to 20.

6.2 Runtime

First, we demonstrate the efficiency performance of these algorithms. When measuring the runtime, we varied the minimum utility threshold for each dataset.

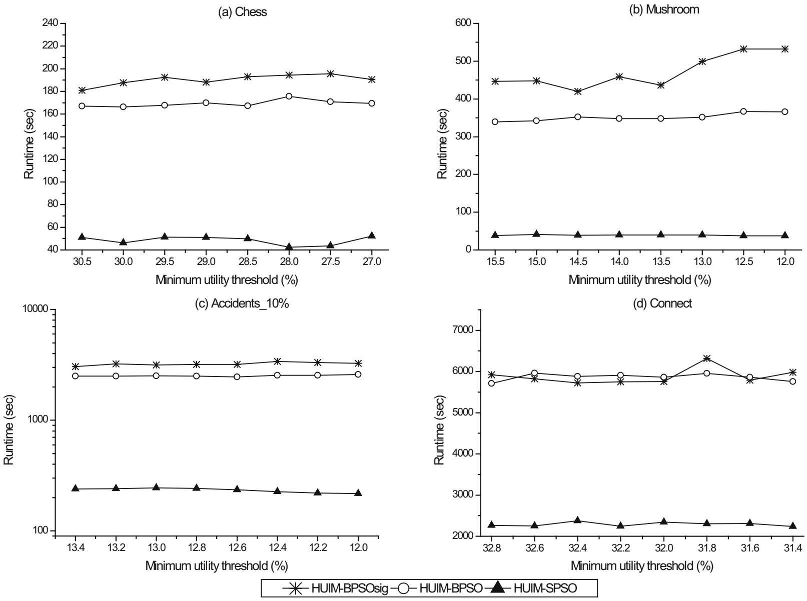


Fig. 1. Execution times for the four datasets

Figure 1(a) compares the execution times for the Chess dataset. We can see that HUIM-SPSO was the most efficient among the three PSO-based HUIM algorithms. On average, HUIM-SPSO was 2.95 times faster than HUIM-BPSO$_{sig}$ and 2.52 times faster than HUIM-BPSO.

For the comparison results on the Mushroom dataset shown in Fig. 1(b), the superiority of the efficiency of the proposed HUIM-SPSO was more obvious. When the minimum threshold changed from 15.5% to 12.0%, HUIM-SPSO always had a steady runtime of approximately 39 s. HUIM-SPSO was one order of magnitude faster than HUIM-BPSO$_{sig}$ and 8.01 times faster than HUIM-BPSO, on average.

From Fig. 1(c), we can see that when the minimum utility threshold changed from 13.4% to 12.0%, HUIM-SPSO was always one order of magnitude faster than HUIM-BPSO$_{sig}$. When the minimum utility threshold was lower than 12.4%, HUIM-SPSO was also one order of magnitude faster than HUIM-BPSO.

When the minimum threshold changed from 32.8% to 31.4% for the Connect dataset, both HUIM-BPSO$_{sig}$ and HUIM-BPSO took more than 5,700 s, and the gap between them was very small, as shown in Fig. 1(d). The proposed HUIM-SPSO algorithm was faster than the above two algorithms, with a runtime constantly close to 2,250 s.

6.3 Number of Discovered HUIs

Because bio-inspired HUIM algorithms cannot ensure the discovery of all itemsets within a certain number of cycles, we compared the number of discovered HUIs among the three PSO-based algorithms.

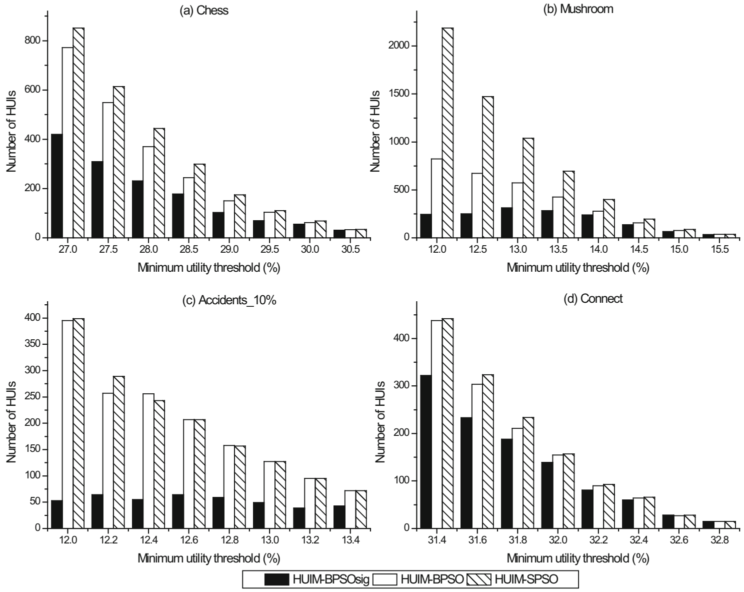


Fig. 2. Number of discovered HUIs for the four datasets

Figure 2 illustrates the number of discovered HUIs for the three algorithms. For all four datasets, the proposed HUIM-SPSO always discovered more HUIs than the other two PSO-based HUIM algorithms, except when the minimum threshold was 12.4% for the Accidents_10% dataset. Generally, the superiority of the number of results of HUIM-SPSO became more obvious as the minimum utility threshold decreased. The maximal gap between HUIM-SPSO and the two counter algorithms appeared for the Mushroom dataset.

6.4 Diversity Comparison

Different from typical optimization problems, such as the traveling salesman problem, which has relatively few best values, all the itemsets with utilities no lower than the minimum threshold are the targets of HUIM. Because the distribution of HUIs is not even, diversity of the population of particles is essential; that is, the higher the diversity within one population, the higher the chance that more results will be discovered within fewer iterations.

To measure the degree of diversity, we propose the *bit edit distance* based on the edit distance used in the field of natural language processing. The edit distance is the minimum number of editing operations – including insertion, deletion, or substitution – needed to transform one string into the other. The bit edit distance (BED) is defined as follows:

$$BED(P, P_t) = NBits, \tag{15}$$

where *NBits* is the number of bitwise-complement operations transformed from position vector P to P_t.

We can see from Eq. 15 that the higher the value of $BED(P, P_t)$, the higher the level of diversity between P and P_t. For example, if we transform $P = <11010>$ to $P_t = <10111>$, three bitwise-complement operations are needed; that is, transform the second bit from 1 to 0, transform the third bit from 0 to 1, and transform the last bit from 0 to 1. Thus, $BED(P, P_t) = 3$.

To measure the diversity of the entire population, we use the pair of position vectors with the highest degree of diversity and average degree of diversity of all pairs of position vectors. Thus, the maximal bit edit distance and average bit edit distance are defined. Let $P_1, P_2, \ldots, P_{SN}$ be position vectors in one population. The maximal bit edit distance (Max_BED) is defined as

$$Max\_BED = max\{BED\big(P_i, P_j\big) | \, 1 \le i \le SN, \; 1 \le j \le SN, i \ne j\}. \tag{16}$$

The average bit edit distance (Ave_BED) is defined as

$$Ave\_BED = \frac{\sum_i \sum_{j \ne i} BED(P_i, P_j)}{SN \times (SN - 1)}. \tag{17}$$

Tables 6, 7, 8 and 9 show the comparison results of bit edit distances for all four datasets for different numbers of iterations. Generally, the HUIs discovered by HUIM-SPSO had a higher Max_BED and Ave_BED than those discovered by HUIM-BPSO for

Table 6. The bit edit distances for the Chess dataset with $\delta = 28.5\%$

Number of iterations	HUIM-BPSO$_{sig}$		HUIM-BPSO		HUIM-SPSO	
	Ave_BED	*Max_BED*	*Ave_BED*	*Max_BED*	*Ave_BED*	*Max_BED*
2000	9.2	18	8.65	14	13.6	31
4000	8.64	14	8.41	15	13.97	32
6000	8.22	13	6.77	11	14.94	36
8000	8.96	16	5.91	12	15.71	36
10000	9.19	16	7.86	15	15.72	36

Table 7. The bit edit distances for the Mushroom dataset with $\delta = 15.5\%$

Number of iterations	HUIM-BPSO$_{sig}$		HUIM-BPSO		HUIM-SPSO	
	Ave_BED	*Max_BED*	*Ave_BED*	*Max_BED*	*Ave_BED*	*Max_BED*
2000	11.83	23	7.05	12	10.52	20
4000	11.21	22	8.37	14	10.44	19
6000	11.30	22	7.81	14	10.46	22
8000	10.06	21	7.58	12	10.96	22
10000	10.14	17	7.62	13	10.18	21

all four datasets. For the other comparison algorithm, HUIM-BPSO$_{sig}$, the mining results of HUIM-SPSO still had a higher level of diversity on the Chess and Connect datasets, and a comparable level of diversity on the Mushroom and Accidents_10% datasets. For the Mushroom and Accidents_10% datasets, the diversity of the HUIs discovered by HUIM-SPSO was better than that of the HUIs discovered by HUIM-BPSO$_{sig}$ when the iteration number was high.

Table 8. The bit edit distances for the Accidents_10% dataset with $\delta = 13.0\%$

Number of iterations	HUIM-BPSO$_{sig}$		HUIM-BPSO		HUIM-SPSO	
	Ave_BED	*Max_BED*	*Ave_BED*	*Max_BED*	*Ave_BED*	*Max_BED*
2000	16.95	35	7.09	11	8.6	25
4000	15.98	34	7.80	13	10.19	26
6000	14.44	34	7.33	12	11.13	24
8000	10.50	17	7.29	12	11.13	22
10000	10.83	18	7.62	13	10.84	23

Table 9. The bit edit distances for the Connect dataset with $\delta = 31.6\%$

Number of iterations	HUIM-BPSO$_{\text{sig}}$		HUIM-BPSO		HUIM-SPSO	
	Ave_BED	*Max_BED*	*Ave_BED*	*Max_BED*	*Ave_BED*	*Max_BED*
2000	7.87	13	6.49	12	17	35
4000	7.85	15	7.82	13	15.53	37
6000	8.39	14	7.33	14	13.07	33
8000	8.31	15	7.55	14	14.64	35
10000	7.44	13	7.74	14	15.19	36

Different from the above two sets of experiments, the diversity measure of HUIM-BPSO$_{\text{sig}}$ was better than that of HUIM-BPSO. This is because HUIM-BPSO uses an OR/NOR-tree structure to save valid combinations to discover HUIs. Thus, the prefix structure makes itemsets in the same branch have higher similarity.

7 Conclusions

In this paper, a new HUIM algorithm called HUIM-SPSO was proposed based on S-PSO. In contrast to typical PSO, S-PSO tends to change elements of positions with high velocity rather than simply resorting to sigmoid function transformation. The modeling process of HUIM using S-PSO was described. To measure the diversity of the discovered results, the bit edit distance was proposed. The experimental results demonstrated that the proposed algorithm was efficient and effective.

Acknowledgments. This work was partially supported by the National Natural Science Foundation of China (61977001), the Great Wall Scholar Program (CIT&TCD20190305), and Beijing Urban Governance Research Center.

References

1. Chen, W.-N., Zhang, J., Chung, H.S.H., Zhong, W.-L., Wu, W.-G., Shi, Y.-H.: A novel set-based particle swarm optimization method for discrete optimization problems. IEEE T. Evolut. Comput. **14**(2), 278–300 (2010)
2. Fournier-Viger, P., et al.: The SPMF open-source data mining library version 2. In: Berendt, B., et al. (eds.) ECML PKDD 2016. LNCS, vol. 9853, pp. 36–40. Springer, Cham (2016)
3. Kannimuthu, S., Premalatha, K.: Discovery of high utility itemsets using genetic algorithm with ranked mutation. Appl. Artif. Intell. **28**(4), 337–359 (2014)
4. Kennedy, J., Eberhart, R.: A discrete binary version of particle swarm algorithm. In: Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics, pp. 4104–4108 (1997)
5. Lin, J.C.-W., et al.: Mining high-utility itemsets based on particle swarm optimization. Eng. Appl. Artif. Intel. **55**, 320–330 (2016)

6. Lin, J.C.-W., Yang, L., Fournier-Viger, P., Hong, T.-P., Voznak, M.: A binary PSO approach to mine high-utility itemsets. Soft. Comput. **21**(17), 5103–5121 (2017)
7. Liu, J., Wang, K., Fung, B.C.M.: Direct discovery of high utility itemsets without candidate generation. In: Proceedings of The 12th IEEE International Conference on Data Mining, pp. 984–989 (2012)
8. Liu, M., Qu, J.-F.: Mining high utility itemsets without candidate generation. In: Proceedings of The 21st ACM International Conference on Information and Knowledge Management, pp. 55–64 (2012)
9. Liu, Y., Liao, W., Choudhary, A.: A two-phase algorithm for fast discovery of high utility itemsets. In: Ho, T.B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 689–695. Springer, Heidelberg (2005). https://doi.org/10.1007/11430919_79
10. Song, W., Huang, C.: Discovering high utility itemsets based on the artificial bee colony algorithm. In: Phung, D., Tseng, V., Webb, G.I., Ho, B., Ganji, M., Rashidi, L. (eds.) PAKDD 2018. LNCS (LNAI), vol. 10939, pp. 3–14. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93040-4_1
11. Song, W., Liu, Y., Li, J.: Vertical mining for high utility itemsets. In: Proceedings of the 2012 IEEE International Conference on Granular Computing, pp. 429–434 (2012)
12. Song, W., Zhang, Z., Li, J.: A high utility itemset mining algorithm based on subsume index. Knowl. Inform. Syst. **49**(1), 315–340 (2015). https://doi.org/10.1007/s10115-015-0900-1
13. Wu, J.M.T., Zhan, J., Lin, J.C.W.: An ACO-based approach to mine high-utility itemsets. Knowl. Based Syst. **116**, 102–113 (2017)
14. Yang, R., Xu, M., Jones, P., Samatova, N.: Real time utility-based recommendation for revenue optimization via an adaptive online top-k high utility itemsets mining model. In: Proceedings of The 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, pp. 1859–1866 (2017)

SS-AOE: Subspace Based Classification Framework for Avoiding Over-Confidence Errors

Xinqi Wang[1,2(✉)], Dan Yu[1,2], and Guanjun Lai[1,2]

[1] Dalian Neusoft University of Information, Dalian, Liaoning, China
jt_wangxinqi@neusoft.edu.cn
[2] Dalian Neusoft Education Technology Group Co. Limited, Dalian, Liaoning, China

Abstract. Misclassification with high confidence can cause great harm, especially in high-risk tasks. This is because when faced with unfamiliar samples which outside the distribution of known samples, classifiers are prone to give overly high prediction confidence. In this paper, we propose a classification framework with effective confidence estimation based on a reduced dimensional space (subspace). Intuitively, our method is designed to give low confidence to "unfamiliar" or out-of-domain test samples, so to "know what we do not know". The effectiveness of this method is supported by a real-world Stroke diagnosis data set. We use multiple metrics for evaluation, including Brier score, calibration curve, and expected calibration error (ECE).

Keywords: Confidence estimation · Out-of-distribution misclassification · Subspace methods

1 Introduction

Classification is a typical approach in machine learning. In practical applications, we often use the average generalization accuracy of the prediction model over the entire sample space. Nevertheless, in high-risk areas, such as fault detection, medical diagnosis, and financial evaluation, users are concerned with how likely the AI system is to predict correctly for a new sample. Thus, users can derive the reliability of the prediction and then assess the risk level of the prediction error [1].

The confidence of prediction can also be used as an AI system's decision-making mechanism. We assume that familiar samples to be drawn from the same distribution as the training samples, while unfamiliar samples are those out of distribution. If a model is able to know that it does not know with respect to predict with a certainty, or know its prediction is most probably incorrect, it can report a rough prediction and remind the decision maker with a confidence reference.

Most probabilistic classification models provide posterior probabilities of the categories with given test samples, such as the Soft-max output of neural networks. Unfortunately, those predictions typically have poor ability to detect its own false predictions on unfamiliar samples or on samples that do not belong to any classes, referred as "*it does*

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 54–66, 2020.
https://doi.org/10.1007/978-3-030-65390-3_5

not know what it does not know" [2]. This lack of interpretation and over-confidence estimates would make classifiers unreliable on high risk tasks.

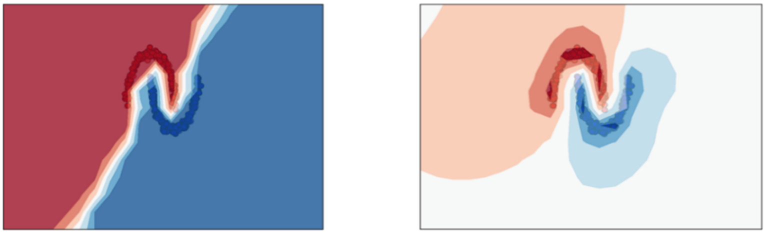


Fig. 1. Illustration of over-confidence in Moon Toy dataset. The left plot represents the KNN's over-confident prediction on out-of-distribution area. The right plot represents our approach of confidence estimation for avoiding over-confidence on out-of-distribution samples.

In this paper, we propose a new approach to deal with the over-confidence problem. For practical applications, when a test sample drops in a confusing area or the area far away from the familiar data, its predicting confidence score would be lower to indicate that the prediction results have a lower degree of reliability, see Fig. 1. Our intuition of confidence estimation is that if a test sample is outside the distribution of the collected data, it would be far from the target cluster formed by the train samples, thus its local density would be larger than the cluster's. Our confidence estimation algorithm also considers a cluster density factor to reflect the coherence of a cluster.

One of the challenges in classification is small sample size (SSS) problem. For many real-world tasks, data is distributed in a high-dimensional and sparse space, where distance analysis would be suffered [3]. As the data dimension increase, the distance between the closest points increase and the farthest points decrease. In other words, the increase in dimensionality reduces the difference between far and near points, gradually approaching to zero, which leads to the difficulty of distance computation.

Dimensionality reduction methods are usually used in high-dimensional data sets, which include feature transformation techniques, such as principal component analysis and singular value decomposition. These transformations usually preserve the original relative distance between samples. In order to obtain category-differentiated projections, we use linear discriminant analysis to find a subspace in which similar objects are grouped together and dissimilar objects are separated. Experiments show that our discriminative subspace classification framework can provide well-calibrated confidence for avoiding over-confidence errors.

In this paper, our contributions are as follows:

- we propose a method to achieve an efficient discriminant subspace. It can exploit the advantages of nearest-neighbor analysis without suffering from the sparsity of high-dimensional data.
- We propose an approach to estimation of prediction confidence. Our method is designed to assign high consistency scores with familiar samples and low consistency score with unfamiliar or confusing samples.

2 Related Work

Bayesian Neural Network. Bayesian approach provides an approximation on the posterior over the parameters of network and uncertainty estimation over the prediction [4]. However, it doesn't solve the over-confidence estimation for the unfamiliar samples, and the computational cost is high.

Confidence Calibration. Some works transform model's output into prediction probabilities, so are capable of identifying misclassifications. Hein *et al.* in [5] show that neural networks are unreliable when tested on semantically unrelated or out-of-domain samples. Guo *et al.* in [6] obtain confidence estimates by using ensembles of networks. There are works on confidence estimation based on low-dimensional feature space without much computational costs [7]. Jiang *et al.* in [8] proposed trust score to better identifies correctly-classified points than the model itself. Mandelbaum and Weinshall in [9] showed distance-based score achieves better results and note that adding a scaling factor to the distance in calculation would be beneficial to confidence estimation. Li and Hoiem in [2] used several confidence evaluation metrics contain Brier score, negative log likelihood of predictions, and expected calibration error (ECE). And they designed a new metric as the measurement of over-confidence faults.

Outlier Detection. In real world applications, data are often collected by sensors or other types of measuring instruments. When there are instrument errors in the equipment, measurement errors caused by human operations, or when there are differences in equipment parameters between different batches of measuring instruments, abnormal data may result. Anomaly detection approaches are distance-based or density-based, such as Local Outlier Factor (LOF). Some approaches use the local distance to the k-nearest neighbors (KNN) to label observations as outliers or non-outliers. Combining the prior work of KNN based outlier detection, our confidence estimation mainly considers using the nearest distance to estimate the class assignment.

Subspace Methods. Subspace methods are a category of classification methods widely applied to classify high-dimensional data [10]. To overcome the difficulty with sparsity in high dimensional feature space, especially lack of sufficient samples, researchers have investigated several dimension reducing algorithms [11]. Common Principle Component Analysis (PCA), Linear Discriminant Analysis (LDA), Local Linear Embedding (LLE), *etc.* are all subspace learning methods based on spectral methods.

3 Methodology

The framework of our method is depicted in Fig. 2. The predicting pipeline is executed in two stages:

Firstly, to deal with within-class scatter matrix singular and computational difficulty of LDA, we achieve embedding by a pre-trained neural network, and project it onto a visualized discriminant subspace.

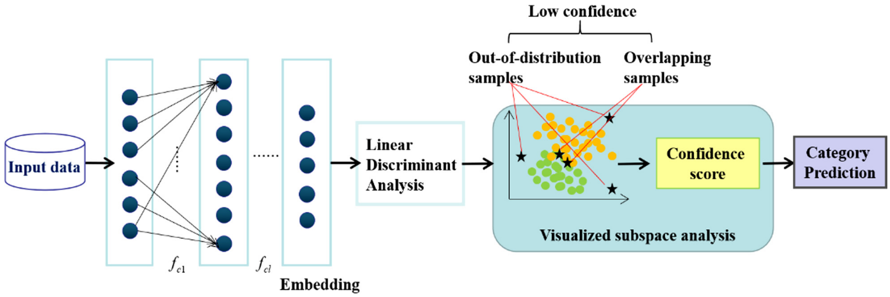


Fig. 2. Pipeline of our proposed classification framework

Secondly, to avoid over-confidence errors, we estimate the test sample's distribution consistency to different clusters on the proposed subspace. We design a new confidence score, for which out-of-distribution and overlapping samples will get lower confidence score.

3.1 Discriminative Visualized Subspace

Embedding from High Dimensional Data. For a multiclass classification problem, we first train a neural network by arranging the labeled-data in the form of matrix and feeding it to the network. We obtain feature vectors through multilayer perception of the network, which would extract discriminatory information from its original space. We assume that L is the number of layers of the pretrained neural network. The embedding can be achieved from a dense layer, which often be the last hidden layer:

$$F(X) = \psi_{L-1}(X) \tag{1}$$

Where $\psi_{L-1}(.)$ denotes the mapping function from input to the last hidden layer.

LDA Projection. Linear Discriminant Analysis (LDA) is often used as a dimension reduction technique in the pre-processing step of a machine learning modeling pipeline. The goal of LDA algorithm is to project the original space onto a lower-dimensional space (where the target dimension $\leq \mathrm{C} - 1$), while maintaining the class-discriminatory information [3].

Suppose in a 3-class classification problem, the first step in LDA is finding two scatter matrices referred to as the "between class" and "within class" scatter matrices. The definitions are as follows:

Variable $S_{W\mathrm{i}}$ is the within-class scatter matrix of the samples from category i relative to the center point from the same category.

$$S_W = S_{W1} + S_{W2} + S_{W3} \tag{2}$$

$$S_{Wi} = \sum_{f \in ci} (f - \mu_i)(f - \mu_i)^T \tag{3}$$

Variable $S_{B\mathrm{i}}$ is the between-class scatter matrix of the center of category i relative to the overall mean point.

$$S_B = S_{B1} + S_{B2} + S_{B3} \tag{4}$$

$$S_B = (\mu_1 - \mu)(\mu_1 - \mu)^T + (\mu_2 - \mu)(\mu_2 - \mu)^T + (\mu_3 - \mu)(\mu_3 - \mu)^T \tag{5}$$

In the formula,$\mu = \frac{1}{N}\sum_{\forall f} f$, $\mathrm{\mu}_i = \frac{1}{n_i}\sum_{f \in ci} f$, $i = 1, 2, 3$. n_i denotes the number of samples of the *i-th* category, *ci* represents the *i-th* category.f denotes the feature vector.

To optimize the transformation space basis vector, the goal of LDA algorithm is to make the distance between samples of different classes after projection and the distance between samples within class closer. The optimization objective function to solve the projection matrix W is as follows:

$$\mathrm{J(W)} = \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|} \tag{6}$$

According to $S_W^{-1} S_B w_i = \lambda w_i$, we can get the eigen vectors by solving the singular value decomposition.

In order to visualize the clustering points, we project the embedding space into a two-dimensional space and coordinates of the corresponding space are denoted as (y_1, y_2). Then we use the space basis vector w_i, obtained in the previous step to perform the projection:

$$y_1, y_2) = \left(w_1^T f, w_2^T f\right) \tag{7}$$

Coordinates Normalizing and Subspace Visualization. Perform minmax-scaling to the coordinates:

$$\mathrm{y}' = \frac{y - y_{min}}{y_{max} - y_{min}} \tag{8}$$

In Eq. (8), y' denotes the coordinate after normalization, y denotes the coordinate before normalization, y_{min} denotes the minimum coordinate, and y_{max} denotes the maximum coordinate. Since LDA projection can be viewed as a projection onto a set of bases with coordinates, we can draw a scatter plot to visualize our samples.

3.2 Confidence Score

In this section, we perform cluster analysis based on LDA projection space and design a method of confidence score estimation. Our classification framework focuses on reducing the confidence of the samples outside the known distribution and the samples from overlapping region, to escape from over-confidence classification errors.

KNN as a Baseline. In Euclidean subspace, KNN algorithm provides a posterior probability estimation, for which a point of x belongs to class C is proportional to its nearest neighbors belonging to class C. Specifically, let $S(x)$ denotes the projection point, k_m denotes the number of samples belong to class C among its k nearest neighbors. Then the probability of assigning x to class C is defined as follows:

$$P_{knn}(C|S(x)) = \frac{k_m}{k} \tag{9}$$

Overlapping Samples Filtering. To filter out samples that are prone to be misclassified, we first sort the samples from training set according to its posterior probability P_{knn}. Then we select a percentage threshold to perform filtering. The top percentage of the samples are retained and the rest are filtered out. Thus, most of the overlapping samples in subspace can be filtered out.

Definition 1. Confidence Score

Our proposed confidence score is to estimate the reliability of an unknown sample to be assigned to the class predicted. Our approach is based on Euclidean local distance between the test point and its K nearest neighbors from the training data set, see Fig. 3.

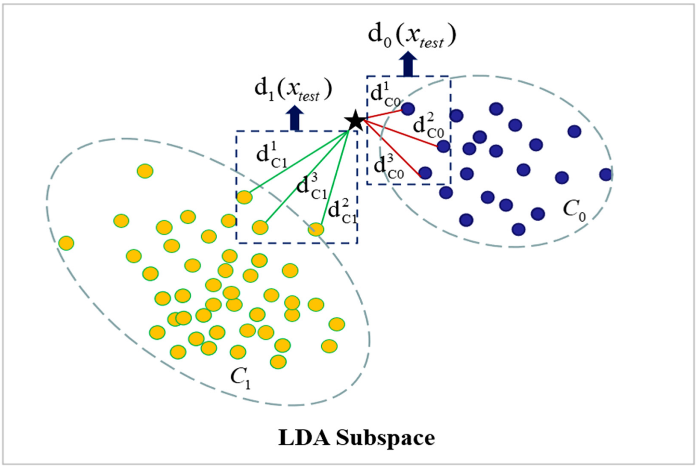


Fig. 3. Diagram of confidence score calculation

For a particular test sample x_{test}, we first compute the local distance of the test sample x_{test} to its nearest samples of category c, which is denoted as $d_c(x_{test})$.

$$d_c(x_{test}) = \frac{1}{k} \sum\nolimits_{i=1,y=c}^{k} \langle dx_{test}, x_i \rangle \tag{10}$$

Definition 2. Density Adaptive Factor

As data in real world is often class imbalanced and distribution density of projecting points from different class is often different, distance-based confidence estimation would need a density adaptive factor.

In this article, we propose a density adaptive factor which is calculated based on average distance between projection points in training data set. Intuitively when the training samples in the space are sparse, the distance between the test sample and the predicted category samples could also be sparse. We define $\overline{\boldsymbol{d_c}}(\boldsymbol{x_{train}})$ as the density adaptive factor, estimating the average distance between points in category C.

$$\overline{\boldsymbol{d_c}}(\boldsymbol{x_{train}}) = \frac{1}{n_c}\sum\nolimits_{i=1,y=c}^{n_c}\left(\sum\nolimits_{j=1,y=c}^{k} d\langle x_i, x_j\rangle\right) \tag{11}$$

Where $\langle dx_i, x_j\rangle$ denotes the Euclidean distance between x_i and x_j, used as a measure of similarity between two samples. The last part of the formula calculates the average distance of the *i-th* training sample x_i of category C from its nearest K neighbors within the same class.

Definition 3. Factorized Confidence Score

We modulate the confidence score $d_c(x_{test})$ with the category density adaptive factor $\overline{\boldsymbol{d_c}}(\boldsymbol{x_{train}})$. Then the factorized confidence score becomes:

$$\boldsymbol{t_c}(\boldsymbol{x_{test}}) = \frac{\overline{\boldsymbol{d_c}}(\boldsymbol{x_{train}})}{\boldsymbol{d_c}(\boldsymbol{x_{test}})} \tag{12}$$

As the average distance factor $\overline{\boldsymbol{d_c}}(\boldsymbol{x_{train}})$ is fixed, the factorized confidence score is inversely proportional to the test samples' local distance. When x_{test} becomes closer to the cluster of category C, the denominator turns to be smaller, and thus the confidence score becomes larger.

We add a fractional denominator for probability interval scaling:

$$\boldsymbol{P}_{\text{cs}}(\boldsymbol{C}|\boldsymbol{x_{test}}) = \frac{\boldsymbol{t_c}(\boldsymbol{x_{test}})}{\boldsymbol{t_0}(\boldsymbol{x_{test}}) + \boldsymbol{t_1}(\boldsymbol{x_{test}}) + \boldsymbol{t_2}(\boldsymbol{x_{test}})} \tag{13}$$

Definition 4. Predicted probability

In a classification framework, the prediction probability is used to classify a test data, denoted as $\boldsymbol{P}(\boldsymbol{c}|\boldsymbol{x})$. We calculate it by two parts, one is the factorized confidence score $\boldsymbol{P_{cs}}$ and the other is the KNN prediction probability $\boldsymbol{P_{knn}}$.

$$\boldsymbol{P}(\boldsymbol{c}|\boldsymbol{x}) = \boldsymbol{\alpha} \times \boldsymbol{P_{cs}}(\boldsymbol{C}|\boldsymbol{x_{test}}) + (1 - \boldsymbol{\alpha}) \times \boldsymbol{P_{knn}}(\boldsymbol{C}|\boldsymbol{x_{test}}) \tag{14}$$

In the formula, α is used as a weight parameter for the proportion of confidence score in computing the final predicted probability.

Algorithm: our proposed classification framework: SS-AOE

```
Input: training samples $x_{train}$, training labels $y_{train}$, test samples $x_{test}$, pre-trained NN model.
Parameters: Number of nearest neighbor $K$, filtering percentage $m$, confidence weight $\boldsymbol{\alpha}$.
1: Achieve discriminative projection coordinates: S ($x_{train}$), S ($x_{test}$).
2: Filtering overlapping train points. filtering out the $m$%train sample with lowest $p_{knn}(y_{train}|S(x_{train}))$ .
3: Derive the density adaptive factor $\overline{d_c}(x_{train})$ for each category. Compute the average $K$-nearest-neighbor distances of each training points cluster.
4: Get the confidence score of $x_{test}$ belonging to each category. Query the $K$ closest training points in each category of $x_{test}$, and calculate their average distance.
5: Factoring the confidence probability $d_c(x_{test})$ and derive the final dence $P_{cali}(c|x)$.
6: Output category prediction and the probability of test sample assigned to that class.
```

4 Experimental Evaluation

4.1 Dataset and Preprocessing

Stroke Diagnosis Dataset. The effectiveness of our proposed method is tested on a real-world application. It contains 2,340 patients' Electrical Health Records (EHRs) collected from hospitals for stroke disease predicting. The EHRs are used to test classification for stoke types. Specifically, it is a 3-class classification problem to detect Ischemic stroke, Hemorrhagic stroke, or Normal situation (Table 1 and 2).

Table 1. Data description

Instance number	Dimension	Datatype
2,340	1,776	Numerical

Table 2. Category distribution

Category	Sample numbers
Ischemic stroke	762
Hemorrhagic stroke	1,313
Normal	265

Data Preprocessing. Perform z-score standardization to the original data. Split the dataset into training (80%) and testing (20%) set.
Parameter Setting. Through the experiments, the parameters which achieve the best result are as follows: Nearest neighbor number K is set to 5, m is set to 20%, α is set to 0.5.

4.2 Confidence Calibration Evaluation

Calibration Metrics. We select 5 metrics to evaluate the effectiveness of the confidence estimation. In the following metrics, the smaller means better. Denote $P_m(y_i|x_i)$ as the assigned confidence in the correct label for the *i-th* sample of totle N samples.

Brier: The root mean squared difference between one and the confidence in the correct label:

$$\frac{1}{N}\sum_i (1 - P_m(y_i|x_i))^2)^{\frac{1}{2}} \tag{15}$$

when the correct label is predicted with high confidence, the Brier score is small.

ECE: Expected calibration error is used to measure the error between accuracy and confidence estimation.

$$\sum_{j=1}^{J} \frac{|B_j|}{N} \left| acc\left(B_j\right) - conf\left(B_j\right)\right| \tag{16}$$

In the formula, B_j is a set of predictions binned by confidence quantile, $acc\left(B_j\right)$ is the average accuracy of the B_j, and $conf\left(B_j\right)$ is the average confidence in the most likely category.

E99: Is the error rate among the subset of samples that have at least 99% confidence in any label. If the classifier is well-calibrated, E99 should be less than 1% [2].

Label Error: Is measured as the percent of incorrect most likely labels, or 1 minus average precision.

Table 3. Classification evaluation

Model	Ischemic stroke		Hemorrhagic stroke		Normal	
	Precision	Recall	Precision	Recall	Precision	Recall
LR	0.63	**0.93**	**0.97**	0.78	0.87	0.85
SVM	0.87	0.54	0.82	0.96	0.89	0.76
RF	0.69	0.82	0.94	0.88	0.87	0.85
SS-AOE	**0.91**	0.85	0.92	**0.97**	**1.00**	**1.00**

Table 4. Confidence evaluation

	Brier	ECE	E99	Label error
KNN predict probability	0.216	0.045	0.018	0.083
Our proposed confidence	**0.213**	**0.030**	**0.000**	0.083

Table 3 shows the classification result of the stroke diagnosis dataset. We compare with other machine learning algorithms: Logits Regression (LR), Support Vector Machine (SVM), Random Forest (RF). Evaluation metrics are the commonly used measurement in medical diagnosis: Precision and Recall.

Table 4 shows the evaluation results for KNN prediction probability (Baseline) and our proposed calibrated confidence. We report 4 metrics including the Brier score, the Expected Calibration Error (ECE) for confidence effectiveness, the Label Error for the discrimination based on confidence value, and the 99% error rate (E99) for over-confidence error measure. All the metric is the lower, the better.

Calibration Curve Plotting. Calibration curves, also called reliability curves, are used for illustrating the properties of probabilistic forecast system [12]. The reliability graph is a line graph of the observed relative frequency (y-axis) versus the predicted probability (x-axis). Specifically, the predicted probability is divided into a fixed number of buckets along the X axis. Then we count the number of events for each bin (category $= 1$). Finally, the counts are standardized. Then we plot the result as a line graph. The better or more reliable the predictions are calibrated, the closer these points will appear along the main diagonal from the lower left corner to the upper right corner of the graph. Below, the diagonal line indicates that the model has been over-confidence; above the diagonal line indicates that the model is under-confidence.

Through the calibration curves in Fig. 4, it's clearly seen that the curve marked by our proposed confidence score is closest to the diagonal, which is proved to be more reliable than other methods. Experiments show that our proposed category density factor improves the reliability of confidence estimation.

4.3 Out-of-Distribution Confidence Evaluation

Subspace Visualization. Figure 5 shows a visualization plot of the subspace we proposed. The points in the picture denote instances of EHRs from a real-world (our proprietary) stroke diagnosis dataset. The different colors are used to denote different class labels. The points in light colors are plotted based on training dataset and the points in black color are plotted based on test dataset.

Through this visualization, we can see that the same category samples are grouped together in the projected subspace. But there exist overlapping samples between orange and blue clusters. Therefore, it is necessary to filter out these confusing training samples

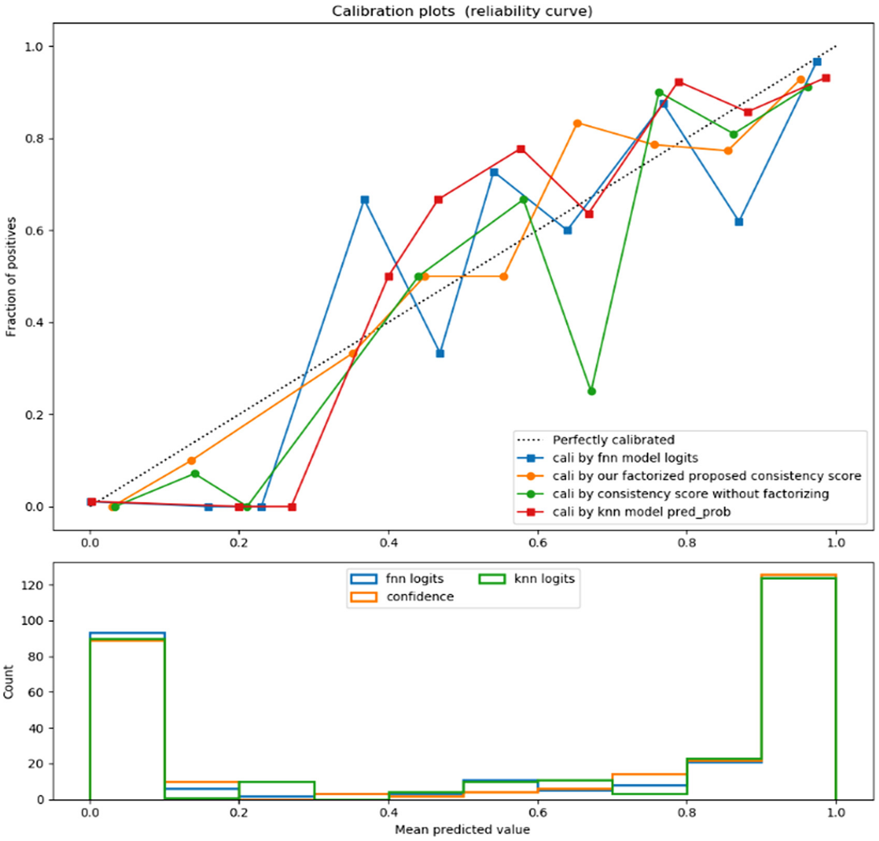


Fig. 4. Calibrated and Uncalibrated confidence Reliability Diagram. Evaluation result of Fully-connected Neural Network (FNN) output Logits, KNN predicting probability, our proposed confidence score, and the confidence score without density factorizing.

which are prone to be misclassified. Figure 5 shows that our filtering step effectively discarded those uncertain training samples, and our subspace is visually well discriminating and generalized.

Heatmap Evaluation. Figure 6 visualizes the confidence score distribution of all points in subspace with two purposes:

- Evaluation on the confidence of familiar samples and out-of-distribution samples.
- Evaluation on the effectiveness of category discrimination.

The predicting confidence estimated by KNN is close to 100% for the points of the true category. But the points far away from the known distribution are also given such high confidence, which are likely to be outliers or out-of-domain data. We can see that our proposed method achieves the highest confidence scores for the points near the familiar points from the same category. As the sample points fall into areas far from

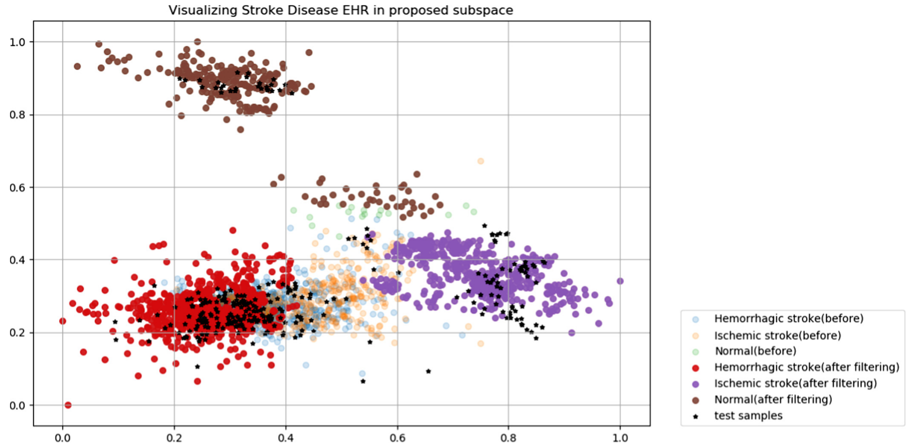
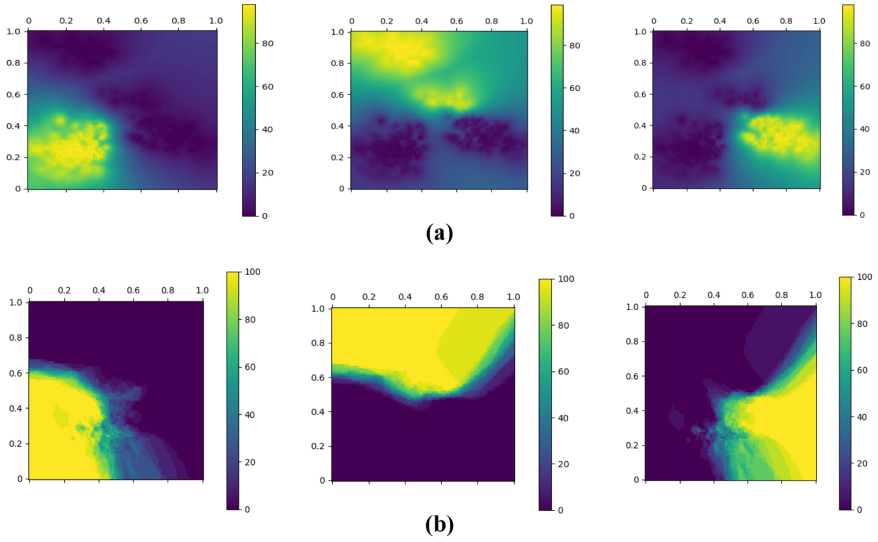


Fig. 5. Discriminative subspace visualization of before & after filtering. Scatter diagram shows the points with two-dimensional coordinates achieved by our proposed method.



(a)

(b)

Fig. 6. Confidence Heatmap plots for our stroke diagnosis dataset. (a) our proposed method predict probability. (b) KNN prediction probability. Left Row: Confidence scores of points assigned to Hemorrhagic stroke disease. Middle Row: Confidence scores of points assigned to normal persons. Right row: Confidence scores of points assigned to Ischemic stroke.

the training data points, the confidence scores gradually decrease. When the points fall in wrong category projection area, they would get the lowest confidence scores. Points in the overlapping region would also get low confidence scores. So, our confidence estimation can reflect the reliability of the prediction and avoid high confidence scores on the unknown data.

5 Conclusion

This paper proposed a classification framework with predictive confidence estimation to avoid over-confidence errors. First, in order to solve the SSS problem, we achieved the data embedding and then projected the features to a visualized discriminative subspace. Second, we designed a confidence score to describe the consistency of the test sample with the known samples and improved it by modifying the density-balancing factor we have designed. Experiments on a real-world brain stroke diagnosis dataset show that our confidence estimation approach is more reliable than the baseline method in predictive probability calibration. Confidence heatmaps show that our method achieved the goal of assigning lower confidence to unfamiliar samples and to those samples prone to be misclassified.

References

1. Liang, S., Li, Y., Srikant, R.: Enhancing the reliability of out-of-distribution image detection in neural networks. arXiv preprint arXiv:1706.02690 (2017)
2. Li, Z., Hoiem, D.: Improving confidence estimates for unfamiliar examples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2686–2695. IEEE, San Francisco (2020)
3. Wang, S., Nie, F., Chang, X., Li, X., Sheng, Quan Z., Yao, L.: Uncovering locally discriminative structure for feature analysis. In: Frasconi, P., Landwehr, N., Manco, G., Vreeken, J. (eds.) ECML PKDD 2016. LNCS (LNAI), vol. 9851, pp. 281–295. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46128-1_18
4. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in Neural Information Processing Systems, pp. 5574–5584, Long Beach (2017)
5. Hein, M., Andriushchenko, M., Bitterwolf, J.: Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 41–50. IEEE, San Francisco (2019)
6. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. arXiv preprint arXiv:1706.04599 (2017)
7. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv preprint arXiv:1610.02136 (2016)
8. Jiang, H., Kim, B., Guan, M., Gupta, M.: To trust or not to trust a classifier. In: Advances in Neural Information Processing Systems, pp. 5541–5552, Montreal, Canada (2018)
9. Mandelbaum, A., Weinshall, D.: Distance-based confidence score for neural network classifiers. arXiv preprint arXiv:1709.09844 (2017)
10. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2790–2797. IEEE (2009)
11. Vaswani, N., Bouwmans, T., Javed, S., Narayanamurthy, P.: Robust subspace learning: robust PCA, robust subspace tracking, and robust subspace recovery. IEEE Signal Process. Mag. **35**(4), 32–55 (2018)
12. Bröcker, J., Smith, L.A.: Increasing the reliability of reliability diagrams. Weather Forecast. **22**(3), 651–661 (2007)

Influence Maximization Based Active Learning in Noisy Setting

Yeliang Xiu[1](✉) and Hui Ma[2]

[1] Renmin University of China, Beijing, China
dadaqingjian@ruc.edu.cn
[2] Shenyang University of Chemical Technology, Shenyang, China

Abstract. In recent years, active learning has attracted great attention. However, there are still many challenges. On the one hand, the features of instances are usually affected by noise in many real-world applications. It is difficult to select the real important instances by sampling strategy purely based on the original features of instances. On the other hand, most of the existing sampling strategies in active learning are based on either informativeness sampling or representativeness sampling through the original features of samples. It is hard to choose the most important instance based on a certain sampling criterion or a rough combination of multiple sampling criteria. To solve these issues, a novel active learning algorithm with a low-rank representation-based influence maximization sampling strategy (**IMLRR**) was proposed in this paper. First, low-rank representation is adopted to construct an true affinity matrix for noisy instances. Then, in such a network, the sampling strategy based on influence maximization is used to select the most informative and representative instances at the same time from unlabeled data set. Finally, our empirical results demonstrate the effectiveness of our approach on different degree of noisy instances.

Keywords: Active learning · Low-rank representation · Influence maximization · Noisy sample

1 Introduction

In many real-world applications, only a small number of instances are labeled, most of them are unlabeled. However, it is really time consuming and expensive to mark all instances. Active learning [24,25,27] was then proposed to address this issue by selecting a subset of most significant instances from the unlabeled data set according to a certain sampling strategy and send them to the expert to mark. Active learning aims to achieve a high classification accuracy using as few labeled instances as possible, thereby minimize the labeling budget by prioritizing the selection of significant value data that can best improve model performance.

Unfortunately, the feature of instances usually contains some noise, which makes it difficult to select the most important sample purely based on the original

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 67–80, 2020.
https://doi.org/10.1007/978-3-030-65390-3_6

features of instances. As shown in the Fig. 1, the similarity of the two clean pictures on the left is 0.89. However, the similarity reduced to 0.69 after adding noise, which shows on the right side. It can be said that the noise in feature space can greatly affect the sampling strategy result of active learning.

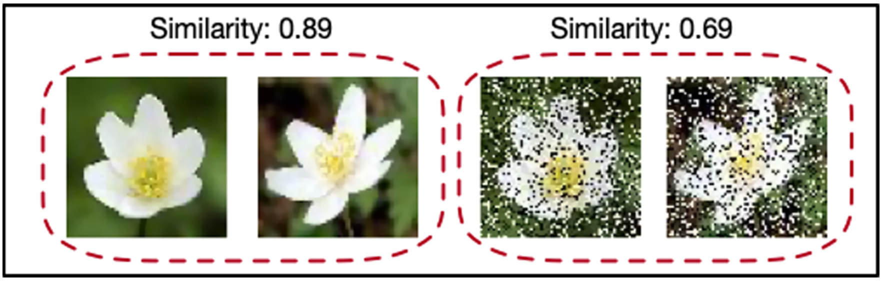


Fig. 1. Example of similarity between samples. The similarity of the two clean images show on the left side of the picture, and the similarity after adding the salt and pepper noise show on the right side.

Traditional active learning algorithm studies focused on to designing a sampling strategy based on a certain criterion to select the samples that have the greatest improvement in model performance, such as active learning algorithms based on representativeness [31], diversity [2], and uncertainty [1]. However, It is difficult to select the best sample for model based on a traditional sampling strategy or a simple combination of multiple sampling strategy from unlabeled data set.

To overcome these challenge, in this paper, we propose a novel active learning algorithm based on influence maximization in noisy setting. The proposed method provides a new sampling strategy for active learning. Since the features of instances are susceptible to noise in the real application, the low-rank representation model is constructed to calculate the true affinity matrix for noisy instances including labeled and unlabeled data set. And then influence maximization algorithm in the social network is introduced as the sampling strategy to find the most informative and representative instances. The main contributions of this paper are summarized as follows:

- In order to deal with the case of noise included in the features of instances, low-rank representation is utilized in our method, which could to calculate the true affinity matrix for noisy instances.
- The proposed algorithm selects both informative and representative instances simultaneously from unlabeled data set by using influence maximization algorithm.
- In the experimental part, a large number of empirical results reveal that the proposed algorithm has higher classification accuracy than the baseline approaches under the data with different degrees of noise.

The rest of this paper is organized as follows: We review related work in Sect. 2 and the low-rank representation and the influence maximization algorithm are described in Sect. 3. Our proposed active learning algorithm (**IMLRR**) is then introduced in Sect. 4. In Sect. 5, some numerical experiments are presented. Finally, we draw a conclusion of this paper and point out the future work in Sect. 6.

2 Related Work

The key task in active learning is to design a sampling strategy in the unlabeled samples pool to select the most valuable instances to update the learning model, which has got the attention of many researchers.

Now there are two known sampling strategies for active learning: informativeness sampling [1,16,29] and representativeness sampling [31]. The strategy of informativeness sampling is to select the most inaccurate instances predicted by the current learning model. This kind of sampling strategy has been widely applied to logistic regression [17], support vector machines [23,27], etc. Whereas those outliers tend to be easily selected in the informativeness sampling strategy which will definitely and significantly affect the performance of the new learning model. Different from informativeness sampling, the representativeness sampling strategy selects the instances which best represents all the unlabeled data set [15]. However, the representativeness-based active learning algorithms are sensitive to the results of clustering, which is one obvious limitation of representative sampling strategy. As a result, some active learning algorithms that combine informativeness sampling strategy and representativeness sampling strategy were proposed. In [12], the authors propose a new method to query the most informative and representative examples where the metrics measuring and combining are done using min-max approach.

Although there are a lot of research about active learning, including image classification [13,30] and natural language processing [21], there are still few active learning algorithms in sample noisy setting. To address these issues, Jian Wu et al. [28] has proposed an active learning approach for multi-label Image classification with sample noise, the author constructed a low-rank model to quantize noise level, and the example-label pairs that contain less noise were also emphasized when sampling. In addition, Donmez P et al. [6] and Sheng-Jun Huang et al. [11] have considered the case of noisy oracle, the proposed approaches may perform well on examples under very noisy oracle.

3 Low-Rank Representation and Influence Maximization

As mentioned above, in this research we will combine the approach low-rank representation (LRR) and influence maximization (IM) to select high value instance among the noisy data. In this section, we will briefly introduce the low-rank representation and influence maximization.

3.1 The Low-Rank Representation

Let $X = [x_1, x_2, ..., x_n]$ in R^{d*n} be a set of data with noise that consists of n instances, and d is the number of all features. The idea of LRR is to capture the representation of each instance by the linear combination of the basis in a dictionary $A = [a_1, a_2, ..., a_n]^T$ [18]:

$$X = AZ \tag{1}$$

where $Z = [z_1, z_2, ..., z_n]$, each z_i can be the coefficient representation of x_i. Then, we can obtain the representation Z^* by solving the problem:

$$\min_Z \ rank(Z), \ s.t., X = AZ \tag{2}$$

We refer to the optimal solution Z^* obtained in the above equation as the low-rank representation of X with respect to dictionary A. However, the optimal solution of the above formula is an NP-hard problem. Fortunately, by matrix completion methods [4], we can solve the following problem which is equivalent to problem(2) in practice:

$$\min_Z \ ||Z||_* \ s.t., X = AZ \tag{3}$$

where $|| \cdot ||_*$ denotes the nuclear norm [8] of a matrix, i.e., the sum of the singular values of the matrix.

In order to build a undirected graph based on affinity matrix to select the most informative and representative instances by influence maximization in unlabeled data set. So the data X itself as the dictionary, i.e., problem (3) can be rewritten as:

$$\min_Z \ ||Z||_* \ s.t., X = XZ \tag{4}$$

In real applications, there are a lot of noisy data. A reasonable strategy is to relax the equality constraint in (4), similar to [3]. Therefore, a more reasonable object function of low-rank representation might be:

$$\min_{Z,E} \ ||Z||_* + \lambda ||E||_{2,1} \ s.t., X = XZ + E \tag{5}$$

where $||E||_{2,1} = \sum_{j=1}^{n} \sqrt{\sum_{i=1}^{n} ([E]_{ij})^2}$ is called as the $\ell_{2,1}$-norm,and λ is a parameter ($\lambda > 0$). In addtition, Eq. 5 can solve by Inexact ALM [18].

3.2 Influence Maximization

The influence maximization problem refers to finding K seed nodes in a network with N nodes, assuming that all seed nodes are activated, and the influence propagation of these seed nodes causes as many nodes in the network to be activated. Dominngos and Richardon [5,22] first studied the impact maximization problem as an algorithm problem. Kempe et al. [14] officially defined the influence maximization as a discretization optimization problem and proved that it is a NP-hard problem in the independent cascade model and the linear threshold model.

Definition 1. *Given a network $G =< V, E >$, influence propagation model, propagation probability, and the number of seed nodes K ($K \ll N$, N is the number of all nodes in network), the problem of influence maximization is to find the node subset S^* under the following condition:*

$$S^* = \underset{S \subseteq V, |S|=K}{\arg\ min}\ \sigma(S)$$

$\sigma(S)$ is an objective function, which represents the expected value of the number of nodes activated when the seed node is S. The objective function $\sigma(S)$ has three important properties:

- Non-negative: For any given instances set S, there is $\sigma(S) \geq 0$
- Monotonic: For any given instances set S and any instance $v \in V/S$, there is

$$\sigma(S \cup v) \geq \sigma(S)$$

- Submodular: For instances set $T \subseteq S \subseteq V$ and any instance $v \in V/S$, it hods that

$$\sigma(T \cup \{v\}) - \sigma(T) \geq \sigma(S \cup \{v\}) - \sigma(S)$$

4 The Proposed Algorithm: IMLRR

In this section. First, the definition of problem will be given. Then, we will briefly describe the framework of proposed method and sampling strategy based on influence maximization.

4.1 Problem Definition

We denote by $X_\ell = \{(x_1, y_1), (x_2, y_2), ..., (x_{n_\ell}, y_{n_\ell})$ the labeled training data set that consists of n_ℓ labeled instances and $X_\mu = \{(x_1, y_1), (x_2, y_2), ..., (x_{n_\mu}, y_{n_\mu})$ be the data set that consists of n_μ unlabeled instances. Active learning aims to select K instances $(x_{s1}, x_{s2}, ..., x_{sk})$ from the pool of unlabeled data set to query its class label in every iteration. For convenience, we divide the all data set X into three parts: the labeled instances X_ℓ, the currently selected data set X_s, and the rest of unlabeled instances X_u.

In order to reduce the cost of sample marking, active learning method has attracted the attention of many researchers. However, previous method did not take into account that the features of samples were easily affected by noise. Removing the noise contained in features of instances can help to select the instance that truly improves the performance of model from the unlabeled data set. In our approach, low-rank representation and influence maximization algorithms were utilized to select the most informative and representative instances simultaneously in noise data.

4.2 The Framework of Proposed Algorithm

Figure 2 illustrates the overall architecture of our method, which consists of three key steps, namely, **Low-rank representation**, **Construct undirected graph** and **Select instances by CELF**. First, we start by using a small set of labeled instances to train a model. In addition, according to low-rank representation algorithm, a low-rank representation Z^* in Eq. 5 relative to the all original data set X can be obtained. And then we can calculate the true similarity (i.e., affinity matrix, denoted by Z^*) between noise instances. Next, based on constructed undirected graph by affinity matrix, the most informative and representative instances are selected by influence maximization algorithm in each iteration. And the selected instances will be given to the expert for marking, and then the model is retrained on all labeled instances. The process is iterated until the model converges.

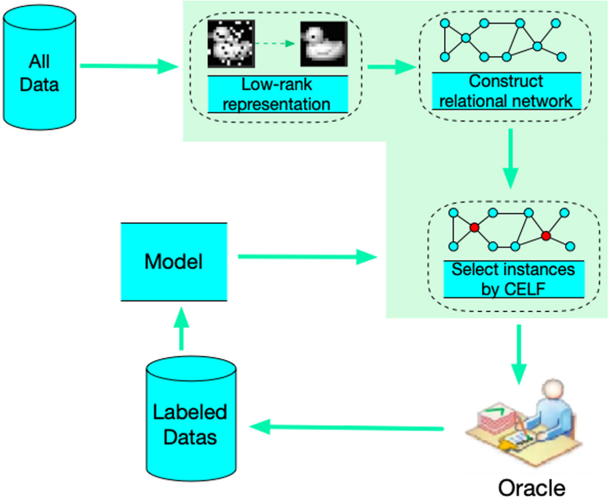


Fig. 2. The framework graph of the IMLRR algorithm

4.3 Construct Relational Network and Sampling Strategy

In this section, we will describe how to utilize the affinity matrix (denoted by Z^* in Eq. 5) to construct edge weights of an undirected graph between instances, and introduce the proposed sampling strategy.

As shown in Eq. 6, the instances vectors correspond to the vertices and the edge weight between x_i and x_j is defined The affinity $W_{i,j}$ between x_i and x_j is defined as $|[Z^*]_{ij} + [Z^*]_{ji}|$.

$$W_{ij} = \begin{cases} |[Z^*]_{ij} + [Z^*]_{ji}| & \mathrm{i} \neq \mathrm{j} \\ 0 & \mathrm{i} = \mathrm{j} \end{cases} \tag{6}$$

To enable affinity matrix W to represent the weight between instances in the influence maximization algorithm [19], it is necessary to make the sum of the weights of all adjacent nodes of $\sum_{j\in\Omega(i)} W_{ij} \leq 1$, $\Omega(i)$ represents all adjacent nodes of node i. So it is necessary to normalize W by the following formula.

$$W_{ij} = \begin{cases} \frac{W_{ij}}{\sum_{j=1} W_{ij}} & \text{i} \neq \text{j} \\ 0 & \text{i} = \text{j} \end{cases} \tag{7}$$

A undirected graph $G = <V, E>$ is constructed through affinity matrix W, where $V = X$ represents the set of all instances, E represents the set of all edges in the network, W_{ij} represents the influence (i.e., similarity) of ith instance to jth instance.

Next, the influence-maximal seed nodes are selected in G by using the influence maximization algorithm in social network. And the greedy method CELF [19] is adopted in this paper. The CELF approach can select the most informative and representative instances from both labeled and unlabeled data set.

A complete description of the proposed algorithm is given in Algorithm 1 in the following part.

Algorithm 1. Active learning based on influence maximization and Low-Rank Representation (IMLRR)

```
Input: data matrix $X$, the number of iterations $T$, the number of selected
       instances in one iteration $K$
Output: $X_\ell$
Initialization: labeled data set $X_\ell = \{x_{y1}, x_{y2}, ..., x_{y_m}\}$ , unlabeled data
set $X_u = X \backslash X_\ell$ , t =0, T =10.
Obtain the low-rank representation $Z^*$ on $X$ by solving problem Eq. 5.
Construct an network by using $W$ in Eq. 7.
while $t \leq T$ do
    Select $K$ instance $X_s$ by CELF algorithm.
    Query the labels of all instances in the $X_s$.
    $X_\ell = X_\ell \cup X_s$
    $X_u = X_u - X_s$
    t = t+1
end
Return $X_\ell$.
```

5 Numerical Experiments

In this section, we first introduce the experimental setting. After that, analysis on the effectiveness of the proposed algorithm and its comparison with other baselines in common active learning methods are presented, in this part we also evaluate the methods on eight data sets including three image data set. The proposed algorithm actually achieves higher classification accuracy on different degree of noisy instances according to experiments results.

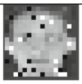
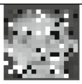
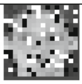
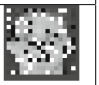
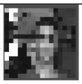
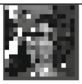
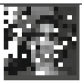
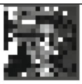
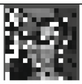
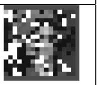
5.1 Experimental Setting

In the experiment part, eight datasets are used for evaluation, including three face image datasets ORL [26], YaleB [9], COIL [20] and five datasets from UCI [7]. The ORL dataset consists of 40 distinct subjects and there are 10 different face images for each subjects. In this paper, we only selected 100 instances from 10 distinct subjects. As to the YaleB dataset, there are 576 images of 10 individuals. The COIL dataset consists of 20 object images. For each object, 72 gray images are taken from different view directions. It is necessary to point out that in this paper these three face images datasets are resized to 16 * 16. The detailed description of all datasets are listed in Table 1 in the following.

Table 1. The description of the datasets

Dataset	#Instances	#Features	#Classes	#The number of labeled for first time	#The number of instances selected in each iteration	#The number of iteration
ORL	100	256	10	10	5	9
COIL	1460	256	20	200	20	20
YaleB	5760	256	10	100	20	20
Heart	174	13	5	25	5	10
Cancer	683	9	2	10	5	20
Tic	958	9	2	10	5	20
Segment	2310	19	7	70	10	20
Waveform	5000	21	3	30	20	10

Table 2. Some examples of original and corrupted images under different densities (Den.) of the salt and pepper noise from the ORL, COIL, YaleB.

Dataset	Clear data	Den.=0.1	Den.=0.15	Den.=0.2	Den.=0.25	Den.=0.3
ORL						
YaleB						
COIL						

To demonstrate the effectiveness of our algorithm IMLRR, the following baseline methods are used to compare with our approach in the experment part.

- Random Sampling (Random): Select query instances from the date set randomly.
- Entropy-based active learning (Entropy) [10]: Select the sample with the largest information entropy under the current classification.
- Support Vector Machine active learning (SVM) [27]: Selects the points closest to the current decision boundary of the SVM classifier as the most informative ones.
- Active learning by Query Informative and Representative instance (QUIRE) [12]: Query the most informative and representative examples where the metrics measuring and combining are done using min-max approach.

In order to make the data sets adaptive to the active learning scenario in our experiment, for each dataset, 70% of the instances are randomly selected as the training instances and regard the left 30% as the testing instances. In each training data set, some instances are treated as labeled ones, and the others are treated as unlabeled ones. The numbers of labeled instances are shown in Table 1. The number of iterations of every algorithm and the number of instances selected in each iteration are presented in Table 1.

After each iteration, the learning model is updated and estimated by a classifier. One-versus-all SVM classifier is chosen as the classifier in this paper.

To illustrate the effectiveness of our proposed algorithm on different degree of noisy instances, every image is perturbed with salt and pepper noise. Some of the sample images and noisy images of these three datasets are demonstrated in Table 2. Furthermore, Gauss noise was also added to those five datasets from UCI. The noise ratios added to all instances are 0, 0.1, 0.15, 0.2, 0.25, 0.3, respectively.

5.2 Comparison and Analysis

Figure 3 shows the trendlines of the final classification accuracy with the incremental noise ratios. The final classification accuracy means the classification accuracy of each algorithm after all iterations finished.

Several observations could be obtained from Fig. 3. First, the trendline of all methods decreases as the noise ratio increases, which shows that, the classification accuracy of every algorithm gradually decreases with the incremental noise. Second, the trendline of proposed algorithm (i.e., the red line) often higher than the others when the noise ratio is bigger than zero, which illustrate that the algorithm proposed in this paper is more effective compared with other algorithms when the sample noise is increased. Next, on Heart and ORL data set, our approach has higher classification accuracy than other algorithms on all different degree of noisy. In particular, on the ORL dataset, the proposed method has a classification accuracy of 6.6% higher than the comparison approaches when noise ratio is 0.3. This shows that, in most cases, the proposed active learning algorithm outperform the other algorithms on the noisy data. Therefore, our method is more effectiveness than baseline algorithms by selectet both informative and representative instances in noise setting.

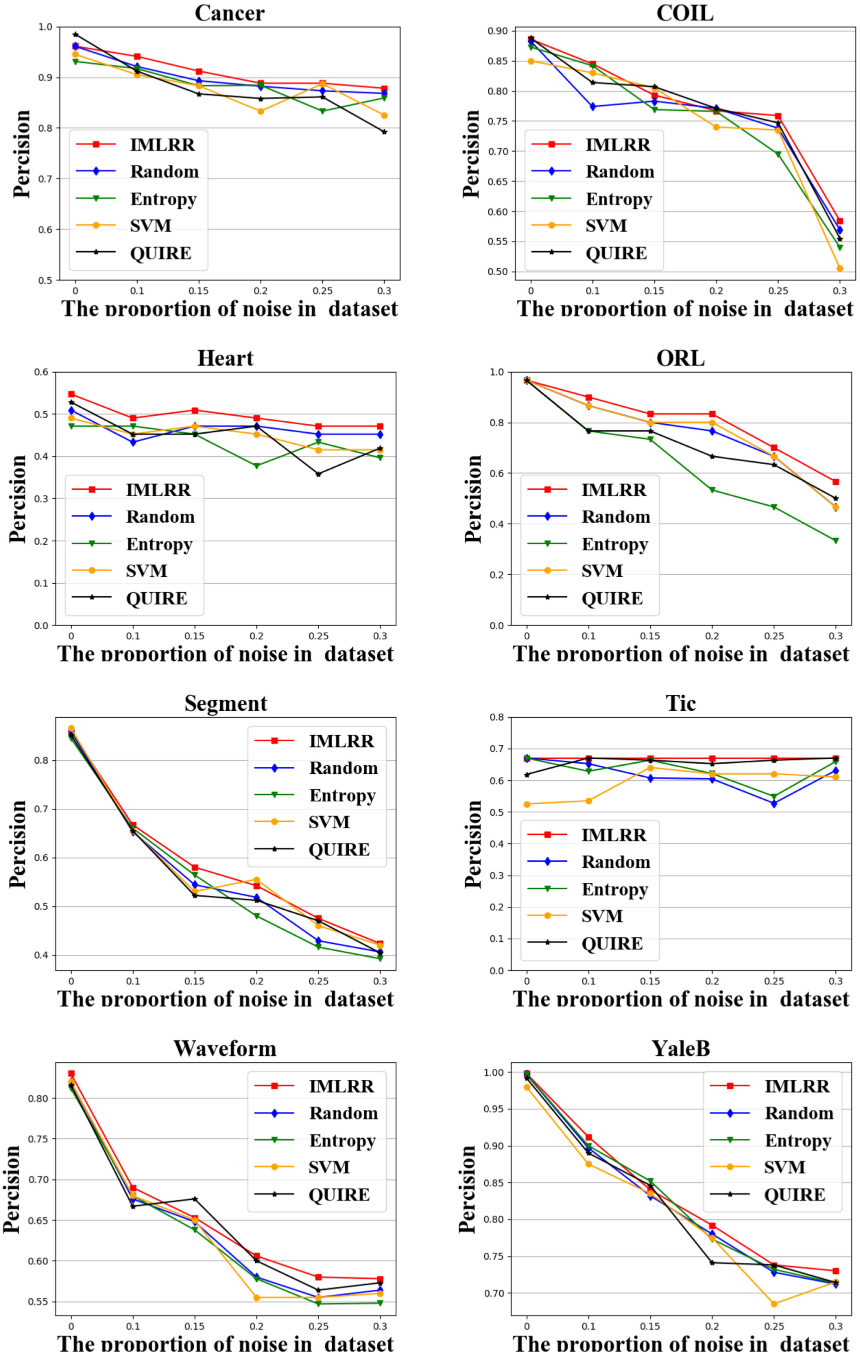


Fig. 3. The classification accuracy under different degrees of noise

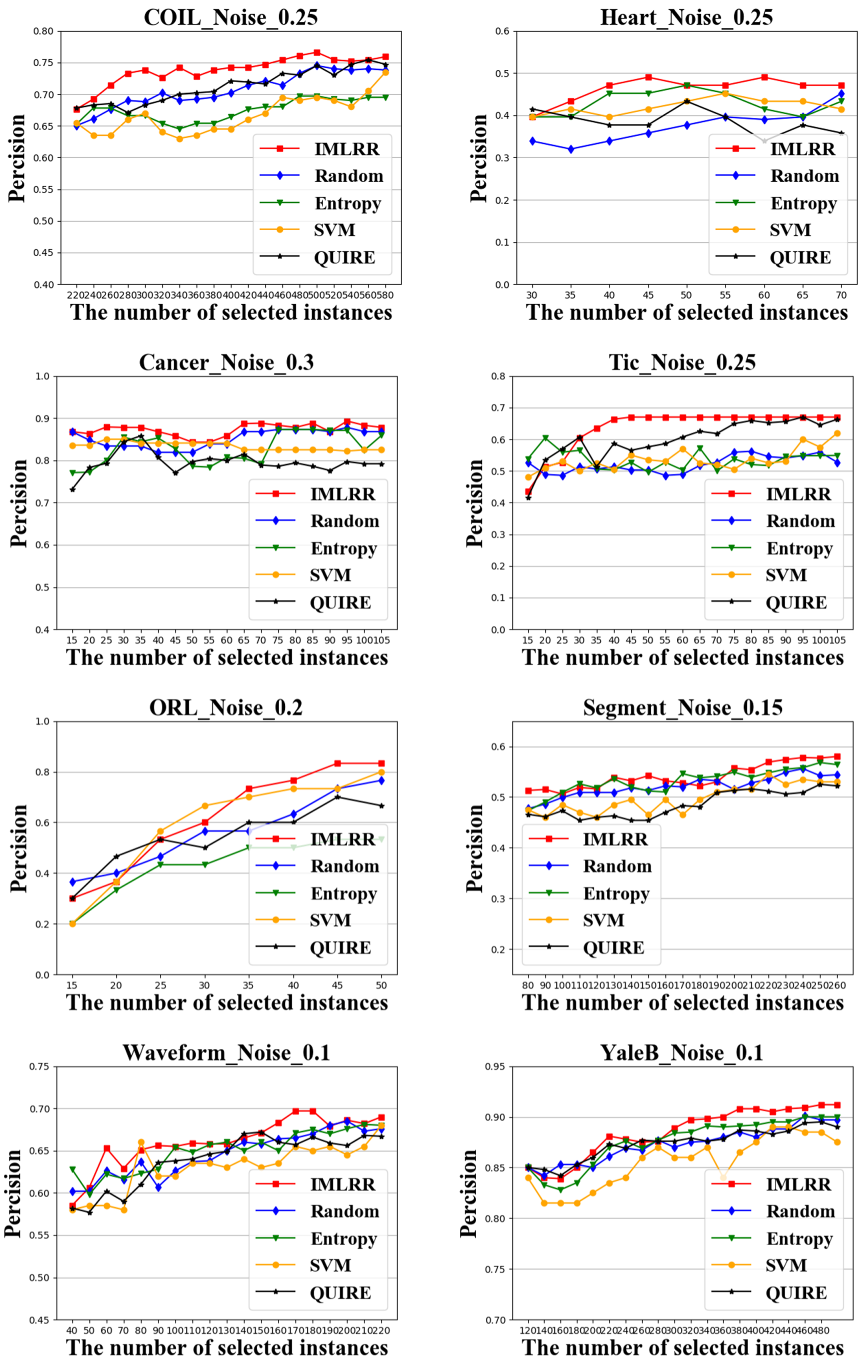


Fig. 4. The comparison of classification accuracy of each iteration under different noises

To further illustrate the effectiveness of proposed algorithm on the noisy data, as shown in Fig. 4, we show the trendlins of classification accuracy with the number of instances increasing at a fixed noise ratio.

Some facts can be observed from Fig. 4. On COIL, Tic, ORL and YaleB data sets, the trendlines increases with the adding of instances, and the classification accuracy of IMLRR is higher than others algorithms on most cases. Specially, the proposed algorithm achieves an accuracy of 0.67 using only 45 labeled instances on the Tic dataset, which is 11.5% higher than the best result of the comparison ones. Therefore, our method can reduce the number of labeled samples and enable the model to achieve higher performance. We attribute this success to the low-rank representation framework. Because the true affinity matrix between instances form noise data can be obtained by low-rank representation approach. In addition, the reason that the trendlines increase gently on Cancer dataset is that the labeled instance is given enough at the beginning. It is worth mentioning that, on Segment and ORL data sets, the classification accuracy of our method increased faster than others algorithms, although the classification accuracy of IMLRR in the previous iterations is slightly lower.

6 Conclusions and Future Work

A novel active learning algorithm based on low-rank representation and influence maximization (i.e., IMLRR) was proposed in this paper. The method can select the most significant instances in noise setting. The low-rank representation approach, which is less-sensitive to noise, is adopted to calculate the true affinity matrix between noisy instances. Next, an undirected graph based on affinity matrix was constructed, and the influence maximization algorithm (CELF) in social network was introduced in the undirected graph to select the most informative and representative instances from unlabeled data set. In addition, Our method can be used as an unsupervised active learning algorithm. Although the proposed sampling strategy in this research does not use the label information of the sample, we will consider adding the label information of the sample in the future, which is believed to further enhance the advantages and effectiveness of our approach.

References

1. Balcan, M.-F., Broder, A., Zhang, T.: Margin based active learning. In: Bshouty, N.H., Gentile, C. (eds.) COLT 2007. LNCS (LNAI), vol. 4539, pp. 35–50. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72927-3_5
2. Brinker, K.: Incorporating diversity in active learning with support vector machines. In: Proceedings of the 20th International Conference on Machine Learning (ICML 2003), pp. 59–66 (2003)
3. Candes, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? J. ACM **58**(3), 1–37 (2011)
4. Candes, E.J., Plan, Y.: Matrix completion with noise. Proc. IEEE **98**(6), 925–936 (2010)

5. Domingos, P.M., Richardson, M.: Mining the network value of customers. In: KDD (2001)
6. Donmez, P., Carbonell, J.G., Schneider, J.: Efficiently learning the accuracy of labeling sources for selective sampling. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 259–268. ACM (2009)
7. Dua, D., Graff, C.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml
8. Fazel, M.: Matrix rank minimization with applications/maryam fazel (2002)
9. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Anal. Mach. Intell. **23**(6), 643–660 (2001)
10. Holub, A., Perona, P., Burl, M.C.: Entropy-based active learning for object recognition. In: IEEE Computer Society Conference on Computer Vision & Pattern Recognition Workshops (2008)
11. Huang, S.J., Chen, J.L., Mu, X., Zhou, Z.H.: Cost-effective active learning from diverse labelers. In: IJCAI, pp. 1879–1885 (2017)
12. Huang, S.J., Rong, J., Zhou, Z.H., Huang, S.J., Rong, J., Zhou, Z.H.: Active learning by querying informative and representative examples. In: International Conference on Neural Information Processing Systems (2010)
13. Joshi, A.J., Porikli, F., Papanikolopoulos, N.: Multi-class active learning for image classification. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2372–2379. IEEE (2009)
14. Kempe, D.: Maximizing the spread of influence in a social network. In: ACM EC, vol. 4, pp. 137–146 (2000)
15. Krishnakumar, A.: Active learning literature survey (2007)
16. Lewis, D., Catlett, J., Cohen, W., Hirsh, H.: Heterogeneous uncertainty sampling for supervised learning (1996)
17. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. CoRR abs/cmp-lg/9407020 (1994). http://arxiv.org/abs/cmp-lg/9407020
18. Liu, G., Lin, Z., Yu, Y.: Robust subspace segmentation by low-rank representation. In: Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML 2010, pp. 663–670, Omnipress, USA (2010). http://dl.acm.org/citation.cfm?id=3104322.3104407
19. Lv, J., Guo, J., Zhen, Y., Wei, Z., Jocshi, A.: Improved algorithms of CELF and CELF++ for influence maximization. J. Eng. Sci. Technol. Rev. **7**(3), 32–38 (2014)
20. Nene, S.A., Nayar, S.K., Murase, H.: Columbia object image library (COIL-20. Technical report (1996)
21. Peshterliev, S., Kearney, J., Jagannatha, A., Kiss, I., Matsoukas, S.: Active learning for new domains in natural language understanding. arXiv preprint arXiv:1810.03450 (2018)
22. Richardson, M., Domingos, P.M.: Mining knowledge-sharing sites for viral marketing. In: KDD (2002)
23. Salcedo-Sanz, S., Rojo-Álvarez, J.L., MartíÂnez-Ramón, M., Camps-Valls, G.: Support vector machines in engineering: An overview. Wiley Interdisciplinary Reviews Data Mining & Knowledge Discovery 4(3), 234–267 (2014)
24. Settles, B.: Active learning literature survey. Computer Sciences Technical report 1648, University of Wisconsin-Madison (2009)
25. Settles, B.: Active learning. Synthesis Lect. Artif. Intell. Mach. Learn. **6**(1), 1–114 (2012)

26. Samaria, F., Harter, A.C.: Parameterisation of a stochastic model for human face identification. In: IEEE Workshop on Applications of Computer Vision (1994)
27. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. J. Mach. Learn. Res. **2**, 45–66 (2001)
28. Wu, J., Guo, A., Sheng, V.S., Zhao, P., Cui, Z.: An active learning approach for multi-label image classification with sample noise. Int. J. Pattern Recogn. Artif. Intell. **32**(03), 1850005 (2018)
29. Zhang, Z., Lai, Z., Xu, Y., Shao, L., Wu, J., Xie, G.S.: Discriminative elastic-net regularized linear regression. IEEE Trans. Image Process. **26**(3), 1466–1481 (2017)
30. Zhang, Z., Xu, Y., Shao, L., Yang, J.: Discriminative block-diagonal representation learning for image recognition. IEEE Trans. Neural Netw. Learn. Syst. **29**(7), 3111–3125 (2017)
31. Xu, Z., Xu, X., Yu, K., Tresp, V.: A hybrid relevance-feedback approach to text retrieval. In: Sebastiani, F. (ed.) ECIR 2003. LNCS, vol. 2633, pp. 281–293. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36618-0_20

Text Mining

DGRL: Text Classification with Deep Graph Residual Learning

Boyan Chen, Guangquan Lu(✉), Bo Peng, and Wenzhen Zhang

Guangxi Key Lab of Multi-Source Information Mining and Security,
Guangxi Normal University, Guilin, Guangxi, China
1119087734@qq.com, lugq@mailbox.gxnu.edu.cn, 450228460@qq.com,
jianjiu17@outlook.com

Abstract. Text classification is one of the most important problems in natural language processing. There are many useful features cannot be captured by traditional methods of text classification. Deep learning models have been proven that is able to extract features from data effectively. In this paper, we propose a deep graph convolutional network model that construct graph base on words and documents. We construct a new text graph based on the relevance of words and the relationship between words and documents in order to capture information from words and documents effectively. To obtain the sufficient representation information, we propose a deep graph residual learning (DGRL) method, which can slow down the risk of gradient disappearance. Experimental results demonstrate the effectiveness of the proposed model on various text datasets.

Keywords: Natural language processing · Text classification · Deep learning · Graph neural network.

1 Introduction

With the development of Internet, the emergence of new communication platforms such as E-mail and Twitter had made the data (eg. text, image) explosive growth. Therefore, data analysis is an important task for information classification, opinion mining, spam filtering. Usually, most data are recorded as text formal, and text classification task is an essential part of information processing technology and data mining analysis.

In recent years, many creative algorithms [15,31–33,36] have been proposed and applied to solve text classification problems, most algorithms generally improve the classification performance through two groups, one is designing the effective representation methods, such as the multi-view method [34] or constructing a suitable feature map [35,37,38], and the other is applying the neural network learning capability to obtain hidden feature representations. In order to extract text information effectively, people usually use some classic bag-of-words

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 83–97, 2020.
https://doi.org/10.1007/978-3-030-65390-3_7

model (BOW) to extract text representation, there are some traditional methods that learning text space representation, such as LDA model [1], N-Gram model. Ganguly et al. used word embedding to improve effectiveness of information retrieval [5]. Trstenjak et al. proposed K nearest neighbor (KNN) algorithm combine with term frequency-inverse document frequency(TF-IDF) method for text classification [25].

The development of pre-trained word embedding and neural networks has led to many classification methods. Dong et al. proposed a tree structured recursive neural network for sentiment classification and proved that recursive neural network capture the semantics of sentences effectively [4]. The common used recurrent neural network is BiLSTM [17]. CNN also obtains higher accuracy because it can capture the local correlation among features [9]. They capture textual features through the characteristics of the network structure effectively.

In addition to these classic neural network models, the emerging graph neural network has also received great attention in recent years. Kpif et al. proposed graph convolutional neural network (GCN) for classification [10]. Yao et al. proposed a graph constructed method based on GCN for text classification [29]. Linmei et al. proposed a heterogeneous information network (HIN) to enrich the relationship between texts and classify documents by GCN [16]. These algorithms demonstrate that they can capture more sufficient text features according to the richer relationship of text data is beneficial to classification performance of model.

Many algorithms obtain the sufficient representation information by increasing network parameters [6,22]. However, GCN generally does not design more than 3 layers because it will be easy to lead to gradient disappear. And the back-propagation of network is too smooth, the features will converge to the fixed value, so the obtained model is not suitable for target tasks [14].

In our work, we propose a novel method to construct graph and a new deep graph convolutional network for obtaining sufficient representation information, and do some experiments to analyze the effective on text classification tasks. We summarize our contributions as follows:

- A novel method has been proposed for constructing text relationship graphs, which consider the relevance of words and the importance of documents.
- We present a deep graph residual neural network for enhancing the ability network representation, and solve the problem that the graph network is easy to produce gradient disappearance.
- Experimental results demonstrate the effectiveness of the proposed model on various text datasets, and the proposed method is better than recent text classification algorithms.

2 Preliminaries

In this paper, there is a novel graph neural network for text classification, which takes the advantage of relation of data and allowing information propagation along the graph. We build the text relation by a new method named WWD (word-word-document) algorithm, which can incorporate the word information to enrich the relations features among the words and the document. Then we proposed a deep graph residual learning method (DGRL) to achieve text classification task.

2.1 Text Analysis

We present the WWD algorithm to build the relation of text data, which contains the information of document and word. Here, we construct the text information graph $G = (V, E)$. The set of nodes V contain the words, documents. The set of edges E contain the relations of the nodes.

First, we construct the correlation between words by pointwise mutual information (PMI) [2,29], which the PMI analyzes semantic correlation between words by calculating the co-occurrence of words. The higher the value of PMI, the greater the correlation between the words.

$$PMI(x,y) = \log \frac{p(x,y)}{p(x)p(y)}, \tag{1}$$

$$p(x,y) = \frac{E_w(x,y)}{S_w}, \tag{2}$$

$$p(x) = \frac{E_w(x)}{S_w}, \tag{3}$$

$E_w(x)$ is the number of sliding windows containing the word x on the corpus, $E_w(x,y)$ is the number of sliding windows containing the word x and the word y, and S_w is the number of the entire sliding window. Specifically, PMI is a calculation of the probability that the words x and y appear together divided by the probability that the words x and y appear separately.

TF-IWF algorithm [28] is a weighting technique for information retrieval and data mining, and it can capture the relationship between words and documents effectively. TF-IWF is the product of term frequency (TF) and inverse word frequency (IWF). Term frequency (TF) indicates the frequency of word in the document, and inverse word frequency (IWF) is a measure of the general importance of a word.

$$P_{TF\text{-}IWF}(i,j) = \frac{N_{i,j}}{\sum_k N_{k,j}} \times \log \frac{\sum_{i=1}^{m} S_i}{S_i} \tag{4}$$

In the TF part, $N_{i,j}$ is the number of occurrences of the word w_i in the document D_j, and the denominator $\sum_k N_{k,j}$ is the total number of occurrences of all words in the file D_j. D_j is the documents belonging to the j_{th} category in the corpus.

The S_i in the IWF represents the sum of the frequency of the word w_i in the corpus, $\sum_{i=1}^{m} S_i$ is the sum of the frequencies of all words in the corpus.

Second, we evaluate the relationship between words and documents by the importance of the words to the document with term frequency-inverse word frequency (TF-IWF) algorithm.

The TF-IWF algorithm determines the importance of words by calculating the distribution of words in the document. The word less appears in all document, the more appear in a topic, the word have greater impact to classification.

2.2 Building Heterogeneous Graph

WWD Matrix. For each node of the graph, we use a pre-trained vector based on word2vec as input features. In order to reduce the impact of infrequence words in documents on text, we filter out the stop words that appear very infrequently in the data. Then we construct text graph by PMI and TF-IWF algorithms. In this way, we embed the relationship between words and words, the relationship between words and documents to the model. And we assume that all nodes are self-connected and the value is set to 1. We represent graph as $G = (V, E)$, the nodes in the graph $|V|$ include word and document. And the edges $|E|$ is relationship of words and documents in the corpus.

We named the graph as the WWD matrix (word-word-document). The weight of edges is defined as follow:

$$W_{x,y} = \begin{cases} PMI(x,y), & x, y \text{ are words,} \\ & PMI(x,y) > 0 \\ P_{TF\text{-}IWF}(x,y), & x \text{ is document, } y \text{ is word .} \\ 1, & x = y \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Finally, the WWD matrix is taken as a part of DGRL method to model the semantic relations.

3 Deep Graph Residual Learning for Text Classification

GCN is a neural network that perform feature processing directly on the graph and get the node's embedding vector based on neighboring nodes around. Kipf and Welling [10] proposed a variant graph convolutional neural network which extends a graph convolution layer, enabling the model to utilize the relative information among words and documents, also they can use the properties of nodes (embedding vectors). GCN finds the neighborhood nodes for each node on the topology graph, and obtains the node information from neighborhoods to associate nodes with each other to achieve mutual characterization. However, the GCN only has two layers, it often does not capture the sufficient information from relative words and the representation of words. Otherwise, increasing the number of networks layers may cause the problem of gradient disappearance. He K. et al. propose deep residual learning for solving the problem of degradation [6]. In view of the great success of deep residual learning, we introduce residual module to graph convolutional networks [13]. It promotes the GCN achieving more reliable convergence results in deeper model and better performance in text classification. In the original graph convolutional network framework, it is essentially to construct a potential map F from the input data to target task. Here, we propose a graph residual network learning framework that learns a new underlying map I by fitting map F. After F transforming H_l, model perform vertex-wise addition on $F(H_l)$ and H_l^{res} to obtain H_{l+1}. The H_l^{res} is residual mapping of H_l. W_l is a set of learnable parameters of the l-th layer:

$$H_{l+1} = I(H_l, W_l) = F(H_l, W_l) + H_l^{res} \tag{6}$$

In the deep graph residual network, graph is represented as $G = (V, E)$, we assume that each node is connected to itself $(v, v) = E$, and n represents the number of nodes, m represents the dimension of the feature vector, that is, $X \in \mathbf{R}^{n \times m}$. Each row vector represents a feature vector of a vertex $x_v \in \mathbf{R}^m$. We introduce the proposed adjacency matrix WWD and degree matrix D, where $D_{ii} = \sum_j M_{ij}$. Combining with the residual network, we can calculate the output L of each GCN layer as:

$$L_n = \begin{cases} \sigma(\tilde{A}XW_{n-1}), & n = 1 \\ \sigma(\tilde{A}(L_{n-1} + L_{n-3}^{res})W_{n-1}), & (n-2) \bmod 2 \\ & = 0 \text{ and } n \neq 0 \\ \sigma(\tilde{A}L_{n-1}W_{n-1}), & \text{otherwise} \end{cases}. \tag{7}$$

where $\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix, W_n is a weight matrix, n is the number of layers. $L_0 = X$, L^{res} indicates that the residual blocks transform L to the same size as the target layer by identity mapping. σ is the activation function. In our method, we use the *LeakyRelu* activation function. At the last layer. We used the softmax classifier and set the node (word/document) embeddings of last layer to the same size as the labels set:

$$Z = softmax(\tilde{A}L_nW_n), \tag{8}$$

The $softmax(x_i) = \frac{1}{z}exp(x_i)$ with $z = \sum_i exp(x_i)$. We use the cross-entropy error over all documents which marked label to compute the loss function:

$$Loss = \sum_{s \in S_D} \sum_{f=1}^{F} Y_{sf} \ln Z_{sf}, \tag{9}$$

where S_D is the set of document indices which marked labels and F is the dimension of the output features equal to the number of classes. Y is the label indicator matrix.

The DGRL framework has been shown in Fig. 1. D_n denotes different documents of dataset, Word_n denotes different words of dataset, the color of the document represents the category. Firstly, we remove word segmentation, stop words, and some words that occur infrequently, then we construct the relationship between words and words, and words and documents into a relational graph. We increase the number of model network layers and keep the model stable by continuously stacking the residual modules.

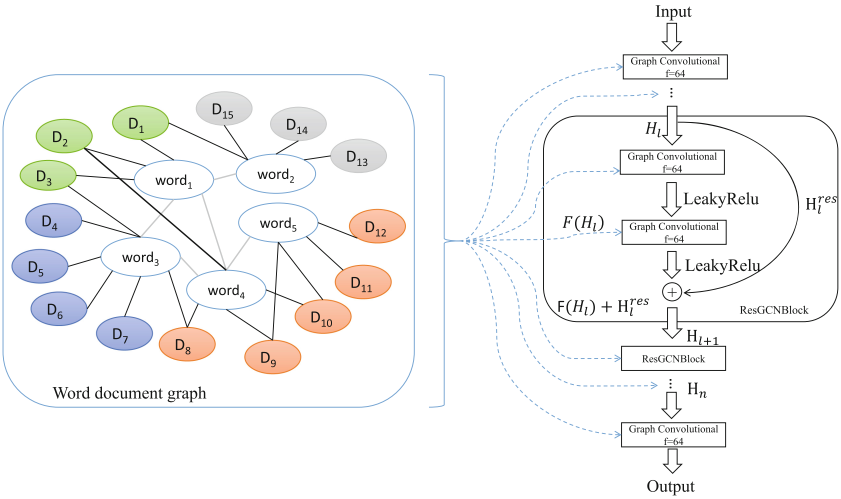


Fig. 1. An overview of the DGRL framework. Colors represent the various category of documents. We embed the word document graph to the GCN, and f is the number of hidden units. (Color figure online)

4 Experiment

4.1 Dataset

We performed experiments on 4 widely used benchmark corpora. These include Ohsumed, R52 and R8 of Reuters 21578 and Movie Review (MR). The distribution of the data sets is shown in Table 1.

Table 1. Summary statistics of datasets.

Dataset	Docs	Training	Test	Words	Classes
Ohsumed	7,400	3,357	4,043	14,157	23
R8	7,674	5,485	2,189	7,688	8
R52	9,100	6,532	2,568	8,892	52
MR	10,662	7,108	3,554	18,764	2

Ohsumed: The Ohsumed data set document is derived from the medical information database. In the experiment, we used the 13929 cardiovascular disease abstract data set from the first batch of 20,000 abstracts in 1991, and excluded documents belonging to multiple categories, and only retained a single category of 7,400 documents, 3357 documents in the training set, and 4043 documents in the test set [29].

R8 and R52: R8 and R52 are two subsets of the Reuters 21578 dataset. R8 has a total of 5485 training files and 2189 test files divided into 8 categories. R52 has 52 categories, divided into 6532 training files and 2568 test files.

MR: MR is a movie review data set for binary sentiment classification. Each comment contains only one sentence [19]. The data set is divided into 5331 positive reviews and 5331 negative reviews. We used the training/test split in Pte: Predictive text embedding through large-scale heterogeneous text networks [24].

Firstly, we preprocessed all datasets by cleaning and tagging the text [9], then we removed the stop words defined in NLTK and the low-frequency fields that appeared less than 5 times in the 20NG, R8, R52, and Ohsumed datasets. During to the documents in MR dataset are very short, we did not delete the words after cleaning and tagging the original text.

4.2 Baseline Model

We compare the DGRL with several advanced text classification and embedding methods in recent years. There is briefly introduction of the comparison algorithm.

TF-IDF+LR: TF-IDF+LR [29] uses word frequency inverse document frequency and bag-of-word to weighting words. And the logistic regression is used as a classifier.

CNN: We tested two different CNN [9] performances: a convolutional neural network model with random initialization word embeddings, and a convolutional neural network model with pre-trained word embeddings.

LSTM: LSTM [17] is a recurrent neural network. It has very good performance in text classification and is widely used. We respectively embed the random initialization words and pre-trained words to LSTM as comparative experiments and performance analysis.

Bi-LSTM: Bi-LSTM [7] is a variant of LSTM. It is a bidirectional LSTM and it is also commonly used for text classification. We embed pre-trained words into the Bi-LSTM model for text classification task.

TextRCNN: TextRCNN extend the RNN method for text classification task, and then use the convolution pooling operation to obtain the classification results of the samples [12].

PTE: PTE namely predictive text embedding [24]. This method first learns word embeddings based on heterogeneous text networks containing words, documents, and labels as nodes, and then get the average value of word embeddings as document embeddings for text classification.

FastText: FastText [8] is a simple and effective method for text classification. It takes the average value of the word/n-gram embedding as the representation of the document, and then it feds to the linear classifier by using a binary graph. And the n-grams model is use n-grams model as predict the specified category.

SMEM: SMEM [21] is a simple word embedding model that uses a simple out-of-pool strategy.

LEAM: LEAM [27] is a label embedding attention model, which embeds words and labels into the same joint space for text classification.

Graph-CNN-C: A graph convolutional neural network model [3] using Chebyshev filters. This model first embeds word similarity graphs and then performs convolution operations.

Text-GCN: Text-GCN [29] model also builds graph convolution network by establishing the relationship between words and words, and words and documents.

DGRL-2 layers: The model use TF-IWF and PMI algorithm to build the graph and uses the GCN [10] to classify text.

4.3 Parameter Settings

In this experiment, we used 300-dimensional word embeddings [20]. In order to take into account the robustness of the model in long text and short text data, we set the window size equal 20, the learning rate is set to 0.02 and the dropout rate is set to 0.3. In terms of data sets, the experiment randomly chooses 10% of the training set as the verification set. In order to prevent overfitting, we set the maximum training batch of the model to 400. If the verification loss of model does not reduce more than 10 consecutive batches, the model will early termination of training.

4.4 Test Performance

We run all models 10 times on document classification task and report mean ± standard deviation, the test accuracy of each model is given in Table 2. DGRL performs better than most of benchmark models on most data sets, which shows the effectiveness of the method on text data. After the pre-trained word embedding of each model, the performance of the model on the text data has significantly improved than that of the model with random initialization embedding. DGRL also obtain great performance when using pre-trained word embeddings to build similarity graphs of words, which shows that constructing similarity graphs of words can maintain the relationship between word nodes.

We use PMI and TF-IWF to construct the relationship between documents and words, then embed them in the deep graph residual learning model. We find that our method has better results. On the four data sets in the experiment, the classification performance is better than Text-GCN.

Table 2. Comparison chart of each comparison model accuracy

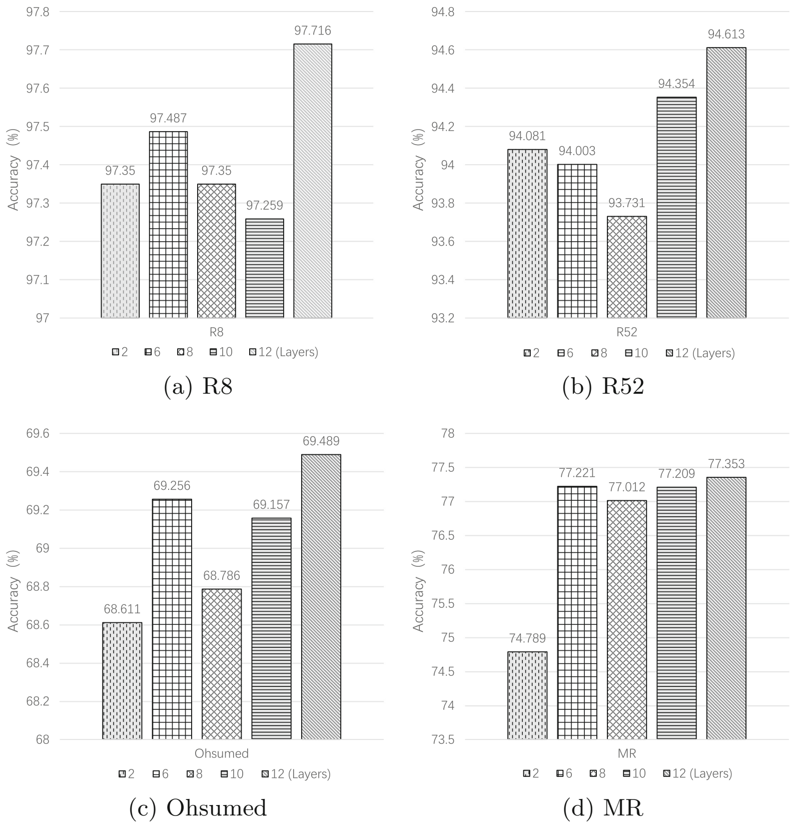
Model	R8	R52	Ohsumed	MR
TF-IDF+LR [29]	0.9374 ± 0.0000	0.8695 ± 0.0000	0.5466 ± 0.0000	0.7459 ± 0.0000
CNN-rand [9]	0.9402 ± 0.0057	0.8537 ± 0.0047	0.4387 ± 0.0100	0.7498 ± 0.0070
CNN-non-static [9]	0.9571 ± 0.0052	0.8759 ± 0.0048	0.5844 ± 0.0106	**0.7775 ± 0.0072**
LSTM [17]	0.9368 ± 0.0082	0.8554 ± 0.0113	0.4113 ± 0.0117	0.7506 ± 0.0044
LSTM(pretrain)	0.9609 ± 0.0019	0.9048 ± 0.0086	0.5110 ± 0.0150	0.7733 ± 0.0089
Bi-LSTM [7]	0.9631 ± 0.0033	0.9054 ± 0.0091	0.4927 ± 0.0107	0.7768 ± 0.0086
Text RCNN [12]	0.9553 ± 0.0015	0.9147 ± 0.0011	0.5869 ± 0.0020	0.6126 ± 0.0022
PTE [24]	0.9669 ± 0.0013	0.9071 ± 0.0014	0.5358 ± 0.0029	0.7023 ± 0.0036
fastText [8]	0.9613 ± 0.0021	0.9281 ± 0.0009	0.5770 ± 0.0049	0.7514 ± 0.0020
fastText(bigrams)	0.9474 ± 0.0011	0.9099 ± 0.0005	0.5569 ± 0.0039	0.7624 ± 0.0012
SWEM [21]	0.9532 ± 0.0026	0.9294 ± 0.0024	0.6312 ± 0.0055	0.7665 ± 0.0063
LEAM [27]	0.9331 ± 0.0024	0.9184 ± 0.0023	0.5858 ± 0.0079	0.7695 ± 0.0045
Graph-CNN-C [3]	0.9699 ± 0.0012	0.9275 ± 0.0022	0.6386 ± 0.0053	0.7722 ± 0.0027
Text GCN [29]	0.9707 ± 0.0010	0.9356 ± 0.0018	0.6836 ± 0.0056	0.7674 ± 0.0020
DGRL-2 layers	0.9735 ± 0.0010	0.9408 ± 0.0008	0.6861 ± 0.0026	0.7479 ± 0.0020
DGRL	**0.9772 ± 0.0008**	**0.9461 ± 0.0015**	**0.6949 ± 0.0037**	0.7735 ± 0.0020

4.5 Experimental Analysis

In the experiment, we plot different curve graph on the data sets to analyze our model. Figure 2 shows the change in the accuracy of the test set at different levels of the model. The abscissa is the network depth and the ordinate is the accuracy. Due to the limitation of computing power, the depth of the experimental model only reaches 12 layers. In order to show the effect of the residual network, it is not build the model consisting of a 4-layers network, and the 2-layers network do not use the residual block.

On the R8 dataset, the classification accuracy of the model with the residual block is higher than that of the model without the residual block. The accuracy of the 8-layers and 10-layers model has decreased, but the accuracy has increased significantly at the 12-layers models and has reached the local highest value. On the R52 data set, the model accuracy without residual block is higher than 6-layers and 8-layers models. But the accuracy is lower than the 10-layers and 12-layers models. On the Ohsumed and MR datasets, the model accuracy with the residual block is significantly improved compared with the model that without the residual block. Although the accuracy of the 8-layers model is slightly lower than that of the 6-layers model, the accuracy of the 8-layers, 10-layers, and 12-layers models shows a steady upward trend. As layers of the model deepens, the accuracy gradually increases.

Through comparison, it is obvious that the accuracy of the model after deepening the layer is more accurate than the model with the shallowest layer in most cases. Although the accuracy may occasionally decrease during the deepening process, in general, the accuracy of the model increases with the number of layers. Although the performance of the experimental equipment is limited, but according to the result, the stacking of the network layers can make the model perform better. Because the remaining network is mainly to solve the problem



(a) R8

(b) R52

(c) Ohsumed

(d) MR

Fig. 2. The effect of different layers on model accuracy

of information loss causing by the deepening of the number of layers, it makes the model stability while deepening the number of network layers, and made the deep residual network model better. When the model is not deep enough, it is difficult to reflect the advantages of the residual network, but the deeper the network, the more obvious the advantages of the residual network. It can be inferred that the accuracy of the model with only 12 layers has not reached saturation. As the number of network layers increases, the accuracy can be further improved.

4.6 Ablation Study

In order to study the benefits of DGRL components on model performance, we conducted ablation studies on DGRL.

The results shown in Table 3. There are the deep graph convolutional residual network which construct graph relationships by words similarity (GCN-S), the graph convolution network construct graph by TF-IWF algorithms (GCN-

TFIWF), the graph convolution network construct graph by TF-IWF and PMI algorithms (GCN-TFIWF+PMI), and the deep GCN-TFIWF+PMI with residual blocks as subjects conduct ablation studies.

First, we remove the relationship graph constructed by the TF-IWF and PMI algorithms, and construct a deep graph residual network with words relationship by calculating the similarity of words. There is no good performance on all data sets.

Then we use the TF-IWF algorithm to construct the relationship graph (GCN-TFIWF). The experimental results show that this model has a good performance on each data set, which shows that the construction of relationship between documents and words is important to the model, and it also verifies the importance of the TF-IWF algorithm in our method. At the same time, we used the TF-IWF algorithm and the PMI algorithm to construct graph relationships for graph residual network (GCN-TFIWF+PMI), compared with GCN-TFIWF, the performance of the model has obvious improved, it also reflects the role of PMI algorithm in DGRL. Finally, we deepen the network with residual module base on GCN-TFIWF+PMI (GCN-TFIWF+PMI+Residual), the accuracy of the model has been further improved, it proves the importance of the residual module in DGRL.

Table 3. Comparison of ablation experiment results

Model	R8	R52	Ohsumed	MR
GCN-S	0.28278	0.28544	0.06876	0.50253
GCN-TFIWF	0.96482	0.92718	0.65051	0.73415
GCN-TFIWF+PMI	0.96803	0.93804	0.67188	0.75213
GCN-TFIWF+PMI+Residual	0.97541	0.94307	0.69433	0.76954

4.7 Case Study

In order to better understand the performance of DGRL, there is comparison of several test case with LSTM [17] and CNN [9] as the comparison model to predict the sample. In the R52 dataset, "dauster says consumers should keep out of quotas consumer countries should not intervene in the distribution of coffee export quotas Brazilian coffee institute president Jorio Dauster said distribution of export quotas should be in the hands of producers." LSTM did not successfully predict the subject of the sample, CNN predicted the sample topic accurately, DGRL calculates the weight of "coffee" through TF-IWF, draws the relationship between the sample and the category, and successfully predicts the topic of this sample.

The case "this attributed the rise in consumer prices to the effects of the February drop in energy prices working their way out of the index, the February increase was less than January increases, but slightly above the average for the later months of inflation in the western industrialised." LSTM [17] and CNN [9] did not successfully predict the category of the article. DGRL can effectively find out which words or combination of words have higher weight in the article through TF-IWF and PMI algorithm, and mine related words through graph convolution network to increase the prediction weight of the corresponding category. The remarkable experimental performance also proves the superiority of the DGRL.

5 Related Work

Many scholars have introduced different neural network concepts to text classification. Miyato et al. extend adversarial and virtual adversarial training to the text domain by applying perturbations to the word embeddings in a recurrent neural network rather than to the original input itself [18]. Kowsari et al. employs stacks of deep learning architectures to provide specialized understanding at each level of the document hierarchy [11].

In recent years, neural networks based on semantic have gradually replaced shallow models such as bag-of-words models and become new hotspots. Neural networks model over word sequences such as CNN and RCNN have achieved promising performances in text classification, but word sequences is not the only factor that improves the classification effect. It is also very important to capture the long-distance relationship of words and the relevance of words. Tai et al. proposed an adaptive recurrent neural network based on dependency trees [23], but the model is based on dependency trees, the long contextual semantics of text is prone to biased weights. We adopt graph convolutional networks to overcome this limitation.

GCN has attracted more and more attention in the area of artificial intelligence and has been applied to natural language processing. Linmei et al. [16] proposed a flexible heterogeneous information network for modeling text and adds an attention mechanism base on dual-level attention mechanism, including node-level and type-level to GCN. Zhang [30] utilize GCN to learn explicit relational knowledge, obtaining relational knowledge through KG Embeddings and GCN, and it combines instance encoder and knowledge-aware attention to extraction long-tail relation. Vashishth et al. extract semantic by LSTM and dating documents and relationship classification by GCN [26]. Our method constructs a novel heterogeneous graph based on the relevance of words and the relationship between words and documents in order to capture information from words and documents effectively.

Compared with the above method, the deep graph residual learning (DGRL) method takes more concise way to construct graph, and obtains the sufficient representation information, the model achieves more effective text classification.

6 Conclusion and Future Work

In this study, we propose an improved text classification method called text classification with deep graph residual learning (DGRL). We first built a new heterogeneous graph for the corpus and classified text by node classification. We found that this method has better ability to capture global features and has better adjustment ability, and the performance of the model is improved in comparison with the methods that in text classification in recent years.

In addition to use the pre-trained word embedding deep graph residual learning model to classify and summarize text, there are many places worthy of our attention to improvement. For example, in this method, our pre-trained word embedding needs to be associated with the label of the document. How to construct a text relationship graph with hidden attributes without labels? This is a question worth thinking about.

Acknowledgments. This work was supported by the Natural Science Foundation of China (No: 81701780 and 61672177); the Project of Guangxi Science and Technology (No: GuiKeAD17195062, GuiKeAD19110133 and GuiKeAD20159041); the Guangxi Collaborative Innovation Center of Multi-Source Information Integration and Intelligent Processing; the Research Fund of Guangxi Key Lab of Multi-source Information Mining and Security (No: 20-A-01-01); Innovation Project of Guangxi Graduate Education (No: JXXYYJSCXXM-012 and JXXYYJSCXXM-011).

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**(Jan), 993–1022 (2003)
2. Bouma, G.: Normalized (pointwise) mutual information in collocation extraction. In: Proceedings of GSCL, pp. 31–40 (2009)
3. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems, pp. 3844–3852 (2016)
4. Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: Adaptive recursive neural network for target-dependent twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 2: Short papers, pp. 49–54 (2014)
5. Ganguly, D., Roy, D., Mitra, M., Jones, G.J.: Word embedding based generalized language model for information retrieval. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 795–798 (2015)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
7. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
8. Joulin, A., Grave, É., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, vol. 2, Short Papers, pp. 427–431 (2017)
9. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751 (2014)

10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: In ICLR (2016)
11. Kowsari, K., Brown, D.E., Heidarysafa, M., Meimandi, K.J., Gerber, M.S., Barnes, L.E.: HDLTex: hierarchical deep learning for text classification. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 364–371. IEEE (2017)
12. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
13. Li, G., Muller, M., Thabet, A., Ghanem, B.: DeepGCNs: can GCNs go as deep as CNNs? In: Proceedings of the IEEE International Conference on Computer Vision, pp. 9267–9276 (2019)
14. Li, Q., Han, Z., Wu, X.M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
15. Lin, Y.S., Jiang, J.Y., Lee, S.J.: A similarity measure for text classification and clustering. IEEE Trans. Knowl. Data Eng. **26**(7), 1575–1590 (2013)
16. Linmei, H., Yang, T., Shi, C., Ji, H., Li, X.: Heterogeneous graph attention networks for semi-supervised short text classification. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 4823–4832 (2019)
17. Liu, P., Qiu, X., Huang, X.: Recurrent neural network for text classification with multi-task learning. In: IJCAI (2016)
18. Miyato, T., Dai, A.M., Goodfellow, I.: Adversarial training methods for semi-supervised text classification, stat 1050, 7 (2016)
19. Pang, B., Lee, L.: Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 115–124 (2005)
20. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
21. Shen, D., et al.: Baseline needs more love: on simple word-embedding-based models and associated pooling mechanisms. In: Proceedings of the 56th Annual Meeting of the ACL, vol. 1: Long Papers, pp. 440–450 (2018)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Computer Vision and Pattern Recognition (2014)
23. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 1556–1566 (2015)
24. Tang, J., Qu, M., Mei, Q.: PTE: predictive text embedding through large-scale heterogeneous text networks. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1165–1174 (2015)
25. Trstenjak, B., Mikac, S., Donko, D.: KNN with TF-IDF based framework for text categorization. Procedia Eng. **69**, 1356–1364 (2014)
26. Vashishth, S., Dasgupta, S.S., Ray, S.N., Talukdar, P.: Dating documents using graph convolution networks. In: Proceedings of the 56th Annual Meeting of the ACL, vol. 1: Long Papers, pp. 1605–1615 (2018)
27. Wang, G., et al.: Joint embedding of words and labels for text classification. In: Proceedings of the 56th Annual Meeting of the ACL, vol. 1: Long Papers, pp. 2321–2331 (2018)

28. Wang, X., Yang, L., Wang, D., Zhen, L.: Improved TF-IDF keyword extraction algorithm. Comput. Sci. Appl. **3**(1), 64–68 (2013)
29. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for text classification. Proc. AAAI Conf. Artif. Intell. **33**, 7370–7377 (2019)
30. Zhang, N., et al.: Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long and Short Papers), pp. 3016–3025 (2019)
31. Zhang, S., Li, X., Zong, M., Zhu, X., Cheng, D.: Learning k for kNN classification. ACM Trans. Intell. Syst. Technol. (TIST) **8**(3), 1–19 (2017)
32. Zhang, S., Li, X., Zong, M., Zhu, X., Wang, R.: Efficient kNN classification with different numbers of nearest neighbors. IEEE Trans. Neural Netw. Learn. Syst. **29**(5), 1774–1785 (2017)
33. Zhu, X., Huang, Z., Yang, Y., Shen, H.T., Xu, C., Luo, J.: Self-taught dimensionality reduction on the high-dimensional small-sized data. Pattern Recogn. **46**(1), 215–229 (2013)
34. Zhu, X., Li, X., Zhang, S.: Block-row sparse multiview multilabel learning for image classification. IEEE transactions on cybernetics **46**(2), 450–461 (2015)
35. Zhu, X., Li, X., Zhang, S., Ju, C., Wu, X.: Robust joint graph sparse coding for unsupervised spectral feature selection. IEEE Trans. Neural Netw. Learn. Syst. **28**(6), 1263–1275 (2016)
36. Zhu, X., Li, X., Zhang, S., Xu, Z., Yu, L., Wang, C.: Graph pca hashing for similarity search. IEEE Trans. Multimedia **19**(9), 2033–2044 (2017)
37. Zhu, X., Suk, H.I., Wang, L., Lee, S.W., Shen, D., Initiative, A.D.N., et al.: A novel relational regularization feature selection method for joint regression and classification in AD diagnosis. Med. Image Anal. **38**, 205–214 (2017)
38. Zhu, X., Zhang, L., Huang, Z.: A sparse embedding and least variance encoding approach to hashing. IEEE Trans. Image Process. **23**(9), 3737–3750 (2014)

Densely Connected Bidirectional LSTM with Max-Pooling of CNN Network for Text Classification

Qinghong Jiang[1](✉), Huaping Zhang[1], Jianyun Shang[1], Ian Wesson[1], and ENlin Li[2]

[1] Beijing Institute of Technology, Beijing, China
jocelyn0709@163.com, {kevinzhang,shangjia,IanWesson}@bit.edu.cn
[2] Training Management Department of the Central Military Commission, Beijing, China

Abstract. Text classification is a fundamental task in natural language processing (NLP). Context semantics can greatly improve the accuracy of text classification tasks. Although there are some popular methods in obtaining semantics, current context semantic analysis techniques, due to limited accuracy, are still a great bottleneck for text classification. This paper introduces a novel model, the densely connected Bidirectional LSTM with Max-pooling of CNN network (Dense-BiLSTM-MP), which greatly enhances the context of semantic information. In this model, a densely connected bidirectional long short-term memory (BiLSTM) model, as well as multiple max-pooling layers of convolutional network, are applied to obtain an increasingly enhanced assessment of context, and extract the key features, respectively. Experiments were conducted on four public datasets: YELP, 20NewsGroup, THUNews and AG. The experimental results show that the proposed model outperforms state of the art methods on several datasets. Furthermore, discussions on the Dense-BiLSTM-MP model's performance in short texts and long texts were given, respectively.

Keywords: Text classification · Dense structure · Deep learning

1 Introduction

With the emergence of the big data era, text has become an important medium of information sharing and dissemination, such as news articles, product reviews, social media posts, etc. How to classify these texts is a large area of research. Many existing methods, such as the TF-IDF weight algorithm, cannot obtain the semantics. Although CNN [13] has achieved good results in text classification, it directly extracts features from texts without considering the context semantics. Usually, the context semantics can greatly help improve the accuracy of text classification tasks.

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 98–113, 2020.
https://doi.org/10.1007/978-3-030-65390-3_8

Many methods, like in the cases of [15,19,29], have made propositions that could enhance context semantics. Among them, the Bidirectional LSTM (BiLSTM) networks [6] are proven to achieve good results. In recent years, with the increase of research into deep network structures, such as Highway Networks [31], ResNet [7,14] and Densely connected CNNs [9], Densely Bidirectional LSTMs [4,26,33] which mainly focused on a dense connection between words, have been proposed as a solution to deal with a variety of NLP tasks, and have achieved flawless results.

The BiLSTM uses memory mechanism to obtain context semantics. The CNN structure uses the convolutional filter which needs a lot of parallel computation to obtain local semantics. The max-pooling layer of CNN can automatically extract key features. Hence, to take advantage of RNN and CNN structures, as well as to reduce the time complexity, we use a BiLSTM structure and a max-pooling layer of CNN.

To capture deeper context semantics, this paper used a densely connected BiLSTM structure. This was to ensure maximum information flow between layers in the network and maximally obtain the context semantics of texts. Additionally, a max-pooling layer of CNN model following every BiLSTM structure was used to extract unbiased features in texts. In the densely connected BiLSTM structure, the inputs of every BiLSTM block is the outputs of all the earlier BiLSTM blocks. Meanwhile the information flow between layers in the network greatly alleviates the vanishing gradients problem. To make good use of BiLSTM structure, we broke down the context features of each word when passing each layer's semantic features to later layers.

Although there are several densely connected structures, like dense CNNs [25], dense BiLSTMs [4], hierarchical RNNs [33], and dense networks [12,16,26, 28], the Dense-BiLSTM-MP model proposed by this paper is rather different from them.

In contrast with the DC-Bi-LSTM model proposed by [4], this paper select a max-pooling layer of a CNN for feature selection to extract key latent semantic features, which make use of each layers' output. The DC-Bi-LSTM model used average pooling, with which only a few words and their combination are useful for capturing the meaning of the text [15]. Additionally, there are differences in the connection method of features, which break down the context features of each word. To prove that breaking down the context features of each word help improve accuracy in the dense structure, comparison experiments were conducted in Sect. 4.4 And results showed that breaking down the context features of each word help greatly improve accuracy in the dense structure. Comparison experiments between the DC-Bi-LSTM model and the proposed model were conducted in this paper, and the proposed model achieved higher accuracy.

Wang [25] proposed a densely connected CNN model with multi-scale feature attention, which used densely connected CNNs with different kernel sizes to extract multi-scale features, which enables the model to produce variable n-gram features. Although the model they proposed can obtain key features through the multi-scale feature attention method, they are not very effective at extracting deeper context semantics which can greatly help improve the accuracy of text classification tasks. In contrast with their model, the model proposed in this paper mainly focuses on mining deeper context semantics of multi-scale semantics.

In contrast with [12,16,26,28,33], which mainly focused on a dense connections between different words in a sentence, which enhances the semantics of words by improving long time memory ability. The Dense-BiLSTM-MP model proposed by this paper focuses on dense connections between features of the same word in different layers of a sentence, and uses a max-pooling layer to extract key semantic features, which enhances semantics of words.

In summary, the contributions of this paper include:

- This paper propose a novel model Dense-BiLSTM-MP that is equipped with dense connections between BiLSTM layers with a max-pooling layer of CNN model after every BiLSTM layer, to extract unbiased features of texts. With dense connections, the largest set of semantic features is obtained, and when passing each layer's semantic features to later layers, we break down the context features of each word, which is different from other dense structures. Although there are several researches on breaking down the context features of each word, we are first to give comparison experiments and analyses on whether breaking down the context features of each word in the densely connected structures help improve the accuracy rate of the text classification task. With every max-pooling layer of CNN model, the most important set of semantic features is extracted.
- Experiments on five text classification tasks were conducted. Results show that the proposed model outperforms the state-of-the-art methods on several datasets.

2 Related Work

2.1 Text Classification

Many text classification algorithms have emerged, ranging from statistical rules and machine learning algorithms to deep learning algorithms, for example: the n-gram model, TF-IDF algorithm [24], naive Bayes (NB), support vector machine (SVM) [20,27] and CNNs [13,25,30]. Conventional researches on text classification mainly focus on feature selection, such as [1,3,5,18,32]. Feature selection is to select the elements that best represent the meaning of the text, and it plays an important role in text classification. Feature selection can not only reduce the scale of the problem, but also improve the performance of classification tasks. Conventional feature selection algorithms consists of mutual information, information gain and Chi-square, et al. Machine learning algorithms consists of naive Bayes (NB) and support vector machine (SVM), et al. Traditional feature selection algorithms, machine learning algorithms and deep learning algorithms can all extract the features of the text well in some fields, but these usually lack semantic extensions. The difficulty of text classification is how to capture useful information more effectively. RNN structures with memory mechanisms like [6,8,10,15,17,21], have been shown to help obtain the semantics of text during classification.

Different from Huynh [10], Song [21], Lee [17], this paper proposed a novel model that the semantics are directly passed to latter layers with a densely connected structure, which can make use of each layer's semantics. And we select a max-pooling layer of a CNN to extract each layer's key semantic features.

2.2 Densely Network

Huang [9] first proposed a Dense Convolutional Network (DenseNet), which connects each layer to every other layer in a feed-forward fashion. For each layer, all the preceding layers are used as inputs, and each layers own outputs are used as inputs into all subsequent layers. By applying information flow directly into subsequent layers, the vanishing gradients problem is alleviated, feature propagation is strengthened, and feature reuse becomes encouraged. Before the Dense Convolutional Network, CNNs such as CNN [13], GoogLeNet [23], Highway Networks [22], ResNets [7] had emerged, as the structures of the convolutional neural network become increasingly deep and the vanishing gradients problem still existed. ResNets [7]and Highway Networks [22] pass signal from one layer to the next layer via directly connections which can allieviate the vanishing gradients problem. Densely network [9] with directly dense connections between convolutional layers can not only help allieviate the vanishing gradients problem but also enhance the parameter reuse.

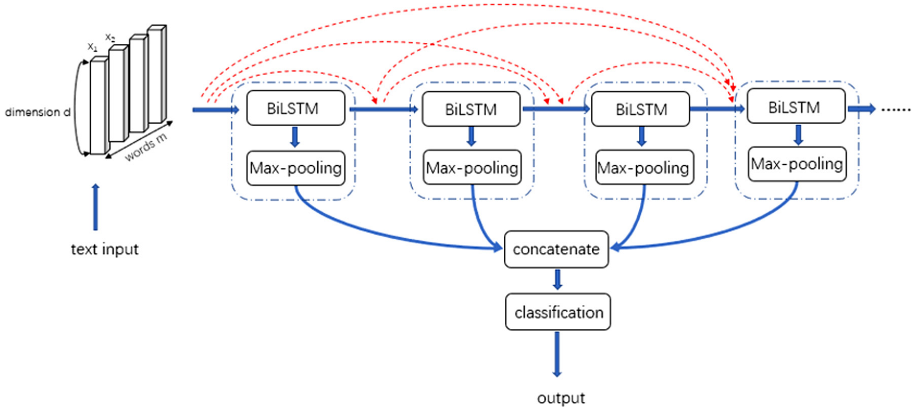


Fig. 1. The architecture of Dense-BiLSTM-MP model

3 Method

The framework of the proposed model is shown in Fig. 1. It mainly consists three (3) parts: (1) the input part, (2) the densely connected Bidirectional LSTM with max-pooling of CNN Network, and (3) the classification part.

For the first part, a text $x_1, x_2, ,$ with length m, and feature dimension d of every words embedding vector, is inputted into the network.

In the second part, the densely connected Bidirectional LSTM with max-pooling of CNN Network consists of many intermediate layers. Each intermediate layer is represented by the blue dotted line, which contains a BiLSTM part and a max-pooling part. A recurrent structure can greatly enhance the processing of context semantics, especially for long texts. BiLSTM is one of the recurrent structures, and has the ability to acquire the context semantics well by using memory mechanisms. Max-pooling layers well suited for extracting the key features of the vector input. A BiLSTM structure is used, followed by a max-pooling layer in each intermediate layer to obtain the semantics of the text. For each intermediate layer, when a text is inputted into the model, each BiLSTM will give a semantics vector output. That output is then fed into a max-pooling layer to get features. The concatenation of the first layer's input and all the prior layers output will become the input of the subsequent layers. In particular, the first intermediate layer's input is the word embedding vector of a text described in the first part, the text's length equals m and feature dimension equals d.

After all the features with the max-pooling layers are gathered, a concatenation operation will be done before the final classification part. For the third part, a linear layer and softmax layer are later used to get the probability of classification. In the densely connected Bidirectional LSTM with max-pooling of CNN Network, this paper mainly aims to get deeper context semantics.

3.1 The Inputs Part

The proposed model is illustrated in Fig. 1. Let $x_i \in R^d$ be the d-dimensional pretrained word vector of the i-th word in a text, and the text's max-length is m. Mean that if the text's length is less than m, then padding it with a fixed word, and if the text's length is more than m, then truncating to m. The input text for the first intermediate layer can be represented as a matrix X:

$$X = [x_1, x_2, x_3, ..., x_m]_{m \times d} \tag{1}$$

3.2 The Densely Connected Bidirectional LSTM with Max-Pooling of CNN Network

Let L be the number of intermediate layers, and l $(1 \leq l \leq L)$ be the layer index. The implementation of intermediate layers is illustrated in Fig. 2. Let

$[x_1, x_2, x_3, , x_m]$ be the input of BiLSTMs, and the outputs of each BiLSTM model be represented as:

$$B_l = [b_l^1, b_l^2, b_l^3, ..., b_l^m]_{m \times 2k} \tag{2}$$

Where k denotes the hidden size of each BiLSTM network, thus $2k$ is the output size of each BiLSTM network. l denotes the layer index. We apply a max-pooling layer to extract the features of semantics, which are generated by the BiLSTM layer, and implement a $[p \times 2k]$ ($p < m$ and 2k is the dimensional of the output of each BiLSTM network) max-pooling matrix with a stride equals to 1 and a padding equals to 0. The output of the l-th intermediate layer can be represented as a q-dimentional vector:

$$C_l = (c_l^1, c_l^2, c_l^3, , c_l^q) \tag{3}$$

$$q = m - p + 1 \tag{4}$$

For the input of each layer l, $X_l = [x_l^1, x_l^2, x_l^3, , x_l^m]_{m \times n}$, $n = d + (l-1) \times 2k$. The input of the first intermediate layer is the network inputs in Eq. (1), and when the layer index $l \geq 2$, the l-th layer's input X_l, and the i-th word embedding vector $x_l^i \in R^n$ of layer l can be represented as follows:

$$X_l = f_{c1}(X_1, B_1, B_2, , B_{l-1}) \tag{5}$$

$$x_l^i = (t_{l-1}^i, t_{l-2}^i, , x_1^i, , r_{l-2}^i, r_{l-1}^i)_n \tag{6}$$

$$b_{l-1}^i = (t_{l-1}^i, r_{l-1}^i)_{2k} \tag{7}$$

Where f_{c1} denotes concatenation of the first layer input and outputs of all upstream layers, $b_{l-1}^i \in R^{2k}$ is the i-th word's semantic vector in the BiLSTM model's output of $(l-1)$-th layer. t_{l-1}^i and r_{l-1}^i are the forward direction context semantic features and the reverse direction context semantic features, respectively.

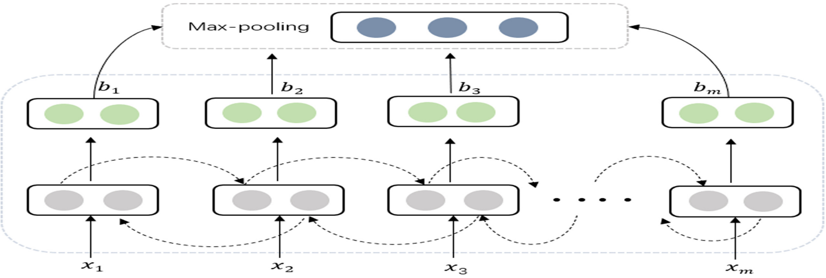


Fig. 2. The architecture of intermediate layer

3.3 The Classification Layer

The input of the classification layer is the concatenation of all the intermediate layers' output, which can be represented as:

$$C = f_{c2}([C_1, C_2, C_3, , C_L]) \tag{8}$$

Where f_{c2} denotes directly concatenation of all the outputs of the intermediate layers. $[C_1, C_2, C_3, , C_L]$ are represented in Eq. (3). After a linear layer and softmax later, we can get the probability of classification.

4 Experiments

4.1 Datasets

Experiments were conducted on five text classification tasks of four public datasets. Table 1 shows the statistics of all the following datasets:

Yelp: Our experiments used Yelp F. from Zhang [30]
20News-20: The corpus is on the web[1]. We used the by-date version of the twenty-newsgroup dataset and sorted training and testing samples of each class from small to large by text size, and then used 900 for training, and used 100 for the testing of each class respectively.
20News-4: The corpus is based on 20News-20 dataset proposed by this paper. We selected 4 major categories: comp, politics, rec, and religion from 20News-20 dataset. Following are the newsgroups in each selected category.

1. **comp**: comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x
2. **politics**: talk.politics.misc, talk.politics.guns, talk.politics.mideast
3. **rec**: rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey
4. **religion**: talk.religion.misc, alt.atheism, soc.religion.christian

AG: The AGs corpus of news articles was obtained on the web[2]. It contains 496,835 categorized news articles from more than 2000 news sources. 4 classes: World, Sci/Tech, Business and Entertainment, from this corpus were chosen to construct the AG dataset, using only the description fields. The number of training samples and testing samples of each class are 30,000 and 1900, respectively.
THUNews: The corpus was obtained on the web[3]. The THUCNews.zip file consists of 14 classes, including sport, entertainment, etc. The experiments used 114781 as training datasets and 13512 as testing datasets. The THUNews datasets is a Chinese corpus, and this paper used ICTCLAS which is a Chinese word segmentation system.

[1] http://qwone.com/~jason/20Newsgroups/.
[2] http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html.
[3] http://thuctc.thunlp.org/message.

Table 1. Data statistics.

Dataset	Yelp	20News-20	20News-4	AG	THUNews
Training	650k	18k	13.6k	120k	114.78k
Testing	50k	2k	1.36k	7.6k	13.51k
Classes	5	20	4	4	14
Avg-words	153	367	381	41	165

4.2 Experiment Setup

We pre-trained word vectors with a skip-gram model of word2vec, with 200 as the size of word embedding. Different datasets have different lengths of input text, where AG was 100, Yelp and THUNews were 200, and the others were 400. For the architecture, this paper uses stochastic gradient descent with a batch of 128. The initial learning rate is 0.01, and was decreased by 5% every 10 epochs. "$L2$" regularization was applied, and the momentum was set to be 0.78, with a decay weight of 0.001.

The hidden size of each BiLSTM model in the intermediate layers was 100, the number of each BiLSTM's hidden layer was 2, and the dropout rate of each BiLSTM model was 0.8. A $[p \times 200]$ max-pooling matrix with a stride equals to 1 and a padding equals to 0 was used, with p denotes to the length of input text minus 2.

The number of layers were 3 in our experiment, and accuracy of the results were calculated to be the number of correctly predicted samples, divided by the total number of samples in the test set.

4.3 Comparison Experiment

The traditional statistics algorithm and deep learning algorithms, which include the TF-IDF weight algorithm, CNN [13], BiLSTM [6], RCNN [15], CNN+SVM [2], DC-Bi-LSTM [4], and SR-RCNN [11], were used as a comparison with the proposed model.

The DC-Bi-LSTM model is a dense structure based on BiLSTM models. To compare with the Dense-BiLSTM-MP model, a layer index of 3 is used.

The SR-RCNN model is a hierarchical model based on CNN and RNN models. To compare with the Dense-BiLSTM-MP model, a layer index of 3 is used.

Additionally, a Stacked-BiLSTM-MP model were compared with Dense-BiLSTM-MP model. The Stacked-BiLSTM-MP model was also proposed by this paper.

The difference between the Dense-BiLSTM-MP model and the Stacked-BiLSTM-MP model is that each intermediate layers input is the output of its prior layer instead of the combination of the first layers input and outputs of all upstream layers. The main purpose of the proposed Stacked-BiLSTM-MP model is to observe whether the densely connected structure has a better performance in obtaining the context semantics.

Table 2. Accuracy of all the models on the five text classification tasks.

Model	Yelp	20News-20	20News-4	AG	THUNews
TF-IDF	54.8	82.3	92.7	57.31	91.5
CNN	**60.53**	90.5	98.46	63.78	98.85
CNN+SVM	58.16	86.2	89.72	62.51	97.8
BiLSTM	54.14	92.31	97.21	62.65	**99.98**
RCNN	55.67	93.39	97.29	61.67	99.93
DC-Bi-LSTM	55.33	94.75	98.23	63.64	99.32
SR-RCNN	49.28	97.1	98.97	63.3	98.78
Stacked-BiLSTM-MP	60.12	94.17	97.31	61.93	99.56
Dense-BiLSTM-MP	60.37	**97.16**	**99.34**	**65.75**	**99.98**

4.4 Results and Model Analysis

The results are listed in Table 2. Compared to previous methods on five text classification tasks, the results of the proposed method outperforms other methods on several datasets. The accuracy of Yelp is 60.37%, which is a bit lower than the CNN [13] model, but better than the results of other models. In the 20News-20 dataset, the accuracy of the proposed model is 97.16%, which is nearly 4% points higher than the RCNN model accuracy and a bit higher than SR-RCNN model. In the 20News-4 dataset, the accuracy of the proposed model is 99.16%, which is more accurate than all the other models. In the AG dataset, the accuracy of the proposed model is 65.75%, the result also outperforms other models. At last, in the THUNews dataset, the proposed model also has a good performance. All the results showed that the proposed model performs better in several datasets in the text classification tasks.

Regarding the structure of the corpora: the experimental results show that the performance advantage is obvious due to the baseline in small sample sizes and short text, but there is no obvious advantage in larger sample sizes, such as the YELP dataset. The proposed model makes the most of context semantics by concatenating all the upstream layers output, which can be mining deeper semantics. This is especially useful for some insufficient training samples and short text, such as the 20News-20 dataset and the 20News-4 dataset and the AG dataset.

To prove the proposed model can acquire more semantics with a densely connected structure, and that the proposed model is more accurate, this paper compared a Stacked-BiLSTM-MP model with the proposed model on AG datasets. The layer index of the Stacked-BiLSTM-MP model and Dense-BiLSTM-MP model is 3.

The Stacked-BiLSTM-MP model consists of stacked BiLSTMs, but each BiLSTM model's input is the output of the previous BiLSTM instead of all the upstream layers' output. Thus the Stacked-BiLSTM-MP model is not a densely connected structure. The purpose is to show that the densely connected structure can capture more semantics from the text.

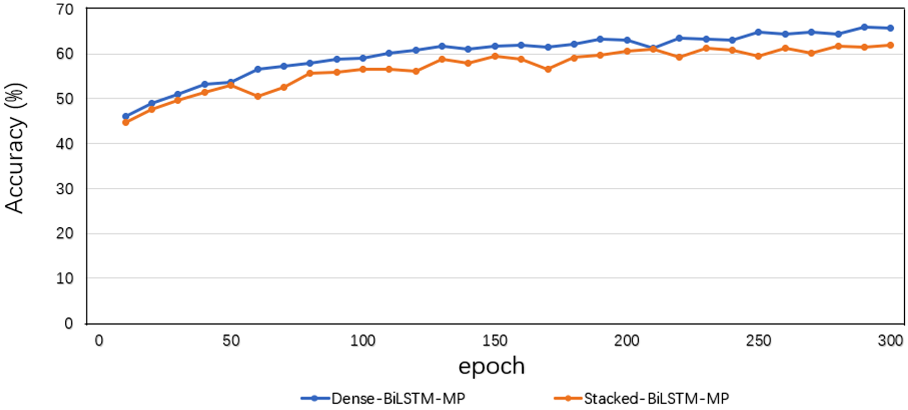


Fig. 3. The accuracy rate of the Dense-BiLSTM-MP model and the Stacked-BiLSTM-MP model on the AG dataset as the number of iterations increases

Figure 3 shows that as the number of iterations increases, the proposed model outperforms the Stacked-BiLSTM-MP model, and that the accuracy rate of the proposed model is consistently higher than the Stacked-BiLSTM-MP model. This proves that Dense-BiLSTM-MP model can obtain deeper context semantics during the process of text classification.

When passing semantic features to later layers in the Dense-BiLSTM-MP model, we broke down the context features of each word, with the purpose of combining left context of each word and right context of each word, respectively. To prove that breaking down the context features of each word in the densely connected structures help improve the accuracy, this paper compared a Dense-BiLSTM-MP-DP model with the Dense-BiLSTM-MP model on AG datasets. The Dense-BiLSTM-MP-DP model was also proposed by this paper. The difference between them is that the Dense-BiLSTM-MP-DP model directly passing context features to later layers without breaking them down.

The result is shown in Table 3. With each same layer, the accuracy rate of the Dense-BiLSTM-MP model has a higher accuracy rate than the Dense-BiLSTM-MP-DP model. The Dense-BiLSTM-MP model achieved the highest accuracy rate in the layer index 6, which is 1.41% higher than layer index 6 of the Dense-BiLSTM-MP-DP model.

Results show that breaking down the context features of each word when passing semantic features to later layers in a dense structure help achieve a higher accuracy rate. And to further prove it, we list the most important words which are most frequently selected in the max-pooling layer in Table 4. Comparison experiments were conducted on the AG dataset. The layer index of the Dense-BiLSTM-MP model and Dense-BiLSTM-MP-DP model is 3. In the category

Table 3. Accuracy of the Dense-BiLSTM-MP model and the Dense-BiLSTM-MP-DP model on the AG dataset with layer increases.

Layer	Dense-BiLSTM-MP	Dense-BiLSTM-MP-DP
2	62.09	62.01
3	65.75	63.84
4	67.77	66.57
5	67.84	66.05
6	**68.47**	**67.06**

Sci/Tech, the most important words selected by the Dense-BiLSTM-MP model and the Dense-BiLSTM-MP-DP model are Grid,Computing,Microsystem and Grid,Computing,on-demand, respectively. In the category Business, the most important words selected by the Dense-BiLSTM-MP model and the Dense-BiLSTM-MP-DP model are trade,business,market and markets,US,business, respectively. Words selected by the Dense-BiLSTM-MP model represent the category better.

Table 4. Comparison of the most important words selected by the Dense-BiLSTM-MP model and the Dense-BiLSTM-MP-DP model.

	Dense-BiLSTM-MP	Dense-BiLSTM-MP-DP
Sci/Tech	Sun Offers Pay-for-Use **Grid Computing** Sun **Microsystems** has introduced a pay-for-use pricing model for grid computing enabling customers to gain access to computing cycles on an on-demand basis	Sun Offers Pay-for-Use **Grid Computing** Sun Microsystems has introduced a pay-for-use pricing model for grid computing enabling customers to gain access to computing cycles on an **on-demand** basis
Business	Asian stock markets cheer quot pro-business quot Bush lead SINGAPORE Asian stocks rose Wednesday on indications that US President George W Bush whose **trade** and **business** policies are seen as more **market** friendly in the region will be re-elected	Asian stock **markets** cheer quot pro-business quot Bush lead SINGAPORE Asian stocks rose Wednesday on indications that **US** President George W Bush whose trade and **business** policies are seen as more market friendly in the region will be re-elected

4.5 Layer Analysis

To further prove that the densely connected Bidirectional LSTM with max-pooling of CNN Network can get more accurate results by using a densely connected structure, experiments were conducted on the Stacked-BiLSTM-MP model and the Dense-BiLSTM-MP model with different layers. The comparison experiments were conducted on the AG dataset.

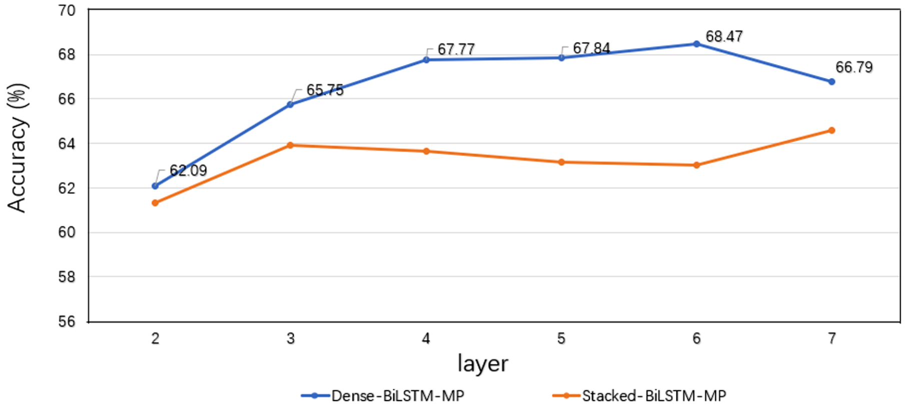


Fig. 4. The accuracy rate of the Dense-BiLSTM-MP model and the Stacked-BiLSTM-MP model on the AG dataset as the number of layers increases

The results is shown in Fig. 4, For the AG dataset, as the layers increase, the accuracy rate increases gradually. With each same layer, Dense-BiLSTM-MP has a higher accuracy rate than the Stacked-BiLSTM-MP model. However, when it reaches to the 7 layer, the accuracy rate decreases, which means that the model may not perform very well when there are too many layers. The most accurate result on the AG dataset occurs when the layer index equals 6. The accuracy with the layer index of 6 is 68.47%, which is nearly 3% points higher than the layer index of 3.

To further prove that the densely connected Bidirectional LSTM with max-pooling of CNN Network can get deeper semantic features and more accurate results by using a densely connected structure with the number increases, experiments were conducted on the 20News-20 dataset and 20News-4 dataset.

The results is shown in Fig. 5. For 20News-20 dataset and 20News-4 dataset, we found that increasing the number of layers causes the accuracy to increase. The most accurate result on the 20News-20 dataset occurs when the layer index equals 5, which is 0.38% points higher than when it is 3. The most accurate result on the 20News-4 dataset occurs when the layer index equals 4 and 7.

By increasing the number of layers, the model obtains more and more semantics by passing semantic features to later layers with a densely connected structure, and the accuracy normally increased when the layer index is less than 7. There is indeed redundant amounts of input in each layers, which may result in the decrease of accuracy when the layer index is much bigger. We plan to address this bottleneck in the future revision. However, the accuracy rate of the

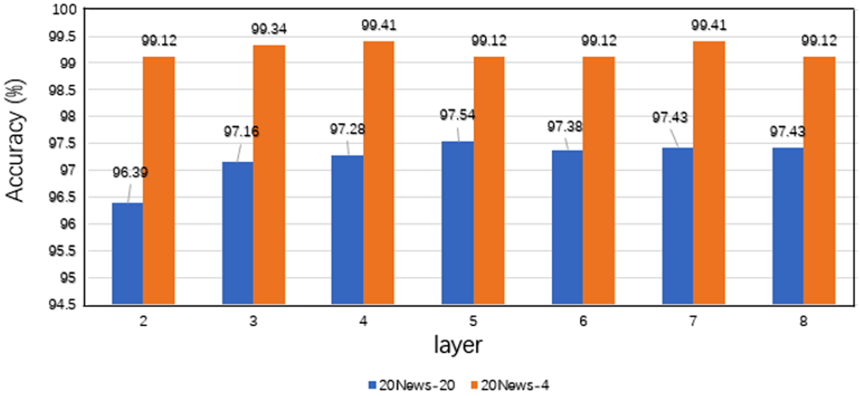


Fig. 5. The accuracy rate of Dense-BiLSTM-MP on the 20News-20 dataset and the 20News-4 dataset as the number of layers increases

Dense-BiLSTM-MP model is rather higher than state of the art methods when the layer index is not very big.

To find the Dense-BiLSTM-MP model's performance on the shortest text and longest text of datasets used in this paper, we give comparison on the AG dataset and the 20News-4 dataset, which are the shortest dataset and longest dataset that this paper used, respectively.

Table 5. Accuracy for the 20News-4 dataset and the AG dataset with different layer.

Layer	AG	20News-4
2	62.09	99.12
3	65.75	99.34
4	67.77	**99.41**
5	67.84	99.12
6	**68.47**	99.12
7	66.79	**99.41**
8	67.43	99.12

The results is shown in Table 5, For the AG dataset, the accuracy increased when the layer index is less than 7, and the highest accuracy occurred in the layer index 6. For the 20Newsgroup dataset, the highest accuracy occured in the layer index 4. The results prove that for short texts which are usually lack of context semantics, the Dense-BiLSTM-MP model is good at mining deeper semantics by using a dense structure. And for long texts, the Dense-BiLSTM-MP model can achieve deeper semantics rapidly without too many layers. Hence, for short text datasets, layer index 6 with the highest accuracy is recommended, and for the dataset with long text, 4 layer is recommended.

5 Conclusion

Facing the bottleneck of the semantics extraction problem, to get deeper context semantics, this paper proposed a novel model, Dense-BiLSTM-MP. The contributions of this paper mainly include: (1) A densely connected BiLSTM structure was proposed to pass each layer's semantic features to later layers, which greatly enhance the context semantics to achieve high accuracy rate. When passing semantic features to later layers, we break down the context features of each word, which is different from other dense structures. This paper was first to prove that breaking down the context features of each word in the densely connected structures help improve the accuracy rate. A max-pooling layer of a CNN was selected for feature selection to extract key semantic features of each layers' output, which was firstly used in dense structures for text classification. (2) The Dense-BiLSTM-MP model is more accurate than most baseline methods on five text classification tasks. Take AG dataset as an example, the accuracy of Dense-BiLSTM-MP with layer index of 3 is about 2% higher than the best performance of baseline methods, especially when Dense-BiLSTM-MP model's layer index is 6, the accuracy is about 5% higher than the best performance of baseline methods. (3) The Dense-BiLSTM-MP model gets high accuracy rate without too much time consumption when the layer index is not very big.

Acknowledgments. This work was supported by foundation project of 17XYY012 and 61772075. This work was also supported by the key special project of the prevention and control of public security risk and emergency technical equipment (2018YFC0832304).

References

1. Bakus, J., Kamel, M.S.: Higher order feature selection for text classification. Knowl. Inform. Syst. **9**(4), 468–491 (2006)
2. Cao, Y., Xu, R., Chen, T.: Combining convolutional neural network and support vector machine for sentiment classification. In: Chinese National Conference on Social Media Processing (2015)
3. Cataltepe, Z., Aygun, E.: An improvement of centroid-based classification algorithm for text classification. In: 2007 IEEE 23rd International Conference on Data Engineering Workshop (2007)
4. Ding, Z., Xia, R., Yu, J., Li, X., Yang, J.: Densely connected bidirectional LSTM with applications to sentence classification. In: Zhang, M., Ng, V., Zhao, D., Li, S., Zan, H. (eds.) NLPCC 2018. LNCS (LNAI), vol. 11109, pp. 278–287. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99501-4_24
5. Du, J., Gui, L., Xu, R., He, Y.: A convolutional attention model for text classification. In: Huang, X., Jiang, J., Zhao, D., Feng, Y., Hong, Yu. (eds.) NLPCC 2017. LNCS (LNAI), vol. 10619, pp. 183–195. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73618-1_16
6. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. **18**(5–6), 602–610 (2005)

7. He, K., Zhang, X., Ren, S., Jian, S.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
9. Huang, G., Liu, Z., Maaten, L.V.D., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017)
10. Huynh, T., He, Y., Willis, A., Rüger, S.: Adverse drug reaction classification with deep neural networks. In: COLING (2016)
11. Jiang, X., Zhang, B., Ye, Y., Liu, Z.: A hierarchical model with recurrent convolutional neural networks for sequential sentence classification. In: Tang, J., Kan, M.-Y., Zhao, D., Li, S., Zan, H. (eds.) NLPCC 2019. LNCS (LNAI), vol. 11839, pp. 78–89. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32236-6_7
12. Kim, S., Kang, I., Kwak, N.: Semantic sentence matching with densely-connected recurrent and co-attentive information. Proc. AAAI Conf. Artif. Intell. **33**, 6586–6593 (2019)
13. Kim, Y.: Convolutional neural networks for sentence classification. Eprint Arxiv (2014)
14. Koutnik, J., Greff, K., Gomez, F., Schmidhuber, J.: A clockwork RNN (2014). arXiv preprint: arXiv:1402.3511
15. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Proceedings of Conference of the Association for the Advancement of Artificial Intelligence AAAI (2015)
16. Lee, C., Kim, Y.B., Lee, D., Lim, H.: Character-level feature extraction with densely connected networks (2018). arXiv preprint: arXiv:1806.09089
17. Lee, J.Y., Dernoncourt, F.: Sequential short-text classification with recurrent and convolutional neural networks (2016). arXiv preprint: arXiv:1603.03827
18. Li, B.: Importance weighted feature selection strategy for text classification (2017)
19. Mitchell, J., Lapata, M.: Composition in distributional models of semantics. Cogn. Sci. **34**(8), 1388–1429 (2010)
20. Smola, A.J., Scholkopf, B.: A tutorial on support vector regression. Stat. Comput. **14**, 199–222 (2004)
21. Song, X., Petrak, J., Roberts, A.: A deep neural network sentence level classification method with context information (2018). arXiv preprint: arXiv:1809.00934
22. Srivastava, R.K., Greff, K., Schmidhuber, J.: Training very deep networks. Computer Science (2015)
23. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
24. Tas, O., Kiyani, F.: A survey automatic text summarization. PressAcademia Procedia **5**(1), 205–213 (2007)
25. Wang, S., Huang, M., Deng, Z.: Densely connected CNN with multi-scale feature attention for text classification. In: IJCAI, pp. 4468–4474 (2018)
26. Xu, C., Huang, W., Wang, H., Wang, G., Liu, T.Y.: Modeling local dependence in natural language with multi-channel recurrent neural networks. Proc. AAAI Conf. Artif. Intell. **33**, 5525–5532 (2019)
27. Xu, X., Zhang, B., Zhong, Q.: Text categorization using SVMs with Rocchio ensemble for internet information classification. In: Lu, X., Zhao, W. (eds.) ICCNMC 2005. LNCS, vol. 3619, pp. 1022–1031. Springer, Heidelberg (2005). https://doi.org/10.1007/11534310_107

28. Yoo, Y.H., Han, K., Cho, S., Koh, K.C., Kim, J.H.: Dense recurrent neural network with attention gate (2018)
29. Zhang, J., Lertvittayakumjorn, P., Guo, Y.: Integrating semantic knowledge to tackle zero-shot text classification. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long Papers), pp. 1031–1040. Association for Computational Linguistics, Minneapolis, USA (Jun 2019)
30. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in Neural Information Processing Systems, pp. 649–657 (2015)
31. Zhang, Y., Chen, G., Yu, D., Yaco, K., Khudanpur, S., Glass, J.: Highway long short-term memory RNNs for distant speech recognition. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5755–5759. IEEE (2016)
32. Zhang, Z., et al.: Inductive structure consistent hashing via flexible semantic calibration. In: IEEE Transactions on Neural Networks and Learning Systems (2020)
33. Zhao, Y., Shen, Y., Yao, J.: Recurrent neural network for text classification with hierarchical multi-scale dense connections. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 5450–5456. AAAI Press (2019)

A Context-Aware Computing Method of Sentence Similarity Based on Frame Semantics

Wenjing Liu[1], Tiexin Wang[1,2](✉), Zhibin Yang[1,2], and Jingwen Cao[1]

[1] College of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics, 29#, Jiangjun Road, Jiangning District,
Nanjing 211106, China
{liuwenjing,tiexin.wang,cjw1028}@nuaa.edu.cn,
yangzhibin168@163.com

[2] Key Laboratory of Safety-Critical Software, (Nanjing
University of Aeronautics and Astronautics), Ministry of Industry and Information Technology,
Nanjing, China

Abstract. Sentence similarity computing is a typical technology used in natural language processing, which aims at finding valuable information from documents. By adapting advanced technologies, such as machine learning and deep learning, current sentence similarity computing methods mainly deal with key words and structures of sentences. The main drawback of current methods is taking no consideration of the influence of sentences context. In this paper, we propose a frame semantics theory based computing method that is built upon FrameNet. By quantitatively analyzing the semantic relations among frames, sentences similarity can be calculated based on the frames that are evoked by the sentences. We also carry out experiments to evaluate the performance of this method with the help of a prototype software tool we developed.

Keywords: Sentence similarity · Semantic analysis · FrameNet · Frame semantics

1 Introduction

In real world, most of the valuable information is stored in the text form and consists of a large number of documents from various data sources, such as news papers, research papers, books, digital libraries, e-mails and Web pages [1]. To obtain high quality information from text, text mining (also called text data mining) is always adopted. Typical text mining tasks include text classification, text clustering, concept/entity extraction, sentiment analysis, document summary, information filtering, social network summarization [2], etc.

Natural language processing (NLP) technologies have been employed widely in text mining process. As one of the typical technologies of NLP, sentence similarity computing is used as a criterion for finding unseen knowledge from a text database [3].

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 114–126, 2020.
https://doi.org/10.1007/978-3-030-65390-3_9

Moreover, sentence similarity computing methods have been used in diverse fields, such as information retrieval, machine translation, automatic question and answer systems [4].

There exist several typical sentence similarity computing methods, such as corpus-based, knowledge-based and structure-based computing methods [5]. However, the above three kinds of methods have their own shortcomings. The corpus-based sentence similarity computing methods depends on artificial intelligence technologies, such as machine learning algorithm and deep learning algorithm. Similar to other methods using artificial intelligence algorithms, this kind of methods also lack the interpretability of the calculation results. The efficiency of knowledge-based computation methods depends on the extraction, analysis and comparison of keywords of sentences. These methods miss the understanding of the overall semantic of sentences. The work of structure-based computing methods is not mature. Considering the above problems, this paper proposes an English sentence similarity calculation method based on frame semantics “FS3C”.

The main contribution of this paper reflects in two aspects. First, we propose a frame semantics theory based sentence similarity computing method “FS3C”, which belongs to the knowledge-based classification. Comparing with other knowledge-based computing methods that are mainly built upon WordNet [6], FS3C takes the influence of the sentence context into consideration (not only focuses on key words). Second, we build a visual environment of FrameNet “Graphical Interpretation for FrameNet: GIFN”, to enhance the interpretability and persuasiveness of FS3C.

The structure of this paper is as follows. In Sect. 2, we present relevant technologies and tools adopted in FS3C. A general overview of FS3C is given in Sect. 3. Section 4 evaluates the performance of FS3C with experiments. Section 5 shows the related work. Section 6 draws the conclusion and outlines the future work.

2 Relevant Technologies and Tools

This section presents the key technologies and tools adopted in FS3C. Two subsections are divided and introduce briefly “FrameNet” and “Semafor”, respectively.

2.1 FrameNet

FrameNet[1] is a project developed and maintained by the “International Computer Science Institute” from Berkeley. It is based on the “Frame Semantics” theory [7]. It alleges that without all the basic knowledge related to the word, one cannot understand the meaning of a single word [8].

FS3C is built up on the basis of FrameNet, and three main elements “Frame”, “Frame Elements (FEs)”, and “Lexical Units (LUs)” from FrameNet are used in FS3C.

As shown in Fig. 1, a frame contains a set of FEs and can be evoked by a set of LUs. The relation between the “Intentionally_act” frame (contains the FEs “Agent” and “Act”, and can be evoked by LU “act.v”) and the frame “Intentionally_affect” is “inheritance”, while the relation between the frame “Intentionally_affect” and the frame “Import_export_scenario” is “use”.

[1] http://framenet.icsi.berkeley.edu/.

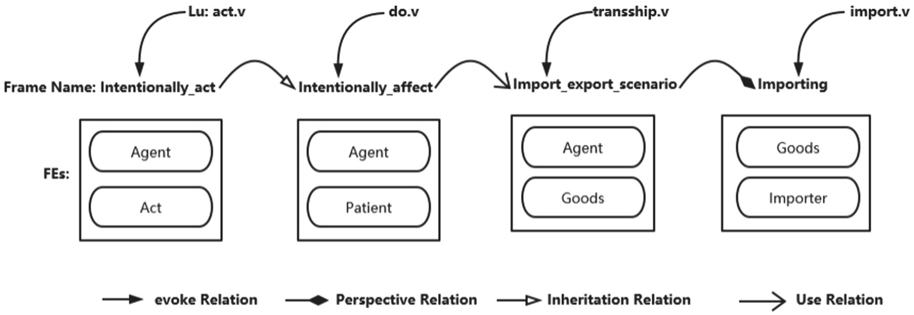


Fig. 1. An illustration of the relations between frames, FEs and LUs

To better use FrameNet, we develop a visual environment "Graphical Interpretation for FrameNet: GIFN". The three kinds of elements "frame", "FEs", "LUs" and relations among frames in FrameNet are extracted and mapped into the graph database neo4j. GIFN is detailed in Subsect. 3.2.

2.2 Semafor

Semafor[2] is an open source software tool developed as a frame-semantic parser [9]. It can automatically process English sentences according to FrameNet.

FS3C combines Semafor to analyze automatically the frames evoked by sentences. We use the following example to illustrate the working mechanism of Semafor. The English sentence "Do you want me to hold off until I finish July and August?" is taken as input to Semafor, and Fig. 2 (taken from the official website) shows the parsing result of this sentence.

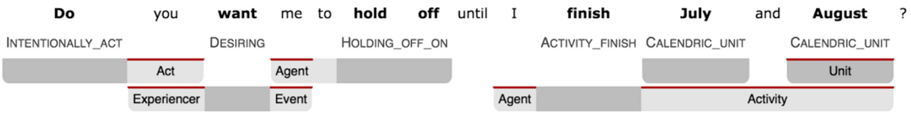


Fig. 2. An illustrating example of the working mechanism of Semafor

In Fig. 2, "Do" evokes the frame "Intentionally_act", which contains the FE "Act". The content of "Act" is "you". "want" evokes the frame "Desiring". The FEs "Experiencer" and "Event" are contained in the frame "Desiring", the content of the two FEs are "you" and "me", respectively.

In FS3C, we use the analysis result of Semafor to compute sentence similarity. The detailed process is shown in Subsect. 3.3.

[2] http://www.cs.cmu.edu/~ark/SEMAFOR/.

3 FS3C Overview

This section contains three subsections. The first subsection introduces the integrated workflow of FS3C. The second subsection presents “GIFN”. The algorithms of computing sentences similarity are illustrated in the third subsection.

3.1 The Workflow of FS3C

As shown in Fig. 3, the input of FS3C is two comparing sentences and the output is the comparing result standing by a value ranging from 0 to 1. The higher of this value means the higher similarity between the two sentences.

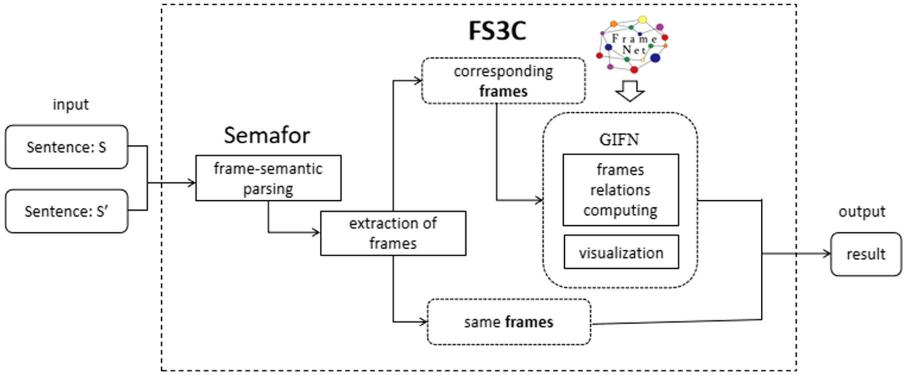


Fig. 3. The integrated working flow of FS3C

First, the input sentences are pre-processed by Semafor. Then, we extract all the frames and form them into two groups, respectively. Next, we visualize the relations (same or corresponding) between the two groups of frames with the help of GIFN. Based on the relations found among frames, the similarity (comparing result) of the two sentences is obtained. Finally, the comparing result is used to determine whether two comparing sentences are similar or not.

3.2 GIFN

In order to enhance the interpretability and persuasiveness of FS3C, we develop GIFN. All the frames, FEs and LUs are stored as nodes, while relations are stored as edges in neo4j. In GIFN, there are 1,019 frame nodes, 11,829 LU nodes, 8,995 FE nodes and 1,507 relations among frames.

Figure 4 shows the relations between the frame “Intentionally_act” and FEs, LUs and the frame “Intentionally_affect”. There are two frame nodes (shown as the biggest circle) “Intentionally_act” and “Intentionally_affect”, linked by an inheritance relation. The nodes shown as the smallest circle represent LUs (totally 16) that can evoke “Intentionally_act”. The nodes shown as the middle-size circle (totally 14) represent the FEs contained in frame “Intentionally_act”.

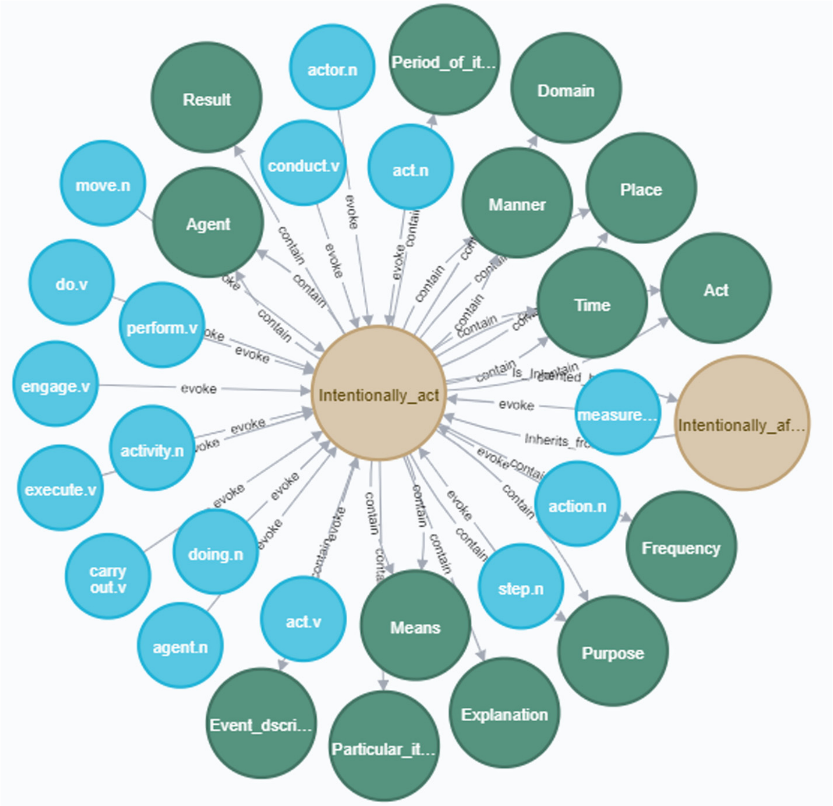


Fig. 4. The relations between the frame "Intentionally_act" and other nodes in GIFN

In FS3C, the semantic relations among frames are used to determine the similarity between sentences. To make good use of these relations, a corresponding value, which ranges from 0 to 1, is assigned to each of them. The mechanism of value assigning is inspired by the work [10]. A higher corresponding value means more connected of two frames.

To obtain reasonable values of relations among frames, we design a two-step corresponding values assigning and adjusting process.

In the first step, we invited 5 researchers to assign values to the relations among frames based on their experience independently. Then, after a round-table discussion, they gave a final set of values to the thirteen kinds frame relations (shown in the first and second column of Table 1). Next, we randomly selected 60 pairs of sentences (15 pairs of them are labeled as not similar while the others are labeled as similar) from the data set MSRP (Microsoft Research Paraphrase Corpus) to test the rationality of the assignment. More information about MSRP is given in Subsect. 4.2. We compared the values of F1-measure corresponding to different thresholds and assigned the threshold corresponding to the maximum F1-measure value. Finally, according to the testing result, we got a threshold "0.598" (if a computing result between two comparing sentences is higher than it, we regard the two sentences are similar). With this threshold, the F1-measure is "0.7957".

Table 1. The assignment of corresponding values to semantic relations in FrameNet

Frame relation	First assigning Corresponding semantic value	Adjusted corresponding values
Inherits from	0.6	0.55
Is Inherited by	0.6	0.55
Perspective on	0.5	0.45
Is Perspectivized in	0.5	0.45
Uses	0.3	0.3
Is Used by	0.3	0.3
Subframe of	0.4	0.35
Has Subframe(s)	0.4	0.35
Precedes	0.2	0.2
Is Preceded by	0.2	0.2
Is Inchoative of	0.3	0.3
Is Causative of	0.3	0.3
See also	0.4	0.4

In the second step, the five researchers adjusted the corresponding values according to the value of F1-measure. The final assigned corresponding values are listed in the third column of Table 1. Furthermore, based on the new assigned values, we adjusted the threshold as "0.5367" (with the F1-measure "0.81").

3.3 Similarity Computing Concerning on Frames

By employing Semafor, we can extract all the frames that are evoked by sentences. Equation (1) is defined concerning on frame semantics and is used to compute the value of similarity between two sentences. Before illustrating Eq. (1), we give some definitions as follows.

Definition: $Frame\_S = \{\mathrm{frame_i} \mid \mathrm{i} \in [1, |\mathrm{Frame\_S}|]\}$

$Frame\_S' = \{\mathrm{frame_j} \mid \mathrm{j} \in [1, |\mathrm{Frame\_S'}|]\}$

$Frame\_same = Frame\_S \cap Frame\_S' = \{\mathrm{frame_k} \mid \mathrm{frame_k} \in Frame\_S, \mathrm{frame_k} \in Frame\_S'\}$

$Frame\_rel = \{<\mathrm{frame_i}, \mathrm{frame_j}> \mid \mathrm{frame_i} \in Frame\_S, \mathrm{frame_j} \in Frame\_S'$, and there exists a path between $\mathrm{frame_i}$ and $\mathrm{frame_j}$ in GIFN$\}$

Frame_S, *Frame_S′*, *Frame_same* and *Frame_rel* are sets of frames (or frame pairs). *Frame_S* and *Frame_S′* contain frames that are evoked by two comparing sentences (i.e., *S* and *S’*), respectively. *Frame_same* contains frames belonging both to *Frame_S* and $Frame\_S'$. While *Frame_rel* contains all corresponding frame pairs in which one frame from *Frame_S* and the other frame from $Frame\_S'$.

As shown in Eq. (1), Frame_score stands for the similarity between two sentences (i.e., “*S*” and “S'”). It is affected by the percentage of overlap frames (elements of “*Frame_same*”) between two sentences. Also, the influence of Path_Score, which is computed in Eq. (2), is taken into account while calculating Frame_score.

$$\text{Frame\_Score} = \frac{|Frame\_same| + Path\_Score}{Maximum(|Frame\_S|, |Frame\_S'|)} \tag{1}$$

$$\text{Path\_Score} = \sum_{i=1}^{|Frame\_rel|} \text{Path\_Value } i \tag{2}$$

$$\text{Path\_Value} = \frac{\sum_{i=1}^{Countpath} weight}{Countpath} \tag{3}$$

As shown in Eq. (2), Path_Score is determined by the sum of all the Path_Value. which is determined by the shortest path between two frames and calculated by employing Eq. (3). The shortest path can be located with the help of GIFN. “*Count_path*” in Eq. (3) means the number of edges (number of semantic relations between two frames) on the shortest path. “*weight*” means the corresponding values of semantic relations between frames.

The pseudo code of FS3C is as follows.

Algorithm 1. FS3C Algorithm

```
Input: S, S’
Output: Frame_score of S and S’
Frame_S = {$frame_i$} for S, Frame_S’ = {$frame_j$} for S’
foreach item frame $\alpha$ ($frame_i$) $\in$ Frame_S do
    Frame_same = Frame_S $\cap$ Frame_S’
    Frame_S = Frame_S – Frame_same
    Frame_S’ = Frame_S’ – Frame_same
end foreach
foreach item frame $\alpha$ ($frame_i$) $\in$ Frame_S do
    ShortestPath <$\alpha$, $\alpha$’> in GIFN, $\alpha$’($frame_j$) $\in$ Frame_S’
    Frame_S = Frame_S – {$\alpha$}
    Frame_S’ = Frame_S’ – {$\alpha$’}
    Path_value (Equation 3)
    Frame_rel = {<$\alpha$, $\alpha$’>}
end foreach
Path_score (Equation 2)
Frame_score (Equation 1)
return Frame_score
```

Taking the two following sentences *S* and *S’* as input, Table 2 and Table 3 show the processing result.

Table 2. The parsing result of S with Semafor

Frame	LU_text	FE	FE_text
Subordinates_and_superiors	Senior		
Delivery	Delivered	Time Theme Deliverer	On Thursday their summary assessments a senior aide
Assessing	Assessments	Assessor	Their summary assessments
Origin	American	Origin Entity	American about 300
Military	Military	Force	Military
Leadership	Officers		
Calendric_unit	Thursday	Unit	Thursday

Table 3. The parsing result of S' with Semafor

Frame	LU_text	FE	FE_text
Military	Air Force military	Force	Military
Leadership	General	Leader	General
Subordinates_and_superiors	Senior		
Presence	Presented	Entity	A senior aide
Assessing	Assessments	Assessor	Their
Origin	American	Origin Entity	American briefing
People_by_vocation	Officers	Person	Officers
Locale_by_use	Base	Locale	Base
Calendric_unit	Thursday	Unit	Thursday

S: "Moseley and a senior aide delivered their summary assessments to about 300 American and allied military officers on Thursday."

S': "General Moseley and a senior aide presented their assessments at an internal briefing for American and allied military officers at Nellis Air Force Base in Nevada on Thursday."

As shown in Table 2, the first column lists all the frames that are evoked by the LUs (words) in S, while all the corresponding LU_texts are listed in the second column. The difference between LU and LU_text is: a LU_text reflects a string (word) itself while a LU also indicates the word feature (e.g., noun, verb). The third column shows the FEs contained by the frame (e.g., the FE "Leader" to the frame "Leadership"). The fourth column presents the concrete content of the corresponding FE. Table 3 shows the parsing result of S'.

From Table 2 and Table 3, we know that *Frame_Same* contains six frames: "Subordinates_and_superiors", "Assessing", "Origin", "Military", "Leadership" and "Calendric_unit". The "|Frame_same|" in Eq. (1) is 6.

With the help of GIFN, we calculate that the Frame_rel in this example contains only one element "<Delivery, Presence>". We assume that if one frame in Frame_S can be linked with one frame in Frame_S' within 10 edges, the two frames are related. The idea of using the shortest path (within 10 edges) of two frame nodes is inspired by [11]. Then, we get the Path_score, 0.45625.

Finally, Frame_score is calculated as 0.71736. Considering the threshold "0.5367", as discussed in Subsect. 3.2, sentence S and Sentence S' is similar, which is consistent with the human labeled judgment.

4 Experiments

This section contains three subsections. The first subsection shows the evaluation criteria. We briefly introduce the data set used to evaluate FS3C in the second subsection. Finally, we analyze the experiment result in the third subsection.

4.1 Evaluation Criteria

We introduce F1-measure and accuracy to measure the effectiveness of FS3C. The F1-measure has been widely used in natural language processing [12]. F1-measure and accuracy are defined as follows.

$$\mathrm{F1} = \frac{2 \times precision \times recall}{precision + recall} \tag{4}$$

$$\mathrm{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

F1-measure is the harmonic mean of precision and recall. A higher value of F1-measure means a better performance method. Precision and recall are defined as follows.

$$\mathrm{precision} = \frac{TP}{TP + FP} \tag{6}$$

$$\mathrm{recall} = \frac{TP}{TP + FN} \tag{7}$$

The definition of TP, TN, FP and FN are as follows.

- TP: number of sentences predicted (by computing result) to be similar that are actually similar.
- TN: number of sentences predicted to be dissimilar that are actually dissimilar.
- FP: number of sentences predicted to be similar that are actually dissimilar.
- FN: number of sentences predicted to be dissimilar that are actually similar.

4.2 Data Set

"Microsoft Research Paraphrase Corpus: MSRP"[3], which has been widely used in evaluating similarity measure techniques [5], is employed as the data set to verify the performance of FS3C.

MSRP data set is a well-known data set for testing effectiveness of sentence similarity computing methods. It contains around 5,700 pairs of sentences, and each pair of sentences has been manually assigned a value 0 (not similar) or 1 (similar). Totally, 67% of the sentence pairs are labeled to be similar, while the other 33% are labeled to be not similar. The sentence pairs are divided into two sets: a training set (4,076 sentence pairs) and a test set (1,725 pairs).

4.3 Computing Result with FS3C

We use the test set to validate FS3C. In these 1,725 pairs of sentences, 1,147 pairs of sentences are labeled similar and 578 pairs of sentences are labeled not similar.

Figure 5 shows the evaluation result concerning "Accuracy" values and "F1-measure" values of other six sentence similarity computing methods [13] and FS3C. Comparing with other methods, we can see that FS3C owns the highest Accuracy value and second highest F1-measure value.

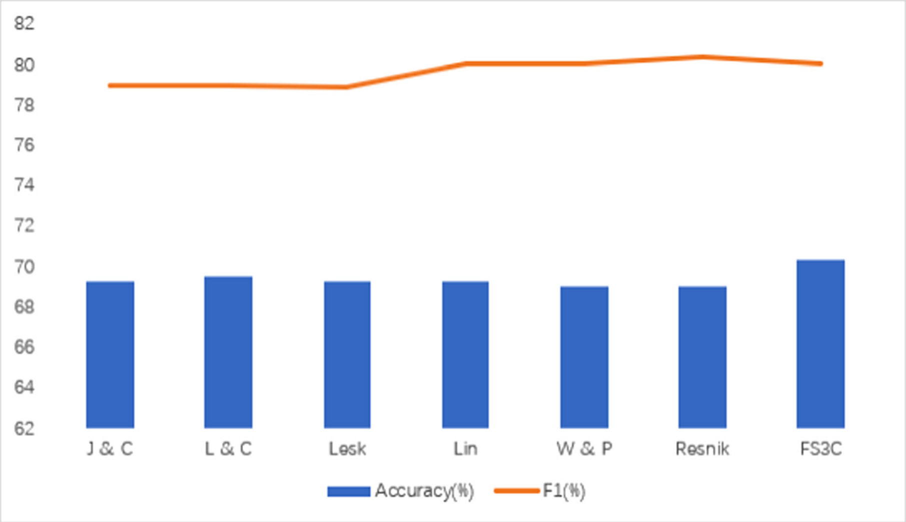


Fig. 5. Accuracy and F1-measure of FS3C and Others against MSRP dataset

Table 4 lists the values of Accuracy, Precision, Recall and F1-measure of the other six computing methods presented in [13] and FS3C. From this table, the value of Accuracy of FS3C, which is based on frame semantics (FrameNet), is better than those six methods

[3] https://www.microsoft.com/en-us/download/details.aspx?id=52398.

built upon WordNet. Although the F1-measure value of FS3C is the second, the Accuracy value and the precision value is much better than Lin, W&P and Resnik. It shows that FS3C provides a new feasible direction for sentence similarity calculation.

Table 4. Performance of FS3C and other knowledge-based computing methods

Method	Accuracy	Precision	Recall	F1-measure
J & C [13]	69.3	72.2	87.1	79.0
L & C [13]	69.5	72.4	87.0	79.0
Lesk [13]	69.3	72.4	86.6	78.9
Lin [13]	69.3	70.2	92.1	80.0
W & P [13]	69.0	70.2	92.1	80.0
Resnik [13]	69.0	69.0	96.4	80.4
FS3C [13]	70.3	72.3	89.6	80.0

Totally, there are 512 pairs of sentences that FS3C makes wrong judgments. To analyze the cause of wrong judgments, we selected and analyzed 20 pairs of sentences (10 pairs of similar sentences that FS3C judged as dissimilar and 10 pairs of dissimilar sentences that FS3C judged as similar) from the 512 pairs of sentences. Two main causes are concluded as follows.

- The content of two sentences is almost the same, except some details of certain places, names or times that Semafor could not realize.

Take the following sentences as an example. S1: “It was a little bit embarrassing the way we played in the first two games,” Thomas said. S2: “We’re in the Stanley Cup finals, and it was a little bit embarrassing the way we played in the first two games.” The only difference is “in the Stanley Cup finals”, which Semafor missed.

- There are few frames that the sentences can evoke. That means some key content of the sentences may not be identified by FS3C.

Take the following sentences as an example. S3: Ballmer has been vocal in the past warning that Linux is a threat to Microsoft. S4: In the memo, Ballmer reiterated the open-source threat to Microsoft. In S3, only two frames: “past” and “threat” are evoked, while in S4, “Relative_time” and “Commitment” are evoked. It can be seen that the same frame accounts for a large proportion of the total number of frames, so FS3C made a wrong judgment.

To improve the performance of FS3C, a potential solution is to compare the FEs contained in the same frames. This is also one of the future research directions of our work.

5 Related Work

Considering the adopted technologies, the computing methods used in sentence similarity computing can be classified into three categories: word-based similarity, structure-based similarity and vector-based similarity [5].

In [14], a model is proposed to consider both the similarities and dissimilarities by decomposing and composing lexical semantics over sentences. Each word is represented as a vector and the cosine similarity is used to construct a words similarity matrix. Moreover, a set of matching functions are proposed to construct semantic matching vector for each word. Another computing method [15] belongs to this category uses Word Sense Disambiguation (WSD) and synonym expansion to provide a richer semantic context.

In [16] and [17], two methods considering the structural information to measure sentence similarity are proposed. Both of the two methods belong to the structure-based similarity category. Focusing on natural language sentences which have no obvious relations or concept overlap, a hybrid approach combining corpus-based ontology and grammatical rules is defined in [16]. Focusing on Short-Text Semantic Similarity (STSS), a model employing a part-of-speech weighting scheme and building on the basis of a statistical bag-of-words approach is proposed in [17].

For the vector-based similarity category, the research work presented in [18] and [19] belong to it. In [18], it firstly uses the pre-trained GloVe word vectors to calculate the average between these vectors. Then, cosine similarity is used to measure the sentence similarity. In [19], their key idea is that similarity in the latent space implies semantic relatedness. They described three ways in which labeled data can improve the accuracy of these approaches on paraphrase classification.

Comparing with the above computing methods, the main novelty of FS3C is: it is built on frame semantics theory and taking sentence context into account. The frames that are evoked by sentences are analyzed and compared instead of analyzing the key words contained in the sentences. The results of the experiments presented in the fourth section prove that FS3C is a feasible and competitive solution.

6 Conclusion

We propose a new sentence similarity computing method "FS3C", which is built on frame semantics theory. First, we developed a graph knowledge base "GIFN" of frames by adopting the content of FrameNet. In GIFN, we assigned values to different semantic relations among frames through a systematic process. Then, we designed a theoretical framework of sentence similarity computing based on frame semantics. Next, we proposed a computing process combining the mature tool Semafor. We also developed a prototype tool to realize the theoretical framework. Finally, to evaluate the efficiency of FS3C, we used the prototype tool to test with MSRP data set and analyzed the results.

The idea of employing frames to determine sentence similarity is new and the result of experiments proves the feasibility of this idea. The future research work can be carried out in two directions: i) taking more content of FrameNet (e.g., FEs) as the similarity comparing elements, and ii) focusing on domain specific sentences that have relatively fixed structures and context, such as bug reports.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (61872182), the Open Fund of the Ministry Key Laboratory for Safety-Critical Software Development and Verification (XCA1816401 and XCA19016-06).

References

1. Allahyari, M., et al.: A brief survey of text mining: classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919 (2017)
2. Li, J., Liu, C., Yu, J.X., Chen, Y., Sellis, T., Culpepper, J.S.: Personalized influential topic search via social network summarization. In: IEEE International Conference on Data Engineering. IEEE Computer Society (2017)
3. Atkinson-Abutridy, J., Mellish, C., Aitken, S.: Combining information extraction with genetic algorithms for text mining. IEEE Intell. Syst. **19**(3), 22–30 (2004)
4. Pawar, A., Mago, V.: Calculating the similarity between words and sentences using a lexical database and corpus statistics. arXiv preprint arXiv:1802.05667 (2018)
5. Farouk, M.: Measuring Sentences Similarity: A Survey. arXiv preprint arXiv:1910.03940 (2019)
6. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
7. Ruppenhofer, J., Ellsworth, M., Schwarzer-Petruck, M., Johnson, C.R., Scheffczyk, J.: FrameNet II: Extended Theory and Practice (2006)
8. Fillmore, C.J., Baker, C.F.: Frame semantics for text understanding. In: Proceedings of WordNet and Other Lexical Resources Workshop, NAACL (vol. 6) (June 2001)
9. Das, D., Schneider, N., Chen, D., Smith, N.A.: Probabilistic frame-semantic parsing. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 948–956. Association for Computational Linguistics (June 2010)
10. Wang, T., Truptil, S., Benaben, F.: An automatic model-to-model mapping and transformation methodology to serve model-based systems engineering. IseB **15**(2), 323–376 (2016). https://doi.org/10.1007/s10257-016-0321-z
11. Li, J., Liu, C., Islam, M.: Keyword-based correlated network computation over large social media. In: IEEE International Conference on Data Engineering IEEE (2014)
12. Derczynski, L.: Complementarity, F-score, and NLP Evaluation. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), pp. 261–266 (May 2016)
13. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: AAAI, vol. 6, No. 2006, pp. 775–780 (July 2006)
14. Wang, Z., Mi, H., Ittycheriah, A.: Sentence similarity learning by lexical decomposition and composition. arXiv preprint arXiv:1602.07019 (2016)
15. Abdalgader, K., Skabar, A.: Short-text similarity measurement using word sense disambiguation and synonym expansion. In: Li, J. (ed.) AI 2010. LNCS (LNAI), vol. 6464, pp. 435–444. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17432-2_44
16. Lee, M.C., Chang, J.W., Hsieh, T.C.: A grammar-based semantic similarity algorithm for natural language sentences. Sci. World J. **2014** (2014)
17. Batanović, V., Bojić, D.: Using part-of-speech tags as deep-syntax indicators in determining short-text semantic similarity. Comput. Sci. Inf. Syst. **12**(1), 1–31 (2015)
18. Putra, J.W.G., Tokunaga, T.: Evaluating text coherence based on semantic similarity graph. In: Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing, pp. 76–85 (August 2017)
19. Ji, Y., Jacob, E.: Discriminative improvements to distributional sentence similarity. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (2013)

Learning the Concept Embeddings of Ontology

Jiangtao Qiu(✉) and Siyu Wang

School of Information, Southwestern University of Finance and Economics, Chengdu, China
Qiujt_t@swufe.edu.cn, siyu_wang@smail.swufe.edu.cn

Abstract. The semantic similarities among concepts play an important role in many tasks. Ontology represents the semantic relationship among concepts. Traditional methods use the path-length between concepts in the ontology to calculate their semantic similarity. However, this simple method cannot present semantic relationship among concepts well. This study seeks to learn the concept embeddings in ontology, and then use the cosine similarity of two embeddings to inform their sematic similarity. We developed a framework, called concept2vec, to perform the task. The experimental results demonstrate that our work is effective on learning representation of concepts in ontology.

Keywords: Ontology · Concept embedding · Biased random walk · WordNet

1 Introduction

The semantic similarity between concepts play an important role on many tasks including information retrieval, text mining and knowledge base, etc. Traditional approaches to modeling semantic similarity compute the semantic distance of the concepts within an ontology (e.g. WordNet). Many studies use the path-length as the semantic distance. However, the semantic distance of any two pair of nodes in the ontology may have a large difference depending on their positions in the ontology even if they have the same path-length. For example, two pairs of concepts <*soccer*, *rugby*> and <*game*, *domesticity*> in WordNet all have a path length of 2 shown in Fig. 1. Nevertheless, <*soccer*, *rugby*> have an obviously shorter semantic distance than <*game*, *domesticity*>. Therefore, the path-length cannot present the semantic relationships among concepts well. The concepts in ontology have specific semantic relationships that need to be addressed with specific techniques.

To deal with above problem, this study seeks to learn a distributed representation for every concept in ontology, called concept embedding. Representation learning is a set of techniques that learn a distributed representation of data, so that they can support effective machine learning. Representation learning of knowledge graph is similar with our task. Although ontology can be regarded a part of knowledge graph, however, the research efforts on the representation learning of knowledge graph do not address the problem of semantic similarity. For example, translation-based graph embedding methods [1] address the hierarchical relationships in Knowledge Base, but they cannot calculate the semantic similarity for any two concepts that do not have a direct link.

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 127–134, 2020.
https://doi.org/10.1007/978-3-030-65390-3_10

Inspired by *Golve* [2], we develop a framework, called concept2vec, to learn the concept embeddings in WordNet for the purpose of calculating semantic similarity of concepts.

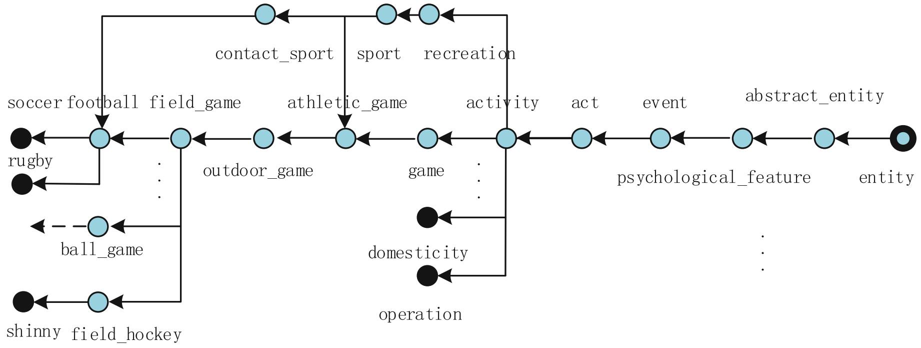


Fig. 1. A part of WordNet

2 Related Works

Starting from the day when Hinton introduced distributed representations for symbolic data in 1986 [6], many efforts on representation learning of word, including *word2vec* [4] and *Golve* [2], have significantly promoted the development of the field.

Ontology is generally considered as a part of knowledge base (KB). Thus, the studies on representation learning of KB also can generated the concept embeddings of ontology. Translation-based models, such as TransE and its extensions [3, 8], can represents both entities and relations in KB as vectors in the same space. However, they fail to calculate the semantic relationship for two concepts that do not have a direct link in ontology.

Preserving network structure is a fundamental requirement for network embedding. Inspired by *word2vec*, many network embeddings algorithms employ random walk models to generate random paths over a network. The nodes position in the generated paths imply the network structure. Considering a node a word, a random path can be regarded as a sentence, and the node neighborhood can be identified by co-occurrence rate as in *word2vec*. Some representative methods of network embeddings include DeepWalk [7] and node2vec [5]. If we see ontology as a tree, the representation learning of network can by employed to generate concept embeddings. However, they do not address our concerns for the sematic relationships among concepts discussed in Sect. 1.

3 Learning Representation of Concepts

3.1 Motivation

Many methods on network embedding, including node2vec [5] and deepWalk [7], are inspired by the word embeddings. They usually contain two steps: (1) generate a nodes sequence; and (2) employ a word embedding tool, such as *word2vec* or *Golve*, to generate the distributed representation of nodes. This study also attempts to employ the two steps to learn representation of concepts in ontology. In practice, however, we find the two steps have their own problems.

First, the concepts in ontology have the specific semantic relationships that need to be addressed. We thus have to develop a specific strategy of random walk for the purpose.

Second, this work is inspired by *Golve*, but *Golve* cannot deal with our task well. We first explore *Golve*. Let X denote the matrix of word-word co-occurrence counts and X_{ij} inform the times of word j occurring in the context of word i. Every word has two roles, including itself and the context of other words. The model thus generates two sets of word vectors, W and $\tilde{\mathrm{W}}$, which are equivalent when X is symmetric. Let $\mathrm{w} \in \mathrm{R}^d$ and $\tilde{\mathrm{w}} \in \mathrm{R}^d$ be word vectors and separate context word vectors, respectively. Let b_i and $\widetilde{b_j}$ denote bias of word w_i and $\widetilde{w_j}$. The objective function of *Golve* takes the form

$$\mathrm{J} = \sum_{i,j \in V} f\left(X_{i,j}\right)\left(w_i^T \widetilde{w_j} + b_i + \widetilde{b_j} - \log\left(X_{ij}\right)\right)^2 \tag{1}$$

where

$$f(\mathrm{x}) = \begin{cases} (x/x_{max})^{\alpha} \ \textit{if} \ x < x_{max} \\ 1 \ \textit{otherwise} \end{cases} \tag{2}$$

If we employ *Golve* to learn the concept embeddings, $f\left(X_{i,j}\right)$ refers to a quantity that is tightly related with the co-occurrence frequency of two concepts. Observing Eq. (1), we can find that if both concepts i and j do not have the co-occurrence, i.e., $f\left(X_{i,j}\right) = 0$, no matter what w_i and $\widetilde{w_j}$ are, they cannot affect objective function J. This observation means that *Golve* fails to separate two concepts that have a great semantic distance.

Both *Golve* and *word2vec* address learning the second-order relationship. That is, if two pair of nodes <A, B> and <B, C> have a large co-occurrence, respectively, A and C will have a strong semantic relationship. However, this paper addresses learning the first-order relationship. That is, if two concepts frequently co-occur in a context window, they should have a strong sematic relationship, and a weak semantic relationship otherwise.

We develop a framework, called concept2vec, to meet the above challenges. It contains two steps: (1) starting a biased random walk in an ontology for building a sequence concepts, and (2) learning representation of concepts from the sequence.

3.2 A Biased Random Walk in Ontology

Ontology exhibits a tree structure with certain properties: (1) concepts are corresponding to nodes in the tree, (2) a *subclass-of* relation between two nodes is represented by an edge, and (3) ontology contains a root node. A part of ontology of WordNet is exhibited in Fig. 1 where the black circle stands for leaves and ‘Entity’ is the root of WordNet. We can find that WordNet actually presents a DAG structure. In order to conveniently discuss our work, this paper deals with WordNet as a special tree structure where a node may have multiple parent nodes. We thus design special strategy of random walk to deal with the specific structure. We give a definition.

Definition 1. (Depth of node) The depth of a node in an ontology is the path-length from the node to root.

We first address the specific semantic relationship among concepts on ontology. They are summarized in Table 1.

Table 1. The sematic relationships between concepts

Semantic relationship	Example
The short path between two nodes in ontology does not means that they definitively have a high semantic similarity. It heavily relies on what levels nodes lie on. Two of sibling nodes have a shorter semantic distance when they are closer to the bottom	Two pair of nodes <*soccer*, *rugby*> and <*game*, *domesticity*> all have a path length of 2, but <*soccer*, *rugby*> have a shorter semantic distance than <*game*, *domesticity*> when <*soccer*, *rugby*> is closer to bottom of the ontology

The distributional hypothesis states that words in a similar context tend to have similar meanings. Inspired by the distributional hypothesis, we seek to employ random walk in ontology to generate a sequence of concepts. If two concepts show in a context window with a high frequency, we argue that two concepts have a high semantic similarity. To address the semantic relationships in Table 1, we propose a depth-first search based biased random walk algorithm. The strategy of the biased random walk is listed as follow.

(1) *Having started a random walk on the tree, we need to make a decision in every node for the next step. If all descendants have been visited, we backtrack to parent with a probability b; Otherwise, visit children.*
(2) *b is adapted depending on the levels of the node in the tree. When the node is closer to leaves, b is smaller. Otherwise, b is higher.*

Let A and B be two neighbor nodes. If both nodes have a large depth, they will be more frequently visited because a small probability of backtracking to parent. Accordingly, they will have a high co-occurrence in a context window.

Before starting the random walk, we need to compute the backtracking probability for every node. The steps are listed as follows

(1) *Set the backtracking probability of each node to zero.*
(2) *Find the longest path from leaf to the root, L_{max}.*
(3) *Let d be the decay rate. For all leaves, their backtracking probability are $d^{L_{max}}$*
(4) *Explore a path from a leaf to root. Let n denote a node and n-1 denote prior node of n within the path. We calculate the backtracking probability for nodes along the path. For the node n, its backtracking probability is*

$$b_n = \begin{cases} b_{n-1}/d \ \textit{if} \ \frac{b_{n-1}}{d} > b_n \\ b_n \quad \textit{otherwise} \end{cases}$$

(5) *repeat step (4) until the paths from every leaf to root are explored.*

A biased random walk algorithm in an ontology is formally described in Algorithm 1.

Algorithm 1: A Biased Random Walk in Ontology

```
Inputs : an ontology, decay rate $d$ in range [0,1], a vector of backtracking probability $b$
Output : a sequence of nodes in the ontology
Steps :
Set node $n \leftarrow root$
$s \leftarrow \{ \}$
$v \leftarrow 0$
LOOP:
    IF $n$==$root$ and all nodes has been visited
        break;
    $v$[n] ← $v$[n] + 1; Append n to $s$
    IF $n$ is a leaf THEN
        $n \leftarrow$ SelectNode(parent($n$), $v$)
    ELSEIF the descendant of $n$ has been visited AND a selection lies in $[0, b[n]]$ THEN
        $n \leftarrow$ SelectNode(parent($n$), $v$)
    ELSE
        $n \leftarrow$ SelectNode(children($n$), $v$)
    ENDIF
END
RETURN s
```

The vector v saves the visited times of nodes of ontology in the random walk. Function SelectNode(l, v) selects a node $n \in l$ from the list l in a probability p_n calculated through softmax function.

$$\mathrm{p}_n = \frac{\exp\left(v(n)^{-1}\right)}{\sum_{m \in l} \exp\left(v(m)^{-1}\right)} \tag{3}$$

where $v(n)$ indicates the visited times of node n. A node in WordNet may have multiple parent nodes. The algorithm also needs selecting one of parent nodes according to the visited times of nodes when the walk backtracks to a parent node.

3.3 Learning Concepts Embeddings

Let N denote a set of nodes in the ontology. $f(X_{ij})$ in Eq. (2) informs the semantic relationships among concepts i and j, which is in a range of [0, 1]. To learn the concept embeddings from the semantic relationships and overcome the limitations discussed in Sect. 3.1, this paper develops a model whose objective function takes the form of

$$\mathrm{J} = \sum_{i,j \in N} f(X_{ij})\left(1 - w_i w_j^T\right)^2 + \left(1 - f(X_{ij})\right)\left(w_i w_j^T - S_{ij}\right)^2 \tag{4}$$

We use inner product of two vectors to measure their similarity. A large $f(X_{ij})$ informs that two embeddings w_i and w_j should have a large similarity. $f(X_{ij})\left(1 - w_i w_j^T\right)^2$ addresses the similarity. On the other hand, when two concepts have a much weak semantic relationship (a small $f(X_{ij})$), their embeddings should has a great distance in the embeddings space. We thus add the path-length among concepts in the ontology as constraints. To make the path-length fit the semantic similarity well in the objective function, we map a path-length to a value in a range of [0, 1] use a radial basis function in Eq. (5).

$$S_{ij} = \exp\left(-\gamma \times len(i,j)^{\rho}\right) \tag{5}$$

where $len(i,j)$ refers to path-length of two concepts i and j in the ontology. When set two parameters $\gamma = 0.2$ and $\rho = 1.3$.

When two concepts have a small semantic relationship, $\left(1 - f(X_{ij})\right)\left(w_i w_j^T - S_{ij}\right)^2$ can enforces their similarity fit S_{ij} well. The gradients of parameters take the form

$$\begin{aligned}\frac{\partial J}{\partial w_i} &= \sum\nolimits_{j \neq i} \left(f(X_{ij})\left(S_{ij} - 1\right) - S_{ij} + w_i w_j^T\right) w_j^T \\ \frac{\partial J}{\partial w_j} &= \sum\nolimits_{j \neq i} \left(f(X_{ij})\left(S_{ij} - 1\right) - S_{ij} + w_i w_j^T\right) w_i\end{aligned} \tag{6}$$

During the training process, each embeddings w_i is investigated and w_j is sampled with a strategy. That is, we extract all nodes satisfied $f(X_{ij}) > 0$ and sample nodes satisfied $f(X_{ij}) = 0$ in a probability *prob* (a hyper-parameter) to build a set D. We employ Adam algorithm to update both w_i and $w_j \in D$.

4 Experiments

This section evaluates the performance of concept2vec using an ontology built from the nouns in WordNet. The hyper-parameters in the experiments are listed in Table 2. They are identified by the hill-climbing algorithm.

4.1 Explore the Performance of Concept2vec on Calculating Semantic Similarity

In this section, we design an experiment to compare the performance of concept2vec with baselines on calculating semantic similarity. WordNet does not give exact semantic similarity among concepts, but we can learn certain semantic relationships. For example, we can safely conclude that for any a node in ontology, it has a shorter semantic distance with its siblings or children than with those which have a larger path-length over 4. Thus

Table 2. The parameters of concept2vec

Context window	10	γ	0.2
η	0.004	φ	0.55
d	0.8	ρ_1	0.1
x_{max}	100	ρ_2	0.99
α	0.7	*prob*	0.1
ρ	1.3		

we employ the idea of negative sampling to build the dataset. First, randomly select a node and one of its sibling or children, then randomly select a node that has a path-length over 4 with it. Consequently, we build a dataset in which each record contains two pairs of concept and a label that indicates whether first pair of concept has a higher semantic similarity than the second pair. The dataset contain 3000 records. Table 3 exhibits a sample where concepts occur in Fig. 1.

Table 3. The example of dataset

First pair of concept (cp1)		Second pair of concepts (cp2)		Label
Soccer	Rugby	Soccer	Domesticity	Yes
Football	Shinny	Soccer	Football	No

We employ concept2vec and baselines including node2vec and TransE to learn concept embeddings, subsequently investigate whether they can give a correct label regarding records. We generate the label "Yes" for a record satisfying $|\text{similarity}(\text{cp1}) - \text{similarity}(\text{cp2})| \leq 0.1$, otherwise "No". The hyper-parameters of baselines used in experiments are listed in Table 4.

Table 4. Hyper-parameters of baselines

Algorithm	Hyper-parameter	Description
Node2vec	iter = 100	Number of epochs in SGD
	Win = 10	Window size
	Step = 20	Length of walks per node
	Walks = 80	Number of walks per source
TransE	Margin = 1	

The experiment employs the accuracy as metric. Table 5 exhibits the experimental results.

Table 5. The accuracy of models

Concept2vec	Node2vec	TransE
98.4%	89.8%	67.1%

We can observe that TransE has little ability to distinguish concepts pairs that have different semantic similarity. Translation-based model focuses on completing KB, and do not meet the challenge of calculating the semantic similarity of concepts. Node2vec have a better performance than TransE, but significantly underperform concept2vec. Those models for the representation learning of network do not address the specific semantic relationships discussed in Sect. 1.

5 Conclusion

Inspired by *Golve*, this paper develops a method, called concept2vec, to deal with learning representation of concepts in the ontology. We build an ontology based on WordNet, and then employ concept2vec to generate concept embeddings for every concepts in the ontology. We compare concept2vec with baselines including node2vec and TransE. The experimental results show that concept2vec outperforms the baselines. We thus conclude that concept2vec is effective on learning representation of ontology.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (71571145).

References

1. Bordes, A., et. al.: Translating Embeddings for Modeling Multi-relational Data, NIPS 2013
2. Pennington, J., Socher, R., Manning, C.D.: *Golve*: global vectors for word representation. In: The Proceedings of EMNLP 2014
3. Hao, J., et. al.: Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts. KDD 2019
4. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
5. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of KDD 2016
6. Hinton, G.E.: Learning distributed representations of concepts. In Proceedings of the 8th Annual Conference of the Cognitive Science Society (1986)
7. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD (2014)

ATextCNN Model: A New Multi-classification Method for Police Situation

Wenhuan Wang[1], Ding Feng[1], Bohan Li[1,2,3](✉), and Jiaying Tian[1]

[1] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
{wangwenhuan,bhli}@nuaa.edu.cn

[2] Key Laboratory of Safety-Critical Software, Ministry of Industry and Information Technology, Nanjing 211106, China

[3] Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, Jiangsu, China

Abstract. The information age poses new challenges to the automatic and rapid classification of public security data. To get rid of the low-efficiency work mode of manual identification, perform rapid automatic classification of police situation, and improve the performance and accuracy of classifiers, many models and algorithms have been proposed. The traditional police situation classification methods almost always use traditional machine learning algorithms. The shortcomings of complex working mode, heavy workload, and poor learning effect have led to the rise of deep learning classification methods. This paper proposes the ATextCNN model, which introduces an attention mechanism in the Input Layer based on the TextCNN model, and uses a word filtering algorithm to calculate the contribution of words to filter nonsense words. The ATextCNN model develops a multi-classification model based on deep learning. We conduct experiments on two real data sets and divided data into nine categories. Experiments show that the ATextCNN model is better than the traditional classification methods in terms of accuracy and efficiency when performing multi-classification tasks, and is better than the state of the art method in practical application. Therefore, our method is more conducive to police situation identification and management.

Keywords: Police situation classification · Attention mechanism · Word filtering algorithm · Contribution degree

This work is supported in part by National Natural Science Foundation of China (61728204), Innovation Funding (NJ20160028, NT2018027, NT2018028, NS2018057), Aeronautical Science Foundation of China (2016551500), State Key Laboratory for smart grid protection and operation control Foundation, Association of Chinese Graduate Education (ACGE).

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 135–147, 2020.
https://doi.org/10.1007/978-3-030-65390-3_11

1 Introduction

The Ministry of Public Security's informatization attempts have emerged for the rapid development of information technology and Internet high-tech solutions have become normal. The Fifth Plenary Session of the 18th CPC Central Committee clarified the big data strategy and promoted the open sharing of data resources. In the field of rule of law, the police situation classification mechanism is also undergoing technological innovation. An police situation classification mechanism based on big data strategy and intelligent classification model which based on prediction, early warning and prevention has been preliminarily formed [6]. In the past, the police situation was mainly classified manually, and the work efficiency was low. A method that can quickly identify the core of police situation is needed to improve the efficiency of the public security organs for processing the police situation and get rid of the traditional police situation processing methods. In response to the above problems, through the classification and analysis of the police information, an automatically and quickly determine the police information category model to make corresponding processing, that is, to classify the police situation text is necessary.

Text classification is a basic problem of Natural Language Processing (NLP). There are many applications of text classification, such as document organization, news filtering, spam detection, information retrieval, and network public opinion discovery. The police situation text studied in this paper are also a kind of text, which has the characteristics of wide range of content, various types, and extensive textual words. Due to the confidentiality and particularity of the police situation text, few people have studied it at present. The length of the police situation text is generally between 5 and 300 words, and its classification belongs to short text classification.

For the special text of police information content, analyze the police information according to different categories is necessary. Police situation's category mainly divided into nine types, namely: help-dispute, theft and robbery, financial fraud, homicide, black evil, gambling, pornography and drugs, natural disaster and others. Distinguish the police situation, and then carry out different processing operations. Traditional methods of police situation classification mainly use machine learning algorithms. For example, [17] uses Naive Bayes (NB) to classify police information, which realizes the automatic classification of police situation and improves the efficiency of warning work to a certain extent. However, due to our high requirements on the extraction of police situation text and keywords in warning messages, classification is easily affected by many nonsense-words. Moreover, when texts are sparse, feature learning will be inaccurate, classification accuracy will be relatively low, and model can not meet the requirements of practical application scenes. Therefore, the overall effect is not ideal.

The use of deep learning methods for text classification has become a research hotspot in text classification today. The integration of related methods of deep learning can better improve the accuracy of text classification, but the cost of training is relatively increased. Many studies have applied deep learning into the analysis of Sina wei-bo comments, movie reviews, news classification, sentiment analysis, etc., and have achieved great experimental results. Based on the thinking of [10] and considering the characteristics of the police situation text, this paper proposes a model combining the attention mechanism with the word filtering algorithm based on word contribution degree and TextCNN model, called ATextCNN model. We introduce the word embedding method of deep learning framework to reduce the complexity of the model and improve the practicality. Also, we use the attention mechanism to improve the performance and interpretability of the model. Besides, the ATextCNN model can reduce the influence of nonsense words by calculating the contribution of words.

In the task of police situation classification, the ATextCNN model extends the deep learning method, combines the attention mechanism, and uses the word filtering algorithm. The main contributions are as follows:

(i) For the first time, we propose to combine the attention mechanism with TextCNN to improve the classification accuracy when classifying the real police situation data set.
(ii) The ATextCNN model proposed in this paper uses the word filtering algorithm for the first time, based on the contribution of words to filter out high-frequency words and redundant words appearing in police situation texts that reduce the accuracy of text classification.
(iii) Experiments show that our model effectively improves the accuracy and efficiency of police situation classification, reduces labor costs, and has significant practical effects.

The remainder of this paper is organized as follows: Sect. 2 discusses the related work in text classification based on traditional machine learning method and text classification based on deep learning method. In the Sect. 3, we propose four modules to illustrate our methods. The experimental description and analysis of the results are given in Sect. 4. Section 5 will give a summary of the work of this paper and describe our future development direction.

2 Related Work

2.1 Text Classification Based on Traditional Machine Learning Method

The traditional machine learning method builds a text classification model through the existing text, and then uses the trained model to classify the unknown text. The key of text classification in traditional machine learning lies in feature extraction and the selection of classification algorithms. The purpose of feature extraction is to remove noise and improve classification performance. Traditional machine learning algorithms are most commonly used, such as K-Nearest Neighbor algorithm (KNN) [9], Naive Bayes (NB) [16], SGD algorithm [14] and Support Vector Machines (SVMs) [8]. Feature extraction of the above algorithms are not exactly the same, but they all rely on the dictionary to extract features. Feature extraction is very important for classification, so the construction of dictionaries is significant, and different types of text processing require different dictionaries. Unfortunately, no uniform dictionary exist.

2.2 Text Classification Based on Deep Learning Method

With the development of deep learning, many methods have been applied to the field of text classification. In [4], emotional tags are integrated to improve Word2Vec as a text representation method. [7] extracts the Bi-LSTM model which extends from deep learning into the analysis of social media user evaluation. [11] proposes the RCNN-HLSTM model based on deep hierarchical network for text classification. The TextCNN model for text classification is proposed in [10]. The above deep learning related algorithms have made some achievements in related fields. Compared with common machine learning methods, deep learning algorithm has better feature learning effect when data is sparse, and is not easily affected by human factors. At present, deep learning algorithms have become a research hotspot in text classification.

Deep learning mainly uses word embedding technologies (such as Word2Vec, GloVe, FastText, WordRank, Text2vec, etc.) to represent text information, calculate semantic associations between words, and learn abstract features of text. Many excellent deep neural network classification algorithms (such as Deep Belief Network(DBN) [3,12], Convolutional Neural Network(CNN) [15], TextCNN [10], Recurrent Neural Network(RNN) [5], and various improved neural network algorithms, etc.) can be used. Most research centers of text classification focus on improving the fusion based on the above-mentioned classic text representation methods and classification algorithms. However, the classification algorithm after fusion is more complicated, with relatively high time complexity, and low applicability.

Text classification algorithms based on traditional machine learning rely on building dictionaries to learn representation features, so the quality of dictionaries largely affects the text classification accuracy. Dictionary construction will be affected by human factors, does not have domain versatility, and has high labor costs and low efficiency. Therefore, this study uses deep learning methods to classify police situation data. Classification methods based on deep learning are currently the hotspots of text classification research. The ATextCNN model proposed in this paper improves the accuracy and efficiency of classification, and has remarkable effect in practical application.

3 The ATextCNN Model

From the perspective of reducing model complexity, model training time, and implementation difficulty, this paper applies the TextCNN algorithm in deep learning, and combines the attention mechanism and word filtering algorithm to construct, train, and evaluate the ATextCNN model.

3.1 Representation of Text Feature

This work uses Word2Vec to vectorize the pre-processed words of police situation data. Word2Vec is an NLP tool launched by Google in 2013. Its feature is to quantify all words so that the relationship between words can be quantitatively measured and the relationship between words can be mined. Word2Vec adopts CBOW or Skip-gram model to predict the semantic information of current words by using the context information, thus generating word vectors, realizing the mapping of word vectors, and calculating the similarity in meaning by calculating the distance between space vectors. Word2Vec overcomes the lexical and dimensional defects of a one-hot word vector and is more conducive to the representation of text. This paper uses Word2Vec in the Gensim library in Python to train word vectors. The structure stored in the final word vector is “ ‘word’ + ‘’ + ‘vector’ ”, and each word has its corresponding vector.

3.2 Attention Mechanism

Some words have strong representative categories after the pretreatment. For example, theft cases often include some words or phrases like a steal, stolen, slip into the room, or over the wall and entering the room. These representative words are often helpful to distinguish the classification of police situation, so this paper considers introducing the attention mechanism into the ATextCNN model [2].

Inspired by human attention, the Attention mechanism based on neural networks was first used in the field of visual image processing [1,13]. When people observe things, they don't see all corners of things at once, but instead, focus on a certain feature or brightly colored part of the thing. For example, when observing people, they pay attention to the face first, and when observing a puppy, pay attention to the color of the coat. Extracting the key information of words in the text is similar to the human attention mechanism. When reading short police situation texts, readers usually combine their own cognition and pay attention to some partial information to quickly grasp the theme of the text.

Example 1. Outside the barbershop, No. 1 *** Road, I had a dispute with the barbershop owner. He dragged me to wash my face without saying he would charge for it, but now he stopped me. (Emergency Reception Tel: ***)

In Example 1, the keyword is ‘dispute’. The keyword highlights that the category of the text is dispute help, which in favor of improving the accuracy of text classification. This paper applies the attention mechanism to the task of police situation text classification, so the attention of this paper refers to the words to the category of the text. Find the keyword that we can quickly filter the less attention-grabbing entries.

The essence of the attention mechanism is a softmax model based on a single-layer neural network. The input is the preprocessed police situation text's word vector, and the output is the probability that the entry belongs to each type of police situation. Assume that any input text is $W = (w_1, w_2, \cdots, w_n)$, where n is the number of words and w_i is the input word vector. Y is the output one-dimensional real vector, which can be expressed as $[y_1, y_2, \cdots, y_k]$, k is the number of categories, and y_k is the score of w_i belonging to category k. The calculation formula of Y is as follows:

$$\boldsymbol{Y} = \boldsymbol{T} \cdot \mathrm{w_i} + \mathrm{b} \tag{1}$$

where $\boldsymbol{T} = (t_1, t_2, \cdots, t_k))$ is the weight matrix and $b = [b_1, b_2, \cdots, b_k]$ is the bias term. The output Y is then converted to probability p of w_i belonging to all categories by sigmoid activation function and softmax function. The output of softmax is as follows:

$$\mathrm{p}\left(\mathrm{y} = \mathrm{r} \mid t_i; \mathrm{T}\right) = \frac{\exp\left(y_r\right)}{\sum_{j=1}^{k} \exp\left(y_j\right)} \tag{2}$$

The above formula calculates the probability that w_i belongs to the category of r. In the training stage, the model takes binary group (w_i, y_i) as the training data, where y_i is the text category of the word w_i. Equation (3) is the loss function, and the stochastic gradient descent method is used to update parameters T and b:

$$L(\mathrm{T}) = \sum_{i=1}^{m} -\log p\left(y = y_i \mid t_i; T\right) \tag{3}$$

Where m is the number of words in the training set. After the training stage, the model will output the probability value of w_i belonging to all categories and regard it as the attention vector of w_i for categories. Thus, the attention matrix A of the word can be obtained, as shown in Table 1. $A[i]$ is the attention vector of t_i, and indicating the confidence of t_i to all categories.

Table 1. The word attention matrix of A.

Attention Vector	1	2		k
t_1	$p\left(y = 1 \mid t_1; T\right)$	$p\left(y = 2 \mid t_1; T\right)$	...	$p\left(y = k \mid t_1; T\right)$
t_2	$p\left(y = 1 \mid t_2; T\right)$	$p\left(y = 2 \mid t_2; T\right)$	...	$p\left(y = k \mid t_2; T\right)$
...	...	...	...	...
t_k	$p\left(y = 1 \mid t_k; T\right)$	$p\left(y = 2 \mid t_k; T\right)$	...	$p\left(y = k \mid t_k; T\right)$

The ATextCNN model adopts attention mechanisms, adding attention mechanism in Input Layer of TextCNN.

3.3 Word Contribution

In this paper, before classification, the contribution of words to classification is calculated to filter words. The attention vector represents the probability value of the text belonging to all categories. In this paper, the probability is used as the confidence of words to police situation text. The greater the probability value, the higher the confidence that the police situation text belongs to a certain category. Therefore, for words with a relatively small maximum probability, their confidence in all categories is not high, and the contribution of such words to text classification is also weak. To improve the classification efficiency, this research filters such words before classification, namely word filtering algorithm. In this paper, the standard deviation of the attention vector is regarded as the contribution of the word to the classification. The calculation method of its contribution to the word w_i is as follows:

$$C_{w_i} = \frac{1}{k} \sum_{r=1}^{k} (att_i[r] - a)^2 \tag{4}$$

Where $att_i[r]$ is the attention of the corresponding category r, and a is the mean of attention, which equals $1/k$.

The word filtering algorithm AFILTER is expressed as Table 2. The function of Delete represents the deletion of word from the text. The algorithm needs to set the hyperparameter h, which has heuristic characteristics. In the experiment, we use cross-validation to select h and analyse the influence of h.

Table 2. The word filtering algorithm AFILTER.

Input	W, A
Output	W’
Step	$a = 1/k$ For each w_i in W $att_i[r] = A[i]$ $C_{w_i} = \frac{1}{k} \sum_{r=1}^{k} (att_i[r] - a)^2$ if($C_{w_i} < h$) Delete(W, w_i)

3.4 The ATextCNN Model

In this study, multiple classification algorithms were used for experimental comparison, and the TextCNN model with good classification efficiency was finally selected. First, the word index of a sentence is transformed by mapping to the word vector, and then the word vector is convolved.

The TextCNN model includes Input Layer, ConvID Layer, MaxPoolingID Layer, Concatenate Layer, Dropout Layer, and Dense Layer. This paper introduces the attention mechanism and word filtering algorithm to the TextCNN model when constructing the police situation text classifier. The ATextCNN model applies the attention mechanism to the Input Layer of TextCNN. Besides, in the Input Layer, the word filtering algorithm is used to filter the useless words of the police situation text. The word filtering algorithm calculates the contribution degree of each word. When the contribution degree is less than the hyperparameter h, the word is discarded, while the word is retained when it is greater than the fixed value. The purpose is to remove nonsense words in the text with less contribution or appearing in many texts. The structure of ATextCNN model is shown in Fig. 1.

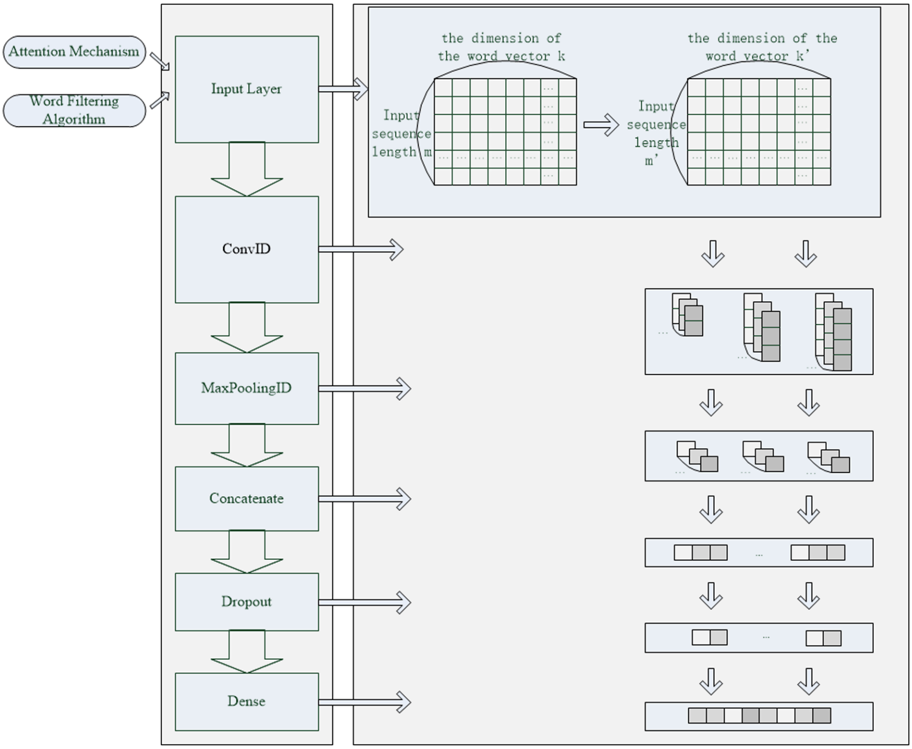


Fig. 1. The structure of the ATextCNN model.

4 Experiment and Evaluation

4.1 Experiment Data

The source of the experiment data is the police situation data of certain city in 2019, and the length of each data text is about 2 to 300 words. The main attributes of each data include information such as ‘category id’, ‘category-name’, and ‘police situation content’. Due to a large number of missing and error problems in the “category-name” attribute, this study manually marked the police situation category. The data set 1 of the police situation is shown in Table 3.

To compare the learning effects of the ATextCNN model, we extract 10,000 data with police situation content longer than 50 words in each category of help-dispute, theft and robbery, pornography and drugs, black evil, and gambling as shown in Table 4 in the last three years. Besides, in the other four categories, each category contains 10,000 data and the text content is more than 100 words.

Table 3. Police situation data set 1 of a city in 2019.

Category id	Category name	Data size	>50	>100
1	Help dispute	10000	1160	44
2	Theft and robbery	10000	2115	80
3	Financial fraud	10000	9100	1583
4	Homicide	10000	2883	525
5	Black evil	10000	1797	166
6	Gambling	10000	1142	270
7	Pornography and drugs	10000	869	79
8	Natural disaster	10000	6760	3897
9	Others	10000	5214	2991
Total/proportion	-	90000/100%	31040/34.5%	9635/10.7%

Table 4. Data set 2 with content more than 50 words or 100 words.

Category id	Category name	Data size	Content length
1	Help dispute	10000	>50
2	Theft and robbery	10000	>50
3	Financial fraud	10000	>100
4	Homicide	10000	>100
5	Black evil	10000	>50
6	Gambling	10000	>50
7	Pornography and drugs	10000	>50
8	Natural disaster	10000	>100
9	Others	10000	>100

4.2 Experiment Setting

In our experiment, police situation text on each category in two data sets is randomly divided into five folds (with four as training data and one as test data). All of the following results are reported and analyzed with an averaged accuracy of five-fold cross-validation. The performance of ATextCNN model is compared with the existing methods (i) BaseLine: TextCNN [10] which has no additional methods; (ii) State of the art: Bidirectional LSTM (Bi-LSTM) model [7] which uses a complex forward and backward recurrent neural networks to capture more contextual information.

4.3 Hyperparameter H

Figure 2 is a schematic diagram of the change of the text length and the proportion of the number of after-filtered terms before filtering with the hyperparameter h after using the word filtering method with the ATextCNN algorithm.

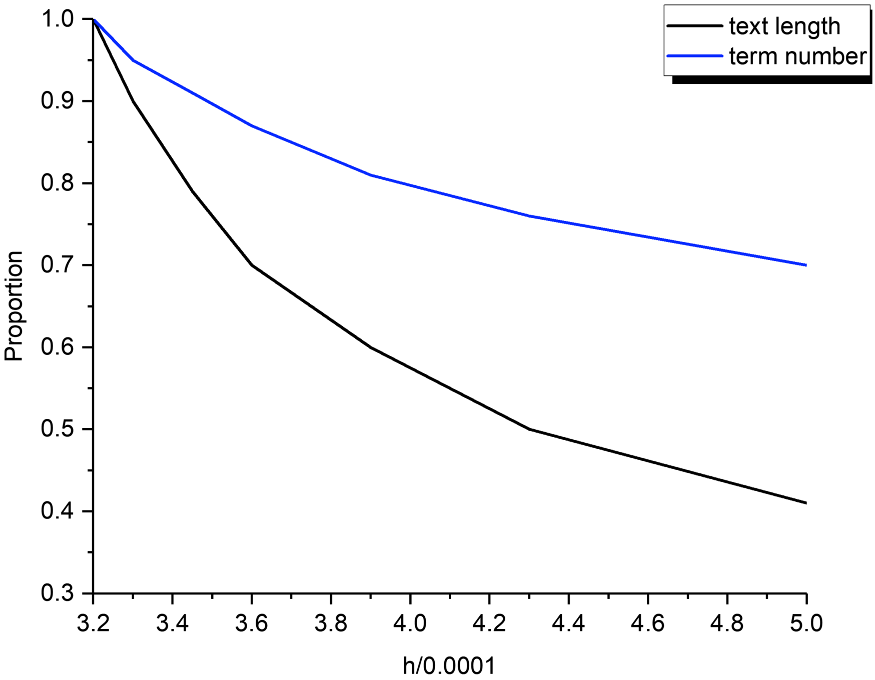


Fig. 2. Changes of text length and term number with h.

The length is the count of characters in the text, and the number of terms is the count of unique words in the text. This paper uses the 5-fold cross-validation method to record the filtering effect of h (as shown in Fig. 2). The result shows that when h is greater than 5.0 * 0.0001, the classification accuracy will less than 0.7, and when h is less than 3.2 * 0.0001, the number of filtered words tends to 0. For the graphic effect, the test interval of hyperparameter h is selected as [3.2 * 0.0001,5 * 0.0001]. Figure 2 denotes that when h is 3.2 * 0.0001, no word will be filtered. When h = 5.0 * 0.0001, the number of words retained is 70.41%, which indicates that word filtering method filters high-frequency words in police situation text. Although the text length is reduced, the terms can still be well retained. Our method is consistent with the general text filtering method, without centralized filtering for short and long sentences.

4.4 Experiment Results and Analysis

Effect of h on Accuracy. In this section, the ATextCNN model experiment is carried out on data set 2, and then the variation trend diagram of the model with hyperparameter h is obtained. As can be seen from Fig. 3, when h = 3.4 * 0.0001, the classification accuracy of the ATextCNN model is the highest; when h's value was greater than 3.4 * 0.0001, the classification accuracy gradually decreased. Therefore, to achieve better experiment results, we uniformly fixed h = 3.4 * 0.0001 in a subsequent experiment.

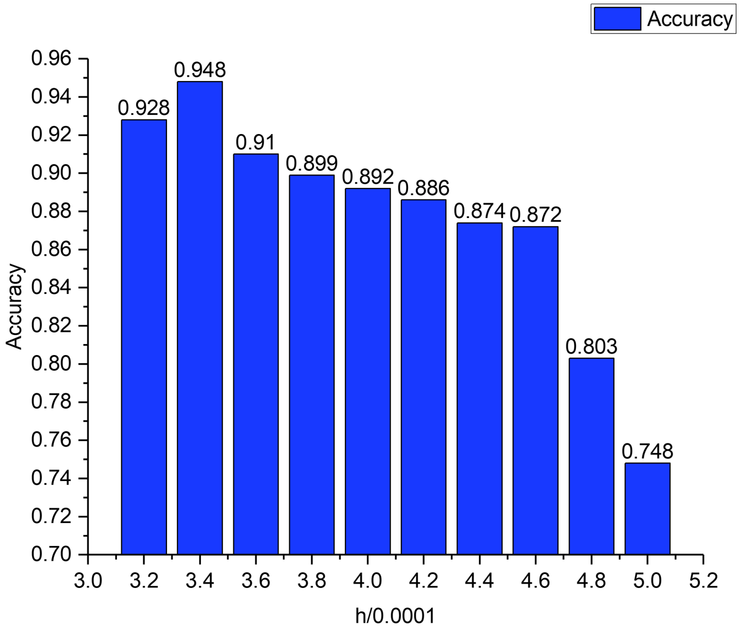


Fig. 3. Changes in accuracy of ATextCNN model with different values of hyperparameter h.

Evaluation of the ATextCNN Model. Table 5 describes the comparison of the ATextCNN model with the combination of attention mechanism and word filtering method, the TextCNN model, and the Bi-LSTM model. By comparing the accuracy of the ATextCNN model and the TextCNN model in multi-classification of police situation data, we can see the obvious effect of attention mechanism and word filtering method on the improvement of accuracy. Also, the experiment results of the TextCNN model basically meet the effect of the model in [10]. However, due to the particularity of the field of police situation text, the accuracy of the classification results is higher than expected. The ATextCNN model can break through the limitations of TextCNN for the reason that the word filtering method filters out the noise items of nonsense words and focuses on the items that meet the human attention point, which greatly improves the classification accuracy and classification efficiency.

When the data set is relatively small (such as data set 1), the ATextCNN model achieves the best results, exceeding TextCNN by 3.8%, and has a higher accuracy rate than the state of the art model. In addition, when the amount of data is relatively large, the Bi-LSTM model performs best. The reason for good performance is that the Bi-LSTM model can get better training results when the amount of data is large for its design characteristics. Bi-LSTM is suitable for modeling time-series data, such as text analysis, and the source of the text in data set 2 is the police situation data in the past three years, with good time-sequence. However, the Bi-LSTM model is more complex and requires a higher modeling environment. In the experimental period, we found that the ATextCNN model is more efficient and less resource overhead than the Bi-LSTM model. Therefore, our model has better practicability while ensuring accuracy.

Table 5. Results of our ATextCNN model against other methods.

Model	Data set 1	Data set 2
ATextCNN Model	93.7	94.8
TextCNN Model	89.9	91.2
Bi-LSTM Model	93.5	95.0

5 Conclusion and Future Work

In this paper, we use Word2Vec to train word vectors, apply the attention mechanism in the Input Layer of the TextCNN model, and introduce the concept of word contribution to filter the police situation text. Compared with previous models and algorithms, the ATextCNN model we proposed has the following advantages: (i) makes up for the feature sparsity problem of short text, which cannot be solved by traditional learning algorithm; (ii) Introduces the attention mechanism to improve the interpretability and performance of the model; (iii) the word filtering algorithm can be used to further process the police situation data, improve the classification accuracy, and reduce resource and time costs. Besides, the accuracy of the ATextCNN model in the field of text classification of crime mode is up to 97%, but limit to the length of this paper, we will not explain in detail. Since police situation data have many categories, in the future, we will optimize the model using the two-layer attention mechanism to improve the accuracy of the model and try to change the application scenario.

References

1. Choi, E., Bahadori, M.T., Sun, J., Kulas, J., Schuetz, A., Stewart, W.: Retain: an interpretable predictive model for healthcare using reverse time attention mechanism. In: Advances in Neural Information Processing Systems, pp. 3504–3512 (2016)
2. He, R., Lee, W.S., Ng, H.T., Dahlmeier, D.: An unsupervised neural attention model for aspect extraction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 388–397 (2017)
3. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)

4. Hu, C., Xiaowei, Z.: Sentiment analysis based on word vector technology and hybrid neural network. Appl. Res. Comput. **35**(12), 3556–3559+3574 (2018)
5. Jagannatha, A.N., Yu, H.: Structured prediction models for RNN based sequence labeling in clinical text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing, vol. 2016, p. 856. NIH Public Access (2016)
6. Jin, G., Zhu, S., Lin, X.: Analysis and prediction on crime in china (2017–2018). J. People's Public Secur. Univ. Chin. (Soc. Sci. Ed.) **34**, 29–38 (2018)
7. Jin, Z., Han, Y., Zhu, Q.: A sentiment analysis model with the combination of deep learning and ensemble learning. J Harbin Inst Technol **50**(11), 32–39 (2018)
8. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0026683
9. Khamar, K.: Short text classification using KNN based on distance function. Int. J. Adv. Res. Comput. Commun. Eng. **2**(4), 1916–1919 (2013)
10. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
11. Liu, Q., Liang, B., Xu, J.: A deep hierarchical neural network model for aspect-based sentiment analysis. Chin. J. Comput. **41**(12), 2637–2652 (2018)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
13. Mnih, V., Heess, N., Graves, A., et al.: Recurrent models of visual attention. In: Advances in Neural Information Processing Systems, pp. 2204–2212 (2014)
14. Prasetijo, A.B., Isnanto, R.R., Eridani, D., Soetrisno, Y.A.A., Arfan, M., Sofwan, A.: Hoax detection system on indonesian news sites based on text classification using SVM and SGD. In: 2017 4th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), pp. 45–49. IEEE (2017)
15. Wang, S., Huang, M., Deng, Z.: Densely connected CNN with multi-scale feature attention for text classification. In: IJCAI, pp. 4468–4474 (2018)
16. Wang, W., Li, B., Feng, D., Zhang, A., Wan, S.: The OL-DAWE model: tweet polarity sentiment analysis with data augmentation. IEEE Access **8**, 40118–40128 (2020)
17. Zhu, Y., Shang, M., Liang, H.: Design and implementation of automatic categorization system of public security information. Master's thesis, University of Electronic Science and Technology of China (2015)

Hierarchical and Pairwise Document Embedding for Plagiarism Detection

Ruitong Zhang, Lianzhong Liu, Jiaofu Zhang, Zihang Huang, Caiwei Yang,
Liangxuan Zhao, and Tongge Xu(✉)

School of Cyber Science and Technology, Beihang University, Beijing, China
{rtzhang,lzliu,sy1939129,pppihf,yang329,zhaolx,xutg}@buaa.edu.cn

Abstract. The rapid development of the Internet, especially the application of search engines and machine translation, makes it easier to copy texts. Most existing text plagiarism detection methods are not capable of dealing with the increasing number of plagiarism sources and the increasingly ambiguous plagiarized texts. In this paper, we pay attention to the task of large-scale text deduplication, and propose a multi-level distributed text computing model, which improves the checking speed through multi-level latent semantic analysis, and combines BERT to judge plagiarized text more accurately. In order to further verify the model, we also combined the latest fuzzy plagiarism technology to construct a three-level data set. The experimental results show that our model performs well when plagiarism data increases and plagiarism ambiguity increases.

Keywords: Plagiarism detection · BERT · LSA

1 Introduction

Text plagiarism detection can also be called deduplication or approximate duplication detection, and its main purpose is to detect plagiarism defined as the use of ideas, concepts, words, or structures without authorization [2]. Among them, text plagiarism has received great technical support due to the development of natural language technology [22,24,25] in recent years. First of all, in the aspect of plagiarism data acquisition, we can use the more accurate search engine [3] to obtain relevant texts from the huge resources of the Internet. Secondly, plagiarism can be constructed by using automatic fuzzy plagiarism software or multiple translations through automatic translation [4]. The update of these technologies makes the cost of plagiarism cheaper and makes the speed of plagiarism faster [5].

In response to these problems, we are inspired by other natural language processing fields, such as large-scale text classification [18,19,23] and clustering tasks [21], which perform hierarchical text deduplication calculations on suspicious document pairs [20]. Finally, this article provides a hierarchical plagiarism detection model used to deal with fuzzy text. The first document layer uses expert knowledge and word embedding method [1] to narrow down document

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 148–156, 2020.
https://doi.org/10.1007/978-3-030-65390-3_12

level candidates based on tag category recognition. Then, a paragraph layer is added to the second layer, which builds a paragraph-level semantic space based on latent semantic analysis [6]. At last, embedded in sentence level, we accurately detect fuzzy text plagiarism in sentence level semantics by BERT [7]. Compared with the existing methods, the checking speed and accuracy of plagiarized text judgment are effectively improved and improved.

2 Related Work

External Plagiarism Detection. External Text plagiarism detection [8,9] is a common sub-direction of plagiarism detection task. External plagiarism detection is performed by comparing the search similarity between suspicious documents and reference documents. Most external plagiarism detection researches divide the task into two stages [10]. First, candidate documents are narrowed to a subset by candidate retrieval [11], and then the similarity of paired data is analyzed by text alignment [12].

Plagiarism Detection Based on Characters. The traditional method adopts the plagiarism detection method based on characters, which mainly compares character features such as characters, strings or words. E.g., digital fingerprint [13], word bag model, vector space [14], etc. However, this method has a limitation, that is, it can't identify synonym substitution, and it is only suitable for identifying plagiarism of copying and pasting because of the loss of the original semantics of data in specific application scenarios.

Plagiarism Detection Based on Semantics. Semantic similarity analysis is used to compare the meanings of sentences, paragraphs or documents. Existing semantic text analysis methods, such as latent semantic analysis (LSA) [15], explicit semantic analysis (ESA) [16] and word embedding [17], are mainly used to detect external plagiarism, which is a universal and successful method. Although this method is effective in identifying fuzzy plagiarized texts, it also has the disadvantage of slow speed.

3 Method

In this section, we will introduce our proposed method of large-scale plagiarism detection based on multi-level structure. Figure 1 shows the frame of our model. Here, we use three levels of content embedding to analyze the coincidence degree between the submitted documents and the corresponding levels of database documents, specifically including document level embedding, paragraph level embedding and sentence level embedding.

3.1 Document Level Embedding

Document layer, as the first level of direct input of documents to be detected, aims to find other documents with high similarity to their themes by taking documents as embedded objects. For the problem formalization, given a candidate document set $D_C = (d_{C,1}, ..., d_{C,n})$, when inputting a document d_I to be detected, the task of the document layer is to retrieve the suspicious document set $D_S = (d_{S,1}, ..., d_{S,k})$ suspected to be plagiarized from the candidate document set D_C. Thereby outputting a document layer detection result, namely a suspicious document pair set $Pair_D = (pair_{D,1}, ..., pair_{D,k})$, wherein each $pair_D = (d_I, d_{S,j})$ represents a document pair combination of a document to be detected and a candidate document suspected of being plagiarized. We define the tasks of document level detection as follows:

$$Pair_D = DocLevelDetect(d_I, D_C) \tag{1}$$

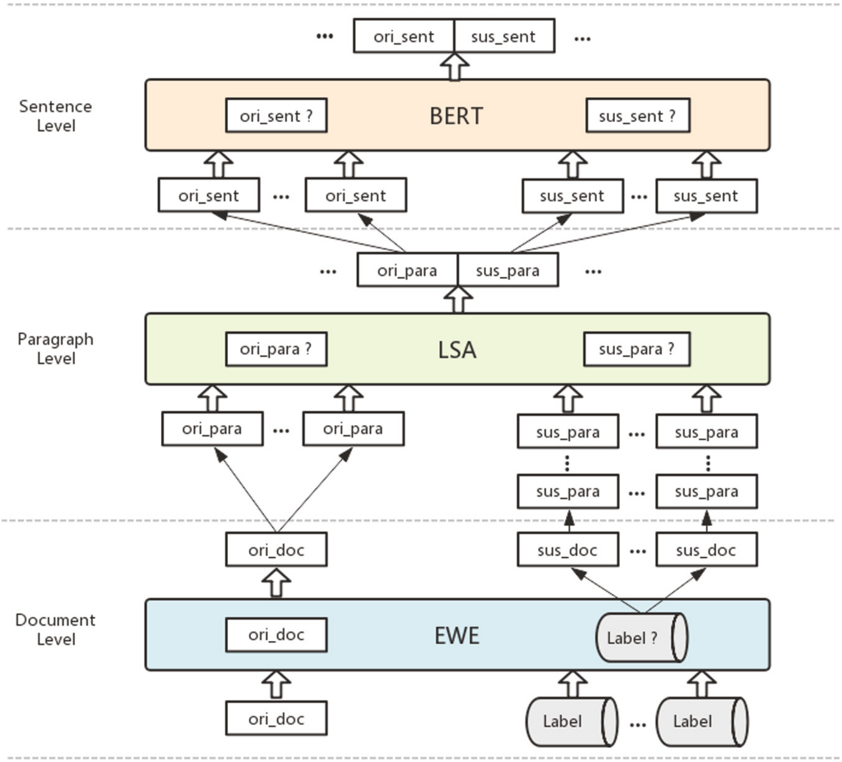


Fig. 1. Multi-level structure text plagiarism

Different from the traditional similarity comparison method, instead of directly comparing the embedding matrices of two documents, we use an unsupervised text classification method using experts and word embedding [1]. Based on this method, the category label is used as the intermediate comparison standard, and the documents to be detected and candidate documents are compared with the predefined categories, so as to find suspicious documents with the same theme as the documents to be detected.

3.2 Paragraph Level Embedding

Before more accurate sentence level detection, some work is needed to further narrow and locate the search scope, which is the reason for paragraph level embedding. For paragraph level, we also give a formal description of the problem. When the input suspicious document pair set $Pair_D = (pair_{D,1}, ..., pair_{D,k})$ is obtained from the previous layer, the task of paragraph level is to analyze and retrieve all documents in paragraph units, so as to obtain a set of pairs $Pair_P = (pair_{P,1}, ..., pair_{P,z})$ of input document paragraphs and candidate document suspicious paragraph combinations, where $pair_P = (p_{I,i}, p_{S,j})$ and p is the paragraph in the document. We define the tasks of paragraph level detection as follows:

$$Pair_P = ParaLevelDetect(Pair_D) \tag{2}$$

For paragraph level detection, we use latent semantic analysis [6]. After obtaining the Term-Document matrix of all paragraphs in the candidate paragraph pair set, the potential semantic space of all paragraphs is obtained by singular value decomposition. For each candidate paragraph pair, we compare the two paragraphs by cosine similarity. And each candidate paragraph pair whose similarity reaches the threshold is regarded as a suspicious paragraph pair.

3.3 Sentence Level Embedding

Sentence level is a more fine-grained detection, through which we can locate specific sentences suspected of plagiarism and plagiarized. For the formalization of sentence-level problems, we express that after giving a set of suspicious paragraph pairs, we check the internal sentences of paragraph pairs, so as to find suspected plagiarized sentences and suspected plagiarized sentence pairs $Pair_S = (pair_{S,1}, ..., pair_{S,h})$, where $pair_S = (s_{I,i}, s_{S,j})$ and s is the sentence in the document. We define the tasks of sentence level detection as follows:

$$Pair_S = SentLevelDetect(Pair_P) \tag{3}$$

For sentence level detection, we choose BERT [7] to compare the semantic similarity of each candidate sentence pair more directly. In order to be more suitable for plagiarism detection environment, we construct a variety of plagiarism sentence pairs and fine-tune the BERT pre-training model. The specific construction methods of plagiarism sentence pairs include: direct copy and paste plagiarism, synonym substitution plagiarism, and multilingual translation conversion plagiarism.

4 Experiments

For evaluating plagiarism detection tasks of large-scale data sources, we pay attention to two important factors: accuracy and speed. For the accuracy evaluation method, we tested the accuracy at different levels of embedding. As for the speed evaluation method, we pay attention to the detection time of single document detection in different scale candidate data sources, and compare it with the sentence model without hierarchical structure.

4.1 Datasets

In order to evaluate our model, we pay attention to the examination of long academic articles. In the experiment, we collected 2000 Chinese academic articles classified by different disciplines from Google Academic and HowNet, which were used as experimental data sets. Then, verify and test the model, we construct plagiarized texts and non-plagiarized texts of corresponding levels based on different levels of raw data obtained after preprocessing, thus obtaining annotated document-level data pairs, paragraph-level data pairs and sentence-level data pairs.

4.2 Results and Discussion

Accuracy and Speed Analysis: Table 1 reports the accuracy analysis of our model at document level, paragraph level and sentence level. It can be observed that the overall performance level meets our expectations. The sentence level uses BERT with an accuracy of 99.9% for semantic calculation, but it also causes a great demand for calculation. In order to improve the overall speed, we added document layer and paragraph layer. Figure 2 shows the comparison of the running time between our full model with hierarchical structure and the traditional model with only sentence layer in different candidate document data scales. In the process of increasing the number of candidate data documents from 250 to 1500, we found that the multiple gap of running time gradually increased from 68 times to 74 times. It shows that when dealing with the larger plagiarism detection data sources, our model has better effect than the model that directly analyzes the similarity of sentence vectors, that is, the model with only sentence layer.

Table 1. Accuracy of documents, paragraphs, sentences and overall models

	Document Layer	Paragraph Layer	Sentence Layer	Overall
Accuracy	94.12%	95.36%	99.90%	89.66%

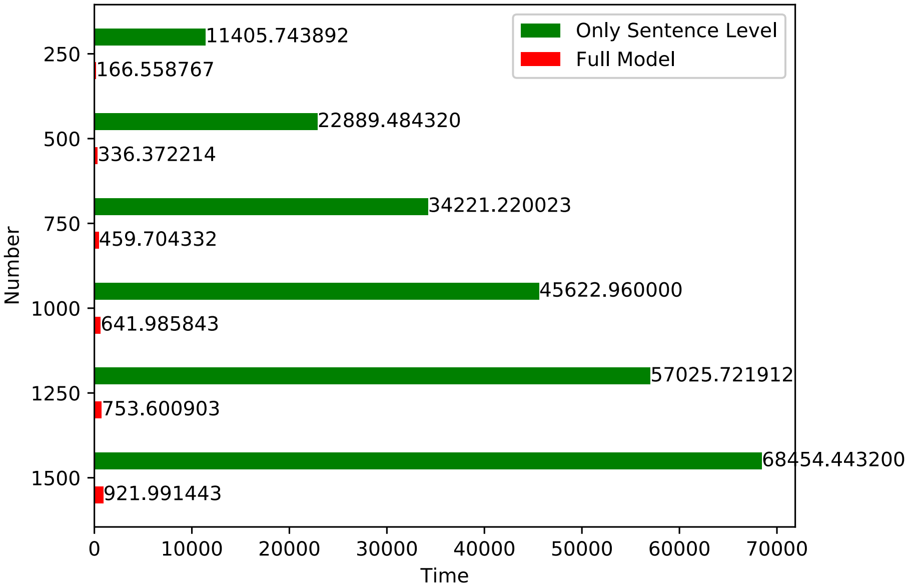


Fig. 2. Running time diagram of data sets with different architectures and scales

Case Analysis: Finally, in order to better understand our model, an example of the output result of the whole model is given in Table 2. The final model outputs plagiarized sentence pairs exceeding the threshold as sentence units. It includes the precise positioning of the sentences in the documents to be detected and in the database documents. Due to the limitation of space, we selected three representative results from the results. From the results of the report, we can see that our model can not only identify the directly copied and pasted text, but also find and locate the copied text with certain ambiguity. For example, we can find that in Example 3, copying sentences changed the semantic structure and words of the original sentences, but we still found plagiarism accurately.

Table 2. Three typical output cases of the model

Case 1 (Probability 0.9993)
Sentences suspected of plagiarism 23-1-2: The following problems are seriously ignored in the application of language processing technology in the current word mnemonic tool software
Plagiarized sentences 53-3-2: The following problems are seriously ignored in the application of language processing technology in the current word mnemonic tool software
Case 2 (Probability 0.9987)
Sentences suspected of plagiarism 238-2-10: In this project, natural language processing and big data analysis are combined to design a comprehensive algorithm, which makes the word memory software get rid of individual tools and become a tool that can be applied to groups
Plagiarized sentences 5-1-9: In this project, natural language processing and big data analysis are combined to design a comprehensive algorithm to solve this problem
Case 3 (Probability 0.9990)
Sentences suspected of plagiarism 111-5-1: Here, if you read a lot of English literature, you don't need endnotes when translating English for reference
Plagiarized sentences 26-3-27: What I want to say is that if a person reads many foreign papers, it is not necessary to indicate at the end to quote foreign languages

5 Conclusion

In this paper, we present a large-scale semantic computing model based on hierarchical structure. Experiments show that BERT has a good effect on text deduplication calculation, and the latent semantic space also solves the need of adding document layer and paragraph layer, that is, the speed problem of large-scale text. Although the advantages of our model are obvious, we also hope to further improve the overall performance through more in-depth research in terms of more delicate thesaurus and more effective paragraph semantic analysis.

Acknowledgments. This work is supported by the National Key Research and Development Program of China under the Grant No. 2018YFC0830804.

References

1. Haj-Yahia, Z., et al.: Towards unsupervised text classification leveraging experts and word embeddings. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 371–379 (2019)
2. Teddy, F.: "We know it when we see it"? Is not good enough: toward a standard definition of plagiarism that transcends theft, fraud, and copyright. In: Proceedings of the 4th Asia Pacific Conference on Educational Integrity, pp. 28–30 (2009)

3. Halavais, A.: Search Engine Society, 2nd edn. Cambridge University Press, Cambridge (2017)
4. Johnson, M., et al.: Google's multilingual neural machine translation system: enabling zero-shot translation. Trans. Assoc. Comput. Linguist. **5**, 339–351 (2017)
5. Hagen, M., Potthast, M., Adineh, P., Fatehifar, E., Stein, B.: Source retrieval for web-scale text reuse detection. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 2091–2094. ACM, November 2017
6. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**(6), 391–407 (1990)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv https://arxiv.org/abs/1810.04805 (2018)
8. Alzahrani, S., Salim, N.: Fuzzy semantic-based string similarity for extrinsic plagiarism detection. In: Braschler, D., Harman, M. (eds.) vol. 1176, pp. 1–8 (2010)
9. Gupta, D.: Study on extrinsic text plagiarism detection techniques and tools. J. Eng. Sci. Technol. Rev. **9**(5), 8–22 (2016)
10. Foltýnek, T., Meuschke, N., Gipp, B.: Academic plagiarism detection: a systematic literature review. ACM Comput. Surv. (CSUR) **52**(6), 1–42 (2019)
11. Asadi, N., Lin, J.: Effectiveness/efficiency tradeoffs for candidate generation in multi-stage retrieval architectures. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 997–1000 (2013)
12. Véronis, J., Langlais, P.: Evaluation of parallel text alignment systems. In: Véronis, J., (eds) Parallel Text Processing, vol. 13, pp. 369–388. Springer, Dordrecht (2000). https://doi.org/10.1007/978-94-017-2535-4_19
13. Alvi, F., Stevenson, M., Clough, P.: Plagiarism detection in texts obfuscated with homoglyphs. In: Jose, J.M., et al. (eds.) ECIR 2017. LNCS, vol. 10193, pp. 669–675. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56608-5_64
14. Erfaneh G., Kayvan B., Kiarash Z., Hadi V.: A deep learning approach to Persian plagiarism detection. In: Proceedings of the Forum for Information Retrieval Evaluation, pp. 154–159 (2016)
15. Alfikri, Z., Purwarianti, A.: Detailed analysis of extrinsic plagiarism detection system using machine learning approach (Naive Bayes and SVM). Telkomnika Indones. J. Electrical Eng. **12**(11), 7884–7894 (2014)
16. Jiang, Z., Chen, M., Liu, X.: Semantic annotation with rescoredesa: rescoring concept features generated from explicit semantic analysis. In: Proceedings of the 7th International Workshop on Exploiting Semantic Annotations in Information Retrieval, pp. 25–27 (2014)
17. Glavaš, G., Franco-Salvador, M., Ponzetto, S.P., Rosso, P.: A resource-light method for cross-lingual semantic textual similarity. Knowl.-Based Syst. **143**, 1–9 (2018)
18. Peng, H., et al.: Large-scale hierarchical text classification with recursively regularized deep graph-CNN. In: Proceedings of the 2018 World Wide Web Conference, pp. 1063–1072 (2018)
19. Peng, H., et al.: Hierarchical taxonomy-aware and attentional graph capsule RCNNs for large-scale multi-label text classification. IEEE Trans. Knowl. Data Eng. (2019)
20. Sun, Q., et al.: Pairwise learning for name disambiguation in large-scale heterogeneous academic networks. arXiv https://arxiv.org/abs/2008.13099 (2020)
21. Yang, R., et al.: Performance-aware speculative resource oversubscription for large-scale clusters. IEEE Trans. Parallel Distrib. Syst. **31**(7), 1499–1517 (2020)

22. He, Y., Li, J., Song, Y., He, M., Peng, H.: Time-evolving text classification with deep neural networks. In: IJCAI, pp. 2241–2247 (2018)
23. Arif, M. H., Li, J., Iqbal, M., Peng, H.: Optimizing XCSR for text classification. In: 2017 IEEE Symposium on Service-Oriented System Engineering (SOSE), pp. 86–95(2017)
24. Bao, M., Li, J., Zhang, J., Peng, H., Liu, X.: Learning semantic coherence for machine generated spam text detection. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2019)
25. Yan, H., Peng, H., Li, C., Li, J., Wang, L.: Bibliographic name disambiguation with graph convolutional network. In: Cheng, R., Mamoulis, N., Sun, Y., Huang, X. (eds.) WISE 2020. LNCS, vol. 11881, pp. 538–551. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34223-4_34

Graph Mining

Evolutionary Strategy for Graph Embedding

Jin Jin[1](✉) and Dan Yu[2,3]

[1] Chengdu Neusoft University, Chengdu 611844, China
jinjin@nsu.edu.cn
[2] Neusoft Education Technology Co. Limited, Dalian, Liaoning, China
[3] Dalian Neusoft University of Information, Dalian, Liaoning, China

Abstract. Graph embedding is an important method for learning low-dimensional representations of vertices in graph data. The problem of graph embedding requires that a better embedding method be used to optimize the corresponding objective function. There are two challenges associated with graph embedding. First, the optimization algorithm is based on gradient descent and falls easily into the local optimum. Second, whether the objective function design is reasonable has a huge impact on the embedding results. To tackle this two challenges, evolutionary strategies are used as the optimization algorithm for graph embedding. Evolutionary strategies do not need to know the specific analytical form of the objective function, and can effectively overcome the challenge of the problem of optimum. In addition, to tackle the challenge of the objective function, this paper improves on the design of the objective function based on the previous research. To verify the effectiveness of the algorithm, experiments on multi-label classification tasks were carried out on four real network data sets. Experiments show the effectiveness and potential of evolutionary strategy for graph embedding.

Keywords: Gradient-free · Graph embedding · Evolutionary strategy

1 Introduction

Graphs such as social networks, word coexistence networks, and communication networks exist in a wide variety of real-world applications. Real world graphs are often high dimensional and difficult to handle. Graph embedding algorithms have achieved excellent performance in graph representation.

There are two challenges associated with graph embedding. First, the optimization algorithm is based on gradient descent and falls easily into the local optimum. Second, whether the objective function design is reasonable has a huge impact on the embedding results.

In this paper, the evolutionary strategies are adopted to deal with the graph embedding problem. Evolutionary strategy is an algorithm based on population iteration. It does not require the gradient information of the objective function [10]. Besides, to overcome the limitations of first-order approximation, the

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 159–167, 2020.
https://doi.org/10.1007/978-3-030-65390-3_13

method proposed in this paper further improves the objective function, and combines first-order approximation and second-order approximation to characterize the network structure more efficiently.

The contributions of this paper are as follows:

Firstly, use evolutionary strategies instead of gradient algorithms to implement the graph embedding optimization algorithms.

Secondly, use a combination of first and second order approximations to characterize the network structure.

Thirdly, the proposed graph embedding model will be used to verify node multi-label classification for all four large real-world data sets. Experimental results have demonstrated the potential of the proposed algorithm.

The remainder of this paper is as follows. Section 2 reviewed the graph embedding method. Then Sect. 3 summarizes the proposed algorithm. Section 4 presents the experimental results. Finally, the conclusions to this study and ideas for future work are outlined in Sect. 5.

2 Related Work

Graph embedding is inspired by word vector learning in natural languages. The DeepWalk method obtains the local context information of the nodes in the network through random walk. To better represent the structural similarity in the graph, LINE [8] carefully designs the objective function to balance the local and global network structures. In addition, in order to simplify the computational problems of large-scale information networks, LINE adopts an edge sampling method. HOPE [4] extends LINE to preserve high-order proximity by decomposing the similarity matrix rather than adjacent matrix.

Node2vec [2] uses a biased random walk on top of Deepwalk, which breadth-first (BFS) and depth-first (DFS) graph features are balanced.

DeepWalk [5] and Node2vec randomly initialize node embedding to train the model. Since their objective functions are non-convex, such initializations are likely to fall into local optimality.

SDNE [9] recommends using depth autoencoders. The autoencoders aims to maintain first - and second-order network proximity. By jointly optimizing these two approximations, SDNE can achieves better embedding results. On the other hand, the powerful feature extraction capabilities of deep learning provide better support for the algorithm.

The deep neural network based approach, SDNE and Deep neural networks for learning graph representation(DNGR) [1] are computationally expensive when dealing with large sparse graphs. Graph convolution network (GCN) [11] deal with this problem by defining convolution operators on graphs.

3 Proposed Algorithm

This part first gives the definition of the problem and related basic concepts, and then gives the framework of evolutionary strategy graph embedding algorithm.

3.1 Problem Definition

Definition 1 *(Graph). A graph is denoted as $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$ represents n vertexes and $E = \{e_{i,j}\}_{i,j=1}^{n}$ represents the edges. Each $e_{i,j}$ is associated with a weight $w_{i,j}$.*

The types of graphs can be directed or undirected. The weights of the edges in the graph can be positive or negative, and the weights discussed in this paper are non-negative. The weights in this paper are non-negative.

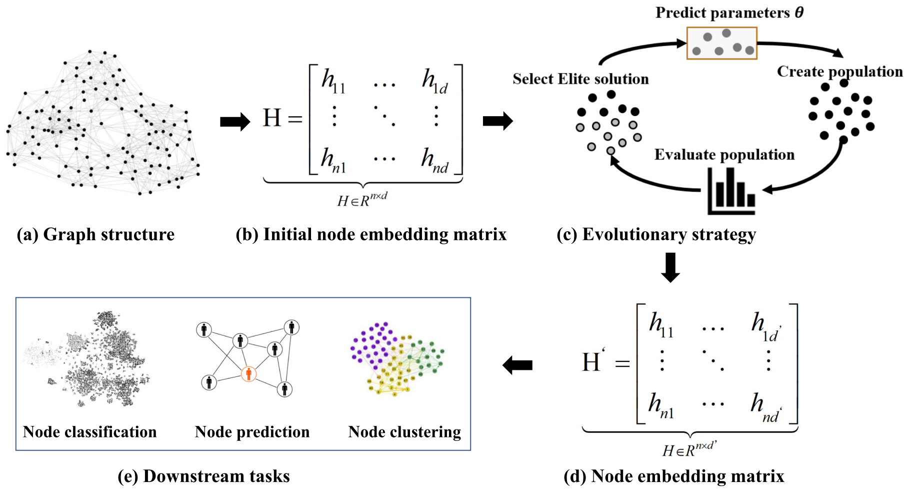
Definition 2 *(First-Order Proximity). The first-order proximity describes the relation between two nodes, which has a edge. For any pair of vertexes, if no edge is observed between node i and node j, their first-order proximity is 0.*

Definition 3 *(Second-Order Proximity). The second-order proximity describes the proximity of the pair's neighborhood structure. The second-order proximity assumes that if two vertexes share many common neighbors, they tend to be similar.*

The second-order approximation can capture the structural information of the graph.

3.2 Evolutionary Strategy Graph Embedding Framework

The general framework of the proposed evolutionary strategy graph embedding method is shown in Fig. 1.


Predict parameters θ
Select Elite solution
Create population
Evaluate population
$H = \begin{bmatrix} h_{11} & \cdots & h_{1d} \\ \vdots & \ddots & \vdots \\ h_{n1} & \cdots & h_{nd} \end{bmatrix}$
$H \in R^{n \times d}$
$H' = \begin{bmatrix} h_{11} & \cdots & h_{1d'} \\ \vdots & \ddots & \vdots \\ h_{n1} & \cdots & h_{nd'} \end{bmatrix}$
$H \in R^{n \times d'}$
Node classification
Node prediction
Node clustering

(a) Graph structure (b) Initial node embedding matrix (c) Evolutionary strategy

(e) Downstream tasks (d) Node embedding matrix

Fig. 1. Framework of evolutionary strategy graph embedding method.

The General Framework of the Algorithm

- Figure 2(a) is the graph structure used to embed the graph. Our goal is to represent the graph structure to be embedded as a vector embedded in Euclidean space. Make adjacent nodes or nodes with similar structures as close together as possible.
- Figure 2(b) is the initialized embedding matrix, where d is the embedding vector dimension of the graph embedding problem. Each row in the matrix represents an embedded vector of vertices. This matrix can be used in certain processes as the initial population for evolutionary calculation through certain processing. For example, it can be expressed as:

$$P = [p_{11} \cdots p_{nd}], P \in \mathbb{R}^{1 \times nd} \tag{1}$$

- Figure 2 (c) is the basic step of evolutionary strategy, that is, the population is initialized with certain parameters, usually Gaussian distribution parameters, the population is evaluated, the parameter update scheme is obtained from the appropriate population, the parameter is updated, and the process is kept circulating until the end condition is met.
- Figure 2(d) is the final embedding matrix obtained.
- Figure 2(e) shows the downstream algorithms, including vertex classification, vertex prediction, vertex clustering and other tasks.

The detailed pseudocode is shown in Algorithm 1.

The objective function defined in this paper borrows the idea of LINE algorithm, introduces the concept of randomness in heuristic algorithm, and controls the proportion of first-order approximation and second-order approximation by generating random Numbers. That is, the objective function is

$$O = -\gamma * \sum_{j \in N(i)} w_{ji} \log p_1 (v_j, v_i) - (1-\gamma) * \sum_{j \in N(i)} w_{ji} \log p_2 (v_j \mid v_i) \tag{2}$$

Where γ is a random number, $\sum_{j \in N(i)} w_{ji} \log p_1 (v_j, v_i)$ is the first-order proximity, and $\sum_{j \in N(i)} w_{ji} \log p_2 (v_j \mid v_i)$ is the second-order proximity.

4 Experimental Results and Analysis

In this part, we verify the effectiveness of the algorithm through a series of experiments on real word data sets.

4.1 Datasets

To evaluate the effectiveness of the embedding results, we choose multi-label classification as the downstream task. The node and edge characteristics of the dataset[1] are shown in Table 1.

[1] The data can be found from CogDL, which an Extensive Research Toolkit for deep Learning on Graphs.

Algorithm 1: The evolutionary strategy graph embedding method

```
Input:
$G = (V, E)$, Objective function $O$, $HorP$, Termination condition $C$,
Downstream task $T$, Algorithm for the downstream task M, Evaluation Metric
Q.
Output:
Embedding matrix $H'$
The value of evaluation metric $q$
Initialize:
Embedding dimension $d$, population $p$, initial parameter $\theta_0$, $t = 0$
Graph Embedding Process:
while ( $t < C$) do
    Sample $\{H_i\}_1^p$ according to $\theta_0$
    Evaluate the fitness $O(P_i)$
    Rank and select the elite population to generate $\theta_t$
    Update the population according to $\theta_t$
end
Return the best embedding H'
Downstream Task:
Evaluate the performance of H' on the downstream task;
Return $q$
```

Table 1. Statistics of the dataset

Dataset	PPI	Wikipedia	Blogcatalog	DBLP
Nodes	3,890	4,777	10,312	51,264
Edges	76,584	184,812	333,983	127,968
Labels	50	40	39	60

The downstream algorithm used in this article is multi-label classification. For multi-label classification problem, the measure metics are Micro-F1 and Macro-F1. Of course, for the graph embedding algorithm, the output of the model is not the classification itself, but a vector representing the node information. In such a situation, we usually take this vector as input and run a simple classification algorithm. For example, Logistic regression.

Baseline Methods. To verify the performance of the algorithm, the following algorithm is used as a comparison algorithm:

Spectral Clustering (SC) [7]**:** This is a clustering method based on graph theory. It directly uses the relevant properties of subgraphs to achieve the purpose of general clustering. After hierarchical clustering, the distance between the internal sub-graphs is as close as possible, and the distance between the sub-graphs is as far as possible.

Particle Swarm Optimization (PSO) [3]**:** PSO is a heuristic or called swarm intelligence algorithm inspired by Bird Flock.

Differential Evolution [6]**:** DE is similar to PSO, which is also a heuristic algorithm. DE also needs no gradient information in the optimization process.

Parameter Settings. In the experiment, a method of proportional random selection was used to distinguish the training set from the test set. Each experiment was repeated 10 times, and the average value of the performance indicators (Macro-F1 and Micro-F1) was recorded.

For the classification algorithm, this paper adopts the Logistic regression implemented by LIBLINEAR package.

Parameter $d = 500$ is the dimension for embedding, $p = 100$ is the population of the evolutionary strategy algorithm, and $C = 1.00E + 05$ is the maximum number of evaluation.

Downstream Tasks. In downstream tasks, the size of the training set are changed from 10% to 90%. Besides, we repeated the experiment 10 times and used the mean score of micro-F1 and macro-F1 as the evaluation index. The classifier is logistic regression.

Experimental Results. The experimental results are recorded in Table 2. The experimental data is the statistical mean data of 10 experiments. It can be seen from the results that the evolutionary strategy graph embedding algorithm can get better results than hierarchical clustering. However, the evolutionary strategy has implicit parallelism. When the amount of data is very large, parallel processing can quickly deal with the problem. In addition, it can be seen from the table that when the proportion of training data is changed, the performance of the algorithm is stable and healthy to some extent. It is shown that the algorithm is not only able to deal with large-scale information network, but also applicable to the real situation where data is difficult to obtain.

To further demonstrate the experimental results, this article uses a reduced version of the wiki data set to visually display the embedding effect of the algorithm. As shown in Fig. 2, the proportion of the training set is 50%, and the data of the same category are represented by the same color. Through t-SNE to visualize the embedding vector to a two-dimensional plane, it can be seen from Fig. 2 (a) that each category in the SC algorithm is relatively scattered, indicating that the effect needs to be improved. Compared with the SC algorithm, the results of the PSO and DE algorithms used in Fig. 2 (b) and (c) have improved performance. Figure 2 (d) shows the best performance when using the ES algorithm.

Table 2. The mean values of Micro-F1 and Macro-F1 on the four data sets. The proportion of the training set ranges from 10% to 90%, the rest are test set. Each test runs 10 times and averaged as the final result.

	%Labeled nodes		10%	20%	30%	40%	50%	60%	70%	80%	90%
PPI	%Micro-F1	SC	0.09	0.09	0.1	0.11	0.11	0.09	0.09	0.09	0.1
		PSO	0.11	0.11	0.11	0.11	0.12	0.11	0.11	0.11	0.11
		DE	0.15	0.15	0.15	0.15	0.15	0.14	0.15	0.15	0.15
		ES	0.2	0.21	0.21	0.22	0.23	0.24	0.24	0.25	0.16
	%Macro-F1	SC	0.12	0.12	0.12	0.13	0.12	0.12	0.13	0.13	0.13
		PSO	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
		DE	0.15	0.15	0.15	0.16	0.15	0.16	0.17	0.16	0.16
		ES	0.23	0.23	0.24	0.25	0.25	0.26	0.26	0.27	0.27
Wiki	%Micro-F1	SC	0.21	0.22	0.23	0.22	0.24	0.23	0.23	0.25	0.26
		PSO	0.24	0.26	0.25	0.25	0.26	0.25	0.26	0.26	0.27
		DE	0.26	0.27	0.26	0.26	0.27	0.28	0.27	0.27	0.27
		ES	0.33	0.34	0.34	0.35	0.36	0.37	0.37	0.38	0.38
	%Macro-F1	SC	0.22	0.23	0.24	0.23	0.25	0.24	0.24	0.26	0.27
		PSO	0.25	0.27	0.26	0.26	0.27	0.26	0.27	0.27	0.28
		DE	0.27	0.28	0.27	0.27	0.28	0.29	0.28	0.28	0.28
		ES	0.34	0.35	0.35	0.36	0.37	0.38	0.38	0.39	0.39
Blogcatalog	%Micro-F1	SC	0.27	0.3	0.31	0.32	0.33	0.34	0.35	0.35	0.36
		PSO	0.27	0.31	0.31	0.31	0.32	0.32	0.33	0.34	0.35
		DE	0.28	0.32	0.31	0.33	0.34	0.35	0.35	0.35	0.37
		ES	0.32	0.33	0.33	0.34	0.34	0.36	0.36	0.37	0.38
	%Macro-F1	SC	0.28	0.29	0.3	0.31	0.31	0.33	0.35	0.35	0.37
		PSO	0.28	0.33	0.32	0.34	0.33	0.33	0.34	0.35	0.36
		DE	0.29	0.33	0.32	0.34	0.34	0.36	0.36	0.36	0.38
		ES	0.33	0.33	0.34	0.35	0.36	0.37	0.37	0.38	0.39
DBLP	%Micro-F1	SC	0.27	0.3	0.31	0.32	0.33	0.34	0.35	0.35	0.36
		PSO	0.28	0.31	0.31	0.33	0.34	0.35	0.35	0.36	0.37
		DE	0.29	0.32	0.32	0.33	0.35	0.36	0.36	0.37	0.38
		ES	0.3	0.32	0.33	0.34	0.35	0.37	0.37	0.38	0.39
	%Macro-F1	SC	0.28	0.31	0.31	0.32	0.34	0.35	0.35	0.36	0.38
		PSO	0.3	0.33	0.34	0.35	0.36	0.36	0.37	0.37	0.38
		DE	0.31	0.34	0.34	0.35	0.37	0.38	0.38	0.39	0.4
		ES	0.32	0.34	0.35	0.36	0.37	0.39	0.39	0.4	0.41

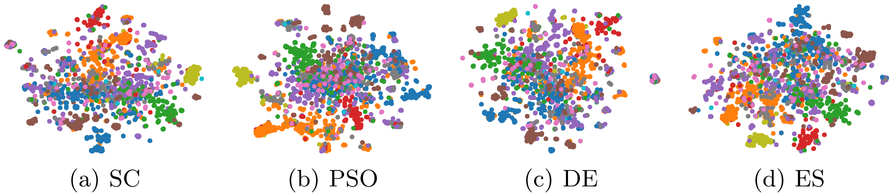


Fig. 2. Framework of evolutionary strategy graph embedding method.

5 Conclusion

This paper presents an evolutionary strategy graph embedding method. This method can deal with the limitation of gradient optimization in graph embedding. It makes the algorithm more flexible and extensible. The proposed algorithm is validated on four classical graph data sets. Experimental results show that the proposed algorithm performance better when capturing the local and global information of the graph.

The evolutionary algorithm used in this paper is a relatively primitive algorithm, but now various variants of evolutionary computation have very good performance. The next step is to further explore the use of improved evolutionary algorithms in graph embedding optimization.

References

1. Cao, S., Lu, W., Xu, Q.: Deep neural networks for learning graph representations. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
2. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864 (2016)
3. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN 1995-International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE (1995)
4. Ou, M., Cui, P., Pei, J., Zhang, Z., Zhu, W.: Asymmetric transitivity preserving graph embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1105–1114 (2016)
5. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710 (2014)
6. Price, K.V.: Differential evolution. In: Zelinka, I., Snasel, V., Abraham, A. (eds) Handbook of Optimization, vol. 38, pp. 187–214. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-30504-7_8
7. Sun, L., Ji, S., Ye, J.: Hypergraph spectral learning for multi-label classification. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 668–676 (2008)
8. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: LINE. In: Proceedings of the 24th International Conference on World Wide Web - WWW 2015 (2015)

9. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1225–1234 (2016)
10. Zhang, Z., Shao, L., Xu, Y., Liu, L., Yang, J.: Marginal representation learning with graph structure self-adaptation. IEEE Trans. Neural Netw. Learn. Syst. **29**(10), 4645–4659 (2018)
11. Zhou, J., et al.: Graph neural networks: a review of methods and applications. arXiv preprint arXiv:1812.08434 (2018)

D2NE: Deep Dynamic Network Embedding

Chao Kong(✉), Baoxiang Chen, Shaoying Li, Qi Zhou, Dongfang Wang,
and Liping Zhang

School of Computer and Information, Anhui Polytechnic University, Wuhu, China
kongchao@ahpu.edu.cn,
{bxchen1996,shyli1996,qzhou1998,qzhou1998,dfwang1998,lpzhang1980}@yeah.net

Abstract. In this paper, we propose a new approach named D2NE, short for Deep Dynamic Network Embedding, to learn the vertex representations for dynamic networks. The algorithm utilizes the graph attention mechanism to refresh embeddings efficiently, in which each update associate with local information only. To address the missing data, which is a common phenomenon in real-world networks, we model the auxiliary side information to capture more information on vertex relations. D2NE is highly efficient to refresh embeddings for dynamic networks, even with standard real-world sparse networks. We conduct extensive experiments on several real-world dynamic networks to validate the performance of D2NE in the link prediction task. Both quantitative results and qualitative analysis verify the effectiveness and rationality of our D2NE method.

Keywords: Dynamic network embedding · Graph attention mechanism · Auxiliary side information · Link prediction

1 Introduction

Network embedding is the problem of embedding network vertices into a low dimensional embedding space. As an effective and efficient solution, it converts the network data into a low dimensional space in which the network structural information and network properties are maximumly preserved. Moreover, since missing data is a common phenomenon in the real-world, the observed edges may not contain sufficient information. Previous works are focused on embedding vertices from a fixed network. Many real-world applications require embeddings generated quickly from unseen vertices because they operate on evolving networks and encounter unseen vertices constantly. Therefore, we have studied the problem of dynamic network embedding. However, the dynamic network embedding is often challenging for two primary reasons: (1) it is costly for refreshing embeddings on dynamic networks; (2) the vertices have few or even no edges in practical applications.

C. Kong and B. Chen—The two authors contributed equally to this work.

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 168–175, 2020.
https://doi.org/10.1007/978-3-030-65390-3_14

In this paper, we attempt to combine graph attention mechanism with auxiliary side information simultaneously for a better trade-off. We propose an approach called D2NE to learn the vertex representations for dynamic networks. We would like to address two challenges highlighted earlier. It exploits the graph attention mechanism to refresh embeddings for dynamic networks efficiently, models the auxiliary side information to handle the missing data. In summary, our significant contributions are as follows:

- We propose a tailored method to learn vertex representations on dynamic networks. The approach is a unified one that integrates the auxiliary side information modeling to capture sufficient information on vertex relations and employs the graph attention mechanism to refresh embeddings for dynamic networks efficiently.
- We illustrate the performance of our algorithm against comparable baselines on several real-world dynamic networks. Empirical study manifests that D2NE outperforms baselines in network embedding for dynamic networks.

The remainder of paper is organized as follows. We shortly discuss the related work in Sect. 2. We formally define the problem in Sect. 3, before delving into details of the proposed dynamic network embedding method in Sect. 4 and report our empirical study in Sect. 5. Finally, we conclude this paper in Sect. 6.

2 Related Work

Existing works have primarily focused on embedding static networks. However, the networks are often dynamic, which means the vertices and relations are constantly changing over time. For example, the joining of new users along with new friendships in social networks will change the network in the next timestamp. The time-series information plays a significant role in dynamic networks [1]. Motivated by recent literature about dynamic network embedding, we deem the dynamic network embedding methods can be categorized into three types: Matrix Factorization (MF)-based, Random-walk-based and Deep-graph-based methods.

The MF-based methods aim to learn the vertex representations by performing eigen-decomposition on the adjacency matrix and attribute matrix [2]. DANE [3] is a typical MF-based method to update the vertex representations dynamically while retaining high-order proximity. Inspired by DANE, Zhu et al. [4] propose an algorithm to incorporate the changes in dynamic networks. Whereas, MF-based methods have the main drawback: they are usually computationally expensive due to the eigen-decomposition operations on data matrices [5]. The work of word2vec [6] inspires many works to learn vertex representations based on the Skip-gram model. Du et al. [7] extend the dynamic network embedding framework for Random-walk-based methods and present a decomposable objective that can learn the representation of each vertex separately. The key idea of Deep-graph-based methods is to extract the spatial features of topological structure, then embed them into a low dimensional space iteratively. Kipf et al. [8]

first propose GCN which is a feature extractor for the graph. Hamilton et al. [9] attempt to address the problem of dynamic network embedding. They propose GraphSAGE-GCN to obtain the tremendous convergence rate.

Similar to the work of GraphSAGE-GCN, Zhang et al. [10] employed the graph attention mechanism to learn the vertex representations for static networks only. Then, Vaswani et al. [11] proposed the transformer model which contained the multi-head attention mechanism. It utilized scaled dot-product attention to make model focus on different types of information. DyRep [12] indicates the association and communication of dynamic network embedding by attention mechanism. It regards the change of vertex representations as the mediator between the two processes, which updates the representations according to new events. However, it ignores the edge types and vertices attributes.

It is worthing to point out these methods have two limitations: (1) they are usually computationally expensive for refreshing embedding in dynamic networks; (2) their performance is rather sensitive to the data quality which makes it difficult to handle the real-world networks. In this paper, we focus on the aforementioned limitations of the existing network embedding methods.

3 Dynamic Network Embedding Approach

In this section, before we overview our proposed approach, we describe a formal definition of the dynamic network embedding problem.

3.1 The Problem Definition

Let $G = (V, E)$ be a dynamic network, where V and E denote the set of vertices and edges respectively over a temporal window $[1, T]$. An evolution of G can be denoted as $\mathcal{G} = \{G_1, \cdots, G_T\}$, where G_t represents a snapshot of $\mathcal{G}$ at time t. v_i^t and v_j^t denote the i-th and j-th vertex in V at time t respectively, where i, j = 1, 2, ... , $|V|$. Each edge carries a non-negative weight w_{ij}^t, describing the strength between the connected vertices v_i^t and v_j^t. Therefore, we can use a matrix $\mathbf{W} = [\boldsymbol{w_{ij}^t}]$ to represent all weights in the dynamic network. Let $\alpha_m(v_j^t)$ represent the observed features of $v_j^t \in V$, i.e., $\alpha_m(v_j^t)$ represents the observed feature vector of v_j^t from V at time t.

Dynamic network embedding aims to map all vertices in the network into a low-dimensional embedding space where each vertex is represented as a dense embedding vector and update the vertex representations over time.

To keep the notations simple, we use $\boldsymbol{v_n^t}$ to denote the embedding vectors for vector v_n^t. As such, all the vertices in the dynamic network can be denoted as a matrix $\mathbf{V_t} = [\boldsymbol{v_n^t}]$ at time t.

3.2 Overview of D2NE

Our proposed dynamic network embedding approach consists of three components as follows:

Step 1: Dual events modeling. For better characterizing dynamic process, we similarly resort to performing a dual events mechanism by judging that there is an eternal edge generated. A event $Eve = (v_i, v_j, k, t)$ covers communication process ($k = 1$) and association process ($k = 0$) respectively. In the former, a temporary edge will exist between two vertices that are not connected. The latter describes the changes in the topological structure of networks. Hence, all the events will be classified into two types.

Step 2: Vertex embeddings refreshing. Inspired by pioneering works [12] and [10], we design two attention layers to refresh vertex embeddings. One is the Temporal Attention Layer (TAL), which focuses on the topological structure of networks. The other one is the Graph Attention Layer (GAL), which handles the vertex feature by neglecting the structural information. Firstly, we obtain the output representation vectors at time t, and then integrate the localized embedding propagation to achieve the current vector embeddings at time $t + 1$ through TAL. Secondly, we calculate the attention coefficients through based on auxiliary side information (i.e. vertex feature vector $\alpha_m(v_j^t)$) and weight matrix $\boldsymbol{W}$. Finally, we obtain the joint embedding vectors after multi-head attention through GAL.

Step 3: Joint formulation. After an event occurs, we can derive the embedding vectors from joint formulation. Specifically, we combine the intermediate embedding vectors from TAL and GAL to form a joint formulation: $\boldsymbol{v_n^t} = \frac{\alpha}{\alpha+\beta}\boldsymbol{z}_{v_i}(t) + \frac{\beta}{\alpha+\beta}\boldsymbol{r}_{v_i}(t)$, where parameters α and β will be learned to select an appropriate proportion to combine the topological structure and vertex feature.

Unlike the earlier DyRep [12], D2NE considers both topological structure and vertex feature to learn the vertex representations for dynamic networks. It is a significant extension to perform the dynamic network embedding task efficiently and effectively.

4 Modeling and Joint Formulation

We employ dual events and attention mechanisms to solve the problem. When an event occurs, the topological structure and vertex feature may change. The key idea of D2NE is to build a unified architecture that utilizes the vertex feature and topological structure to model the evolution through TAL and GAL jointly.

4.1 Dual Events Modeling

The occurrence of $Eve = (v_i, v_j, k, t)$ on the dynamic network is complex. It can be classified into two types of events, one is communication and the other is association. Both of the occurrences are related to the most recent state $\bar{t}$. We model the occurrence of the event by using the conditional intensity function at time t: $\lambda_k^{v_i,v_j}(t) = \psi_k log(1 + exp(\omega_k^T \cdot [z_{v_i}(\bar{t}) \| z_{v_j}(\bar{t})]/\psi_k))$, where $z_{v_i}(\bar{t}) \| z_{v_j}(\bar{t})$ means the concatenation of the most recently updated representation, ω_k^T serves as the model parameter that learns time-scale specific compatibility and ψ_k corresponds to the rate of events arising from a corresponding process.

4.2 Utilizing Topological Structure in TAL

Specifically for the timestamp of current event t_p, the *p-th* event of vertex v_i can formulate the evolution as: $\lambda_k^{v_i,v_j}(t) = \psi_k log(1 + exp(\omega_k^T \cdot [z_{v_i}(\bar{t}) \| z_{v_j}(\bar{t})]/\psi_k))$, where $\bar{t_p}$ signifies the timestamp just before current event, $\boldsymbol{h}_{struct}^{v_j}$ is the output embedding vectors from aggregator function of neighbors. $\boldsymbol{M}_{struct}$, $\boldsymbol{M}_{rec}$ and $\boldsymbol{M}_t$ are three parameters in a neural network. Hence, the form of $\boldsymbol{M}_{struct}\boldsymbol{h}_{struct}^{v_j}(\bar{t_p})$ means the localized embedding propagation. Through the introduction of $\boldsymbol{z}_{v_i}(t_p^{\bar{v_i}})$ which means the recurrent state from the previous representation of vertex v_i, we achieve the form of self-propagation like $\boldsymbol{M}_{rec}\boldsymbol{z}_{v_i}(t_p^{\bar{v_i}})$. At last, the $\boldsymbol{M}_t(t_p - \bar{t}_p^{v_i})$ is the exogenous drive. The temporal point process self-attention which illustrates the computation of $\boldsymbol{h}_{struct}^{v_j}(\bar{t_p})$ by aggregator function: $\boldsymbol{h}_{struct}^{v_j}(\bar{t_p}) = max(\{\sigma(\frac{exp(S_{v_j k}(\bar{t_p}))}{\sum_{k' \in N_{v_j}(\bar{t_p})} exp(S_{v_j k'}(\bar{t_p}))} * (\boldsymbol{M}_h \boldsymbol{z}_k(\bar{t_p}) + b_h))\})$, where k is the neighbor of v_j. The $\frac{exp(S_{v_j k}(\bar{t_p}))}{\sum_{k' \in N_{v_j}(\bar{t_p})} exp(S_{v_j k'}(\bar{t_p}))}$ signifies the attention weight. $\boldsymbol{M}_h$ and b_h are parameters govern the information propagated by each neighbour of v_j.

4.3 Combining Vertex Feature in GAL

In the time t, the input of GAL is a series of vertex feature $\alpha_m(v_j^t)$. Through the self-attention on vertices, a shared attention mechanism a computes attention coefficients between vertices v_i, v_j over time: $e_{v_i v_j}(t) = a(\boldsymbol{W}\alpha(v_i^t), \boldsymbol{W}\alpha(v_j^t))$, where weight matrix $\boldsymbol{W} = [\boldsymbol{w_{ij}^t}]$ is the parameter to initialize the shared linear transformation. We employ softmax function and LeakyReLU nonlinearity across different vertices: $\gamma_{v_i v_j}(t) = \frac{exp(LeakyReLU(\boldsymbol{a}^T[\boldsymbol{W}\,\alpha(v_i^t) \| \boldsymbol{W}\,\alpha(v_j^t)]))}{\sum_{k \in N_{v_i}(t)} exp(LeakyReLU(\boldsymbol{a}^T[\boldsymbol{W}\,\alpha(v_i^t) \| \boldsymbol{W}\,\alpha(v_k^t)]))}$, where $\|$ represents the concatenation operation. Then we perform multi-head attention on this network: $\boldsymbol{r}_{v_i}(t) = \sigma(\frac{1}{Q}\sum_{q=1}^{Q}\sum_{k' \in N_{v_i}} \gamma_{v_i v_{k'}}^q(t)\boldsymbol{W}^q\alpha(v_{k'}^t))$, where $\boldsymbol{r}_{v_i}(t)$ is the output of GAL means the representation of vertex v_i at time t, Q means the number of heads, and $k' \in N_{v_i}$ is the neighbor of v_i.

4.4 Joint Formulation

After $Eve = (v_i, v_j, k, t)$ occurs, representations of v_i and v_j are changed. It is pointing out that $\boldsymbol{z}_{v_i}(t)$ and $\boldsymbol{z}_{v_i}(t)$ are intermediate embedding vectors derived from TAL and GAL respectively. First, we maximize the cosine similarity: $maximize \quad cos(\theta) = \frac{\sum_{i=1}^{n} \frac{\alpha}{\alpha+\beta}\boldsymbol{z}_{v_i}(t) \times \frac{\beta}{\alpha+\beta}\boldsymbol{r}_{v_i}(t)}{\sqrt{\sum_{i=1}^{n}(\frac{\alpha}{\alpha+\beta}\boldsymbol{z}_{v_i}(t))^2} \times \sqrt{\sum_{i=1}^{n}(\frac{\beta}{\alpha+\beta}\boldsymbol{r}_{v_i}(t))^2}}$. Then obtain the ultimate joint embedding vector $\boldsymbol{v_n^t}$: $\boldsymbol{v_n^t} = \frac{\alpha}{\alpha+\beta}\boldsymbol{z}_{v_i}(t) + \frac{\beta}{\alpha+\beta}\boldsymbol{r}_{v_i}(t)$, which can refresh vertex embedding by both topological structure and vertex feature at time t.

5 Empirical Study

To evaluate the vertex embeddings learned by D2NE, we employ them to address the link prediction task of dynamic network. Dynamic link prediction is usually used to predict the presence or absence of edges in the future time by dynamic embeddings. We aim to answer the following research questions. **RQ1:** How does D2NE perform compared with state-of-the-art dynamic network embedding methods in dynamic link prediction task? **RQ2:** Can our proposed D2NE refresh embeddings efficiently? In what follows, we first introduce the experimental settings and then answer the above research questions in turn to demonstrate the rationality of D2NE.

5.1 Experimental Settings

Datasets. The *Social Evolution* is publicly accessible[1], which contains a small social network with 83 vertices and a high clustering coefficient over 2M events. Besides, we extract the “political opinions”, “attitudes towards exercise and fitness”, “current confidence and anxiety level”, and “music sharing” as auxiliary side information to help the embedding. The *Github*[2] contains a large network with 12,328 vertices with low clustering coefficient and sparse events including “signing in”, “following”, “project browsing”, and “coding”.

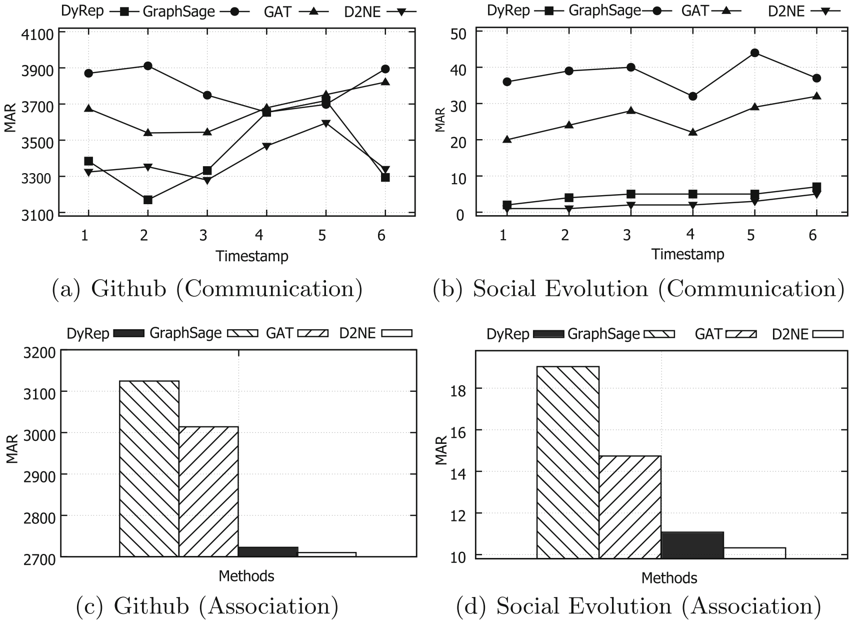
Baselines and Evaluation Metrics. **GraphSage** [9] is an unsupervised inductive method that can quickly generate embedding for the new vertex without additional training. **GAT** [10] is a novel supervised method which only use vertex feature to generate embeddings with multi-head attention in the static network. **DyRep** [12] is an unsupervised method which focuses on the dynamic network with the temporal point process. We choose a vertex v_i and replace vertex v_j with other entities in the network at time t. Then compute the density by dynamic k: $\int_k^{v_i,v_j}(t) = \lambda_k^{v_i,v_j}(t) \cdot exp(\int_{\bar{t}}^{t} \lambda(s)ds)$. Where $\bar{t}$ is the time of the most recent event on either dimension v_i or v_j. After that, we rank all the entities in descending order of the density and report the rank of ground truth. We evaluate our method by Mean Average Rank (MAR) for link prediction task on dynamic network. To demonstrate the fine trade-off between effect and efficiency of our proposed approach, we also measure the elapsed time in the second experiment.

5.2 Performance Comparison (RQ1)

We perform D2NE and baselines in link prediction task for dynamic networks in communication and association respectively. Figures 1(a)–(d) illustrate the performance, where we have the following key observations: (1) Not only in communication but also in association, D2NE outperforms the baselines significantly and achieve the best performance on both datasets. This is due to the factors

[1] http://realitycommons.media.mit.edu/socialevolution1.html.

[2] http://www.gharchive.org/.



(a) Github (Communication)

(b) Social Evolution (Communication)

(c) Github (Association)

(d) Social Evolution (Association)

Fig. 1. Link prediction performance on Social Evolution and Github

that only D2NE combines auxiliary side information and topological structure to learn vertex representations; (2) DyRep is significantly better than GraphSage and GAT because they are not tailored for learning on dynamic networks. It shows that the traditional static network embedding methods could not suitable for dynamic network embedding tasks through refreshing embeddings.

5.3 Efficiency of D2NE (RQ2)

As shown in Fig. 2, we compare the inference time of baselines and D2NE on a timestamp after training. Due to the combination of network topology and vertex feature, the inference time of our algorithm is slightly higher than DyRep

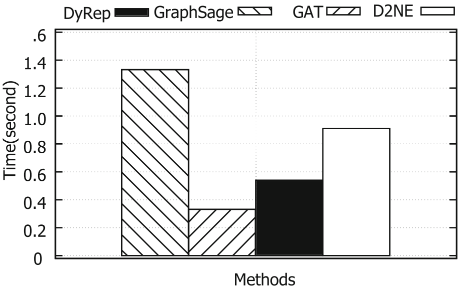


Fig. 2. Elapsed time

and GAT. However, it is still faster than GraphSage, which manifests a fair trade-off between effectiveness and efficiency.

6 Conclusion

In this paper, we have studied the problem of dynamic network embedding. We proposed a Deep-graph-based method to deal with the mentioned challenges. The D2NE introduces the dual events modeling and utilizes the topological structure with TAL then combines the vertex feature with GAL. We have illustrated our proposed method on several real-world networks. Experimental results indicate that D2NE not only outperforms the comparable baselines but also obtains the promising performance.

Acknowledgments. This work was supported by the National Natural Science Foundation of China Youth Fund under Grant No. 61902001 and Initial Scientific Research Fund of Introduced Talents in Anhui Polytechnic University under Grant No. 2017YQQ015.

References

1. Manessi, F., Rozza, A., Manzo, M.: Dynamic graph convolutional networks. Pattern Recogn. **97**, 107000 (2020)
2. Nedic, A., Ozdaglar, A.E.: A geometric framework for nonconvex optimization duality using augmented Lagrangian functions. J. Global Optim. **40**(4), 545–573 (2008)
3. Li, J., et al.: Attributed network embedding for learning in a dynamic environment. In: CIKM 2017, Singapore, pp. 387–396 (2017)
4. Zhu, D., et al.: High-order proximity preserved embedding for dynamic networks. IEEE Trans. Knowl. Data Eng. **30**(11), 2134–2144 (2018)
5. Wang, M., et al.: Learning on big graph: label inference and regularization with anchor hierarchy. IEEE Trans. Knowl. Data Eng. **29**(5), 1101–1114 (2017)
6. Mikolov, T., et al.: Efficient estimation of word representations in vector space. In: Workshop Track Proceedings, ICLR 2013, Scottsdale, Arizona, USA (2013)
7. Du, L., et al.: Dynamic network embedding : an extended approach for skip-gram based network embedding. In: IJCAI 2018, 2018, Stockholm, Sweden, pp. 2086–2092 (2018)
8. Bruna, J., et al.: Spectral networks and locally connected networks on graphs. In: Conference Track Proceedings, ICLR 2014, Banff, AB, Canada (2014)
9. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: NIPS 2017, Long Beach, CA, USA, pp. 1024–1034 (2017)
10. Velickovic, P., et al.: Graph attention networks. In: Conference Track Proceedings, ICLR 2018, Vancouver, BC, Canada (2018)
11. Vaswani, A., et al.: Attention is all you need. In: IPS 2017, Long Beach, CA, USA, pp. 5998–6008 (2017)
12. Trivedi, R., Farajtabar, M., Biswal, P., Zha, H.: DyRep: Learning representations over dynamic graphs. In: ICLR 2019, LA, USA, New Orleans (2019)

Elaborating the Bayesian Priors in Unsupervised Graph Embedding via Graph Concepts

Xiaojun Ma[1], Ziyao Li[2], Siwei Wei[3], and Guojie Song[1(✉)]

[1] Key Laboratory of Machine Perception, Ministry of Education, Peking University, Beijing, China
{mxj,gjsong}@pku.edu.cn
[2] Center for Data Science, Peking University, Beijing, China
leeeezy@pku.edu.cn
[3] School of Electronics Engineering and Computer Science, Peking University, Beijing, China
weisiwei4@pku.edu.cn

Abstract. Unsupervised Graph Embedding yields specific importance because it performs well with inputs limited to the graph structure only. Proximity-preserving models, including link-preserving and Skip-Gram models, prove to be good approaches in both efficiency and accuracy on unsupervised tasks, even compared with state-of-the-art deep models. We first show that the optimization problem these models solve is equivalent to a Bayesian Inference problem, however, these models generally assume a uniform distribution for the target node representations, that is, the representations of nodes are not further constrained. In our paper, we elaborate this Bayesian prior resorting to potential concepts underlying a graph. These graph concepts can be communities in a graph, nodes with different interaction patterns et al. We further derive the optimization objective according to this elaborated prior, and proposed our learning objective. Intuitively, graph nodes of the same concept are embedded close to each other. Our paper proposes a flexible framework which is adaptable to any other proximity-based models. Experiments show that our model significantly elevates the baseline performances of proximity-preserving models, yielding state-of-the-art results on unsupervised learning tasks.

1 Introduction

Graph Embedding (GE) is a task to derive low-dimensional vector representations for graph nodes, generally following an unsupervised setup. The motivation behind is to use dimensionality reduction techniques to compress the sparse, high-dimensional information in the graph adjacency. Traditional unsupervised

X. Ma and Z. Li—Equal Contribution.

This work was supported by the National Natural Science Foundation of China (Grant No. 61876006).

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 176–190, 2020.
https://doi.org/10.1007/978-3-030-65390-3_15

Graph Embedding models, including Deepwalk [15], LINE [17] and their modifications [5,13,19,23] generally follow a proximity-preserving style, that is, the dot product of node representations are constrained to preserve the *proximity* between graph nodes. Definitions of such *proximity* differs among models, but generally, the graph links [17] or the target-context pairs generated from random walk on graphs [15] are used to define the proximity. Accordingly, we name them as link-preserving models and Skip-Gram models in our paper.

These proximity-preserving models yield specific importance because they work well even in the worst cases – when neither node features nor supervisions are unavailable. Meanwhile, although recently there is a prevalent trend of leveraging deep learning tools on relational data [6,10,20,22], these models are mainly proposed for supervised tasks with available node features. When it comes to unsupervised learning, most deep models still refer to proximity-preserving models [6] or Auto-encoders [9,24] to define their optimization objective. In fact, experiments show that these highly-parameterized deep models do not show significant advantage to Deepwalk in unsupervised graph embedding tasks.[1] Therefore, in order to develop better unsupervised graph models, it is still of great importance to take a step backwards to analyze and modify traditional proximity-preserving models.

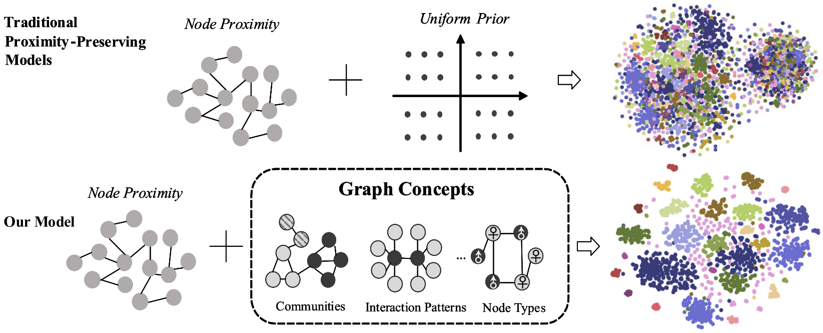


Fig. 1. A general description of the major idea in this paper. Instead of assuming a uniform prior for node representations, our model resorts to graph concepts to constrain the node representations. Nodes of common concepts are embedded closer to each other. Visualization results are obtained from `cora` dataset.

In this paper, we first look into the basic assumptions of these proximity-preserving models under a Bayesian Inference framework. We found that these methods give a reasonable prior to model how graph proximities are generated, but they assign a uniform prior to the distribution of node representations. That is, these representations are not further restricted, and remain independent between graph nodes. However, such independence is unlikely to hold in

[1] See Sect. 4. The same phenomenon is observed and noted in the appendix of GraphSAGE [6].

real-world relational data, where entities would share common or different *concepts* at graph level. Taking social networks as an example, these concepts could be different in-born node types (e.g.. people with different sex, ages or occupations), different clusters in graphs (e.g.. communities or social groups), or different interaction patterns (e.g.. people with different social degrees or clustering coefficients). Assuming a uniform distribution of the node representations in fact ignores these commonalities, which ought to be important information for capturing the manifolds the graph nodes lie on. Therefore, we further elaborate the prior of node representations with these potential graph concepts, and design the optimization objective as to maximize the conditional likelihood of the representations given a graph. Figure 1 shows the major difference between our model and traditional proximity-preserving models.

As is introduced in [16], most Skip-Gram and link-preserving models can be equivalently transformed to implicit matrix decomposition problems, where a matrix denoting node proximity (varies across different models) is factorized into two embedding matrices. As a cross-entropy loss is generally used in these models, solving the matrix factorization is equivalent to solving a Bayesian Inference problem: links (in link-preserving models) or node co-occurrences (in Skip-Gram models) serves as observations of a Bernoulli distribution, which is depicted by the strength of the underlying associations between two nodes. Generally, the association strengths between nodes are modeled with the dot product of node representations. Denoted in formula,

$$\begin{aligned} P(G=(V,E)|\mathbf{U}) = \prod_{i,j \in V} [&\mathbf{I}((i,j) \in E)\sigma(u_i^T u_j) \\ &+ \mathbf{I}((i,j) \notin E)\sigma(-u_i^T u_j)]. \end{aligned} \tag{1}$$

where $\mathbf{U} = (u_1, \cdots u_{|V|})$ is the matrix containing target representations, $\sigma(\cdot)$ is the *sigmoid* function, and E denotes the set of graph links or target-context pairs.[2]

The negative logarithmic probability of $P(G|\mathbf{U})$ is directly minimized in traditional proximity-preserving models, which is equivalent to optimize the true posterior probability $P(\mathbf{U}|G)$ according to the Bayesian formula

$$P(\mathbf{U}|G) \propto P(G|\mathbf{U})P(\mathbf{U}), \tag{2}$$

with $P(\mathbf{U})$ modeled with a uniform prior, i.e., the embedding vector u_is are not further restricted. In our paper, we elaborate the underlying uniform assumption over $P(\mathbf{U})$ resorting to graph concept, and accordingly propose a new learning objective for unsupervised Graph Embedding tasks: beside assuming proximities between nodes are observations from a Bernoulli's distribution, we also assume that node representations are observations of the concept they belong to, and

[2] While Skip-Gram models including Deepwalk and node2vec [5] optimize a target embedding and a context embedding for each node, for convenience, we use one matrix $\mathbf{U}$ to denote node embedding vectors following the first-order setup of [17]. Implementing $\mathbf{U}$ with two different matrics does not influence the overall framework of our paper.

each concept is depicted with a Gaussian distribution prior. We then maximize the new posterior probability $P(\mathbf{U}|G)$ under our own setup.

On the one hand, the new optimization objective better captures underlying concepts of a given graph, with μ_c depicting the concept's center and Σ_c its variances. On the other hand, it encourages representations of nodes under the same concepts to converge together through maximizing $P(\mathbf{U})$. This coincides with the intuition that similar nodes shall have similar representations. Besides, another by-product of this model is that it generates a likelihood for every learned representation. These likelihoods show the *confidence* of the learned representations, and are potentially useful in downstream applications, such as i) abnormality detection and ii) sample weights in downstream supervised learning tasks.

To summarize, the characteristics of our proposed model lies in four aspects. **i)** We elaborate the basic assumption of Skip-Gram and link-preserving models through resorting to underlying *concepts* in a graph. **ii)** Our proposed learning objective can be instinctively associated with any proximity-preserving models besides Deepwalk, LINE, node2vec etc., and thus our model enjoys the same efficiency of these models, which are generally known as fast algorithms. **iii)** The new model generates features and variances for every pre-assumed, deterministic graph concepts via encouraging the representation of nodes under the same concept to converge together. **iv)** The new model generates a likelihood for representations of each node, which is potentially useful in various downstream applications such as credit evaluation or abnormality detection.

2 Related Work

2.1 Unsupervised Graph Embedding

The motivation behind most unsupervised Graph Embedding models is to preserve the proximities between graph nodes. These proximity-preserving models can be divided into two types, namely link-preserving models [13,17] and Skip-Gram based models [5,15], where the major difference in between is whether links or the target-context pairs obtained via random walks are preserved. After [11] revealed a connection between Skip-Gram models and implicit matrix factorization problems, [16] unified most proximity-preserving graph embedding models under the matrix factorization framework. In our paper, we interpret the matrix factorization problem as a Bayesian inference problem, and derive the basic assumptions of these models. These assumptions are further elaborated to derive our own optimization goal.

2.2 Graph Neural Networks

Graph Neural Networks (GNNs) [3,6,7,10] apply deep learning tools on relational data. As GNNs are gradually becoming a most heated topic in the field of relational data modeling, countless new architectures are proposed in the recent two years [8,12,18,20]. However, most of these works focus on supervised learning tasks on graphs, and optimization goals designed for these deep models remains preserving node proximity or Auto-encoders. However, as [6] stressed in its appendix, Deepwalk can indeed outperform state-of-the-art deep architectures on transductive, unsupervised learning tasks. To further elevate the performance of GNNs on unsupervised learning tasks, we abstract different GNN models as node encoders and studies the optimization goals they share.

2.3 Uncertainty Learning on Graphs

As an important by-product of our model is the uncertainty of graph concepts and node-wise likelihoods, we also compare our model with several uncertainty learning approaches on graphs. Graph2Gauss [2] embeds graph nodes as individual Gaussian distributions using unsupervised personalized ranking formulation. HIB [14] models node representations as random variables and the model is trained using variational information bottleneck principle. DVNE [25] learns a Gaussian distribution in the Wasserstein space of each node which models the uncertainty of nodes. However, the *uncertainty* these model try to capture are all node-wisely defined, which may somehow be controversial: for each concept these model learn, only one observation can be seen, and thus the meaning of the estimated uncertainty is actually ambiguous. To stress this issue, we resort to pre-defined underlying concepts in graph and regard the nodes they contain as multiple samples. We believe this is a more founded way of capturing graph uncertainties.

3 Model

3.1 Notations and Problem Definition

In this paper, we focus on a Bayesian inference problem of deriving optimized node representations. Given a graph $G = (V, E)$, the problem is to find an optimum hypothesis of the node embedding matrix $\mathbf{U} = \{u_1, u_2, \cdots, u_{|V|}\}$ with $u_i \in \mathbf{R}^d$. The posterior probability we want to maximize is the same as traditional proximity-preserving models, which is shown in Formula (2). Besides, we also assume that a set of graph concepts $\mathcal{C} = \{c_1, c_2, \cdots, c_{|\mathcal{C}|}\}$ is available, along with the belonging relationship between nodes and concepts. Although concepts as ground truth are sometimes hard to obtain, these concepts can be generated through different unsupervised methods. For example, graph partitioning methods [1] generates community concepts at graph level; dividing the graph nodes according to degrees or clustering coefficients generates concepts with regard to node interaction patterns.

The prior probability of the embedding matrix $P(\mathbf{U})$ is then modeled with these concepts. By assigning a unique Gaussian distribution for every concept c, and regarding every representation of a node in a concept as an observation from the distribution, we have a prior probability $P(\mathbf{U})$ as

$$P(\mathbf{U}) = \prod_{i \in V} \sum_{c \in \mathcal{C}} \mathbf{I}(i \in c) \phi(u_i; \mu_c, \Sigma_c), \tag{3}$$

$$\phi(u_i; \mu_c, \Sigma_c) = \frac{\exp\left\{-\frac{1}{2}(u_i - \mu_c)^T \Sigma_c^{-1}(u_i - \mu_c)\right\}}{\sqrt{(2\pi)^d \det(\Sigma_c)}}. \tag{4}$$

3.2 Loss and Optimization

Combined with the conditional probability $P(G|\mathbf{U})$ defined in Formula (1), the general loss of our proposed optimization goal is defined as the negative logarithmic posterior probability, i.e.

$$\mathcal{L}(\mathbf{U}; G) = -\log(P(G|\mathbf{U})) - \log(P(\mathbf{U})) \tag{5}$$

$$= O_1 + O_2 + C \tag{6}$$

where O_1 is the loss of preserving the node proximity, O_2 is the loss of conforming the concept distribution, and C is a constant value. O_1, O_2 takes following form:

$$O_1 = -\sum_{(i,j)\in E} \log\sigma(u_i^T u_j) + \sum_{(i,j)\notin E} \log\sigma(-u_i^T u_j), \tag{7}$$

$$O_2 = \frac{1}{2}\sum_{i\in V}\sum_{c\in\mathcal{C}} \mathbf{I}(i \in c)\cdot \tag{8}$$

$$[(u_i - \mu_c)^T \Sigma_c (u_i - \mu_c) + \log\det(\Sigma_c)]. \tag{9}$$

The same as traditional proximity-preserving models, we estimate objective O_1 through a negative sampling strategy: for each node in a positive proximity pair, we randomly sample a given number of negative contexts following the distribution of $p(k) \propto \deg(k)^{0.75}$, where $\deg(k)$ denotes the degree of a node k[3], i.e.

$$O_1' = \sum_{(i,j)\in E} -\log\sigma(u_i^T u_j) + N\mathbb{E}_{k\sim p(k)} \log\sigma(-u_i^T u_k) \tag{10}$$

where N is the number of negative samples.

Optimizing the entire co-variance matrices Σ_cs for all concepts in $\mathcal{C}$ could be a challenge. However, as it is plain to prove that O_1 enjoys an orthogonal invariance to all $\mathbf{U}$, we can expect that the learned node representations and concept distributions are already rotated and lies in a group of standard orthogonal basis. Accordingly, Σ_cs become diagonal matrices, denoted as

$$\Sigma_c = \begin{pmatrix} \sigma_{c,1}^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{c,d}^2 \end{pmatrix},$$

and the number of parameters to be estimated reduces from d^2 to d. The objective O_2 then becomes

[3] Set the same as [17] and [6].

$$O_2' = \frac{1}{2}\sum_{i\in V}\sum_{c\in\mathcal{C}} \mathbf{I}(i\in c)\cdot\left[\|(u_i-\mu_c)\odot(\sigma_c^{-2})\|_2^2 + \sum_{1\le t\le d}\log\sigma_{c,t}^2\right]$$

where $\odot$ is the element-wise product operation and $\sigma_c^{-2} = \left(\sigma_{c,1}^{-2},\cdots,\sigma_{c,d}^{-2}\right)^T$.

As O_1' is the likelihood of a discrete distribution while O_2' of a continuous one, we control the two objective with a hyper-parameter weight α to ensure that they are of the same scale.[4] The final objective we optimized in our model is defined as

$$\mathcal{L} = O_1' + \frac{\alpha}{d}O_2'. \tag{11}$$

3.3 Learning the Variances

While learning $\sigma_{c,t}$ as free parameters seems good enough, as $\sigma_{c,t}$ levels off to 0, the gradient moves towards $+\infty$, and the training process suffers from great instability. To stress this problem, we firstly explicitly derive the optimum solution of $\sigma_{c,t}$s. Note that in Equation (??), the optimum solution of $\sigma_{c,t}$ is

$$\arg\min_{\sigma_{c,t}^2} = \frac{1}{|c|}\sum_{i\in c}(u_{i,t}-\mu_{c,t})^2, \tag{12}$$

where $u_{i,t},\mu_{c,t}$ denotes the t-th element in the corresponding vector. Therefore, instead of learning free parameters for $\sigma_{c,t}$, we simulate these variables by sampling a fixed number of nodes under the concept c, i.e.

$$\sigma_{c,t}^2 = \mathbb{E}_{v\sim p(v)}(u_{v,t}-\mu_{c,t})^2 \tag{13}$$

$$p(v) = \begin{cases} 0 & v\notin c \\ \frac{1}{|c|} & v\in c \end{cases} \tag{14}$$

3.4 Learning with Deep Models

To combine the proposed optimization goal with neural networks, we treat the given neural architecture as a node encoder, i.e. a function from a graph node to its representation:

$$u_i := u(i;G) = GNN(i). \tag{15}$$

Therefore, the restrictions derived from both O_1' and O_2' are applied to the output vector of the neural network. In addition, we do not input constant node features to these neural networks for a fair competition between deep and traditional models. As we will introduce in Sect. 4, when these input features are unavailable, deep models do not show advantage to simple models in unsupervised tasks. This is probably because the objective of unsupervised tasks is not enough for these deep models to learn the aggregation patterns in local neighborhoods.

[4] Empirically, we divide O_2' by the representation dimension d because O_2' is calculated through summing over all the elements in the representation vectors.

4 Experiment

In this section, we demonstrate the effect of our proposed learning objective via experiments over two kinds of real-world datasets. We first introduce the datasets and baselines we use in this section. The performance of the unsupervisedly-learned representations from our models and different baselines are then compared via node classification tasks. The performance of the unsupervisedly-learned representations from our models and different baselines are then compared via node classification tasks. Besides, we illustrate how well the graph concepts are captured via visualization experiments on three datasets. The effect of different αs and ds in our model is also discussed.

4.1 Experiment Setup

Table 1. The statistics of the four graph datasets in our experiments.

Dataset	$\|V\|$	$\|E\|$	$\frac{\|E\|}{\|V\|}$	$n\_classes$
`cora`	$2,708$	$5,429$	2.00	7
`citeseer`	$3,312$	$4,732$	1.43	6
`dblp`	14389	111858	7.78	8
`europe-airports`	399	5995	15.0	4
`usa-airports`	1190	13599	11.4	4

Datasets. Three citation networks and two airline networks are used in our experiments. These two airline networks are the same as in [5]. The statistics of these datasets are presented in Table 1. Specifically:

- `cora` is a citation network consisting of $2,708$ Machine Learning papers with each one classified into one of seven topics. A link presents a citation relationship between two papers. Although word features are available in the dataset, we do not use them in our experiments in order to test the models' capability of capturing graph structures.
- `citeseer` is a citation network consisting of $3,312$ Computer Science papers with each one belongs to one of six research fields. A link presents a citation relationship between two papers. Similar to `cora`, we do not use the word embeddings available in the dataset.
- `dblp` is a co-author network constructed with papers from 2013 to 2017 in eight artificial intelligence conferences. It contains 14389 papers. A link presents a citation relationship between two papers. Compared with previous networks, `dblp` is relatively dense.
- `europe-airports` is a airline network consisting of 399 airports. A link presents a commerical flight between two airports. Airports are classified into four classes according to their level of activity, which is measure using total number of landings and takeoffs.

- `usa-airports` is a airline network consisting of 1190 airports. A link presents a commerical flight between two airports. Airports are classified into four classes according to their level of activity, which is measure using total number of people that passed this airport.

Baselines. The baselines against which we evaluate our model are briefly introduced below. These baselines not only include traditional proximity-preserving algorithms (LINE, Deepwalk) and graph neural networks (SDNE, GCN), but also community-preserving algorithms (GNE) because our model resort to deterministic graph concepts including communities. The dimensions of the output representations of all models are controlled as $d = 64$. For Graph Neural Networks (GCN), the number of layers is set as 2, and all hidden dimensions are set as 64. For all community-preserving algorithms, the input community information is obtained through an unsupervised graph partitioning approach [1]. A most suitable learning rate for each model is used for fair competition, and an early-stop strategy is leveraged to output the optimal representations during the training process. All models are trained to their convergence.

- **LINE** [17] is a simple link-preserving model designed for large-scale unsupervised Graph Embedding tasks. It preserves the first- or second-order proximity defined through direct adjacencies between graph nodes.
- **DeepWalk**[5] [15] analogizes random walks on graphs as sentences in languages and uses a Skip-Gram model to preserve the co-occurrences of nodes. A hierarchical *softmax* architecture is used in the original version, while we use the negative sampling strategy to accelerate the optimization process.
- **GCN** [10] is a model based on a variant of convolutional neural networks. It operates directly on graph and encodes the local neighborhood information of a given node via an aggregating process. In our experiment, we treat GCN as a node encoder as described in Sect. 3.4. The same loss function as DeepWalk is applied to the output vector of GCN to build a unsupervised learning model. Also, for a fair competition, we use one-hot encoding as input to GCN.
- **SDNE** [21] is a deep model proposed particularly for unsupervised graph embedding. Node representations are expected to preserve node-wise local structures via an Auto-encoder, and pair-wise proximities are also preserved.
- **GNE** [4] is a hierarchical community-preserving model, in which representations of nodes in a community is restricted on a sphere around the community center. GNE also refer to Skip-Gram models to define its objective.

4.2 Node Classification with Different Graph Concepts

Our Model. We evaluate the objective we derive in combination with other proximity-preserving models, in which **LINE+GC** (Graph Concepts)

[5] We choose Deepwalk as a representative of all Skip-Gram based models including node2vec [5] *et al.*, of which the performance is evaluated analogous to Deepwalk and thus not shown.

and **Deepwalk+GC** are evaluated as representatives. LINE+GC and Deepwalk+GC are implemented merely by adding the graph concept prior loss, O_2', to the corresponding original models. Specifically in LINE+GC, we optimize the objective $\mathcal{L}$ defined in Formula (11) with E defined as the set of graph links, and in Deepwalk+GC, E is the target-context pairs generated with random walks on graphs, with $win\_size = 5$ and $num\_walks = 50$.[6] As we find using identical matrix for both target and context nodes in Deepwalk out-performs using different ones, we modified Deepwalk and Deepwalk+GC accordingly.

In addition, we evaluate the derived objective with deep models as described in Subsect. 3.5, and we use **GCN** as representative of deep encoders. **GCN+GC** is implemented by applying the elaborated loss to the output vector of GCN. The elaborated loss is obtained by adding the graph concept prior loss O_2' to the loss used in DeepWalk.

For citation networks, since papers with the same topic tend to densely interlinked, we choose to use the communities in a graph as the concepts we preserve. A fast community-unfolding algorithm [1] is used to generate the communities, which are the same as the inputs of GNE.

For airline networks, since airports activity is highly correlated to their degree centrality, we use degree as graph concept. We uniformly divide node degree into ten intervals, nodes in the same interval share a concept. In airline networks, airports' activity level are highly dependent on their structural identity, and almost with no regard to their positional similarity. Thus, deep models and community preserving models perform poorly, so we omit them in Table 3.

We conduct node classification experiments on three citation networks and two airline networks. For citation networks, the subject a paper belongs to is predicted. For airline networks, the activity level of an airport is predicted. A multi-label logistic regression model is trained with the node representations derived from baselines and our models. A 10-fold Cross-Validation is conducted, where we separately train the classification model with 1 fold (10%) and 9 folds (90%) of data to simulate situations with either insufficient or abundant labeled data. Table 2 shows the accuracies in the Cross-Validation test (*averaged across the ten folds*) on citation networks. The accuracies on airline networks.[7]

In most of the cases, our model beats strong baselines to a significant degree. This corroborates the significant meaning of elaborating the Bayesian priors. Also, this shows that by using different graph concepts in different kinds of networks, our method can easily adapt to totally different situations. Deep models do not show significant privilege to "shallow" ones in the task of unsupervised learning. As we have introduced, the reason that these strong neural baselines fail to yield ideal results is two fold: i) these baselines are specifically proposed for graphs with node features; ii) the strategy these baselines use to conduct unsupervised tasks is primitive, and the parameters in the elegant neural structures are not well-learned. Meanwhile, the results are unsatisfactory even in SDNE, a baseline specifically proposed for unsupervised graph embedding. We reckon

[6] The same parameters are set for Deepwalk baseline.

[7] *Macro f1s* are not shown, in which a trend similar to *micro f1s* is observed.

Table 2. Results of classification tasks on citation networks using community as graph concept(*micro f1s*).

Model	`cora`		`citeseer`		`dblp`	
% train	10%	90%	10%	90%	10%	90%
LINE	46.6 ± 0.7	53.1 ± 0.5	43.2 ± 1.0	49.2 ± 0.7	32.2 ± 0.2	33.3 ± 0.2
Deepwalk	68.0 ± 0.8	72.6 ± 0.7	45.7 ± 1.3	49.4 ± 1.6	32.5 ± 0.2	32.6 ± 0.3
GCN	66.8 ± 1.1	71.1 ± 1.0	45.7 ± 1.3	50.1 ± 1.4	36.4 ± 1.2	38.0 ± 2.0
SDNE	61.5 ± 1.0	67.9 ± 0.7	40.0 ± 0.8	42.4 ± 1.0	39.1 ± 0.6	41.4 ± 0.6
GNE	62.8 ± 0.1	**77.5 ± 0.4**	**46.4 ± 0.1**	37.6 ± 0.1	42.3 ± 0.1	44.6 ± 0.3
LINE+GC	59.5 ± 1.3	65.6 ± 1.2	43.2 ± 1.0	49.2 ± 0.7	**43.6 ± 0.2**	**45.3 ± 0.4**
Deepwalk+GC	**68.5 ± 1.0**	73.0 ± 0.9	**48.3 ± 0.5**	**52.7 ± 0.4**	**43.5 ± 0.3**	45.0 ± 0.4
GCN+GC	**69.5 ± 0.8**	**75.4 ± 0.9**	46.1 ± 0.9	**51.5 ± 1.3**	**43.6 ± 0.6**	**46.3 ± 0.6**

this is because Auto-encoder frameworks are not suitable in this task with great sparsity exists in the inputs. As for the community-preserving models, GNE uses the same community information available for LINE+GC and Deepwalk+GC, but our model outperforms this baseline specially designed to preserve communities. A possible explanation is that it highly relies on communities as ground-truth, which is commonly unrealistic in most relational data. The noises in the unsupervisedly-learned communities under our setup probably lead to fluctuations in the model. Besides, it assign spheres with identical sizes for communities of different scales, which is somehow unreasonable. In our model, we believe it is the simplicity that leads to representations which are robust to the noises in the communities, or to say other potential graph concepts. Different variances are learned for difference model, which also implicitly fixes the problem of GNE.

Table 3. Results of classification tasks on airline networks using degree centrality as graph concept(*micro f1s*).

Model	`europe-airports`		`usa-airports`	
% train	10%	90%	10%	90%
LINE1st	32.0 ± 1.7	36.3 ± 2.1	43.5 ± 1.5	48.5 ± 1.6
LINE2nd	38.2 ± 1.5	42.6 ± 3.0	49.6 ± 1.6	56.0 ± 1.6
DeepWalk	38.8 ± 1.2	44.4 ± 2.6	48.5 ± 2.1	55.3 ± 2.2
LINE1st+GC	32.9 ± 1.6	37.6 ± 2.5	45.6 ± 1.1	53.2 ± 1.7
LINE2nd+GC	**41.6 ± 1.3**	**51.7 ± 2.7**	**51.4 ± 0.6**	**59.1 ± 0.9**
DeepWalk+GC	**40.8 ± 1.2**	**50.1 ± 2.7**	**50.3 ± 1.2**	**58.5 ± 1.0**

Visualization. To demonstrate the effect of assigning graph concept priors in proximity-preserving models, we visualize the output representations of our model LINE+GC and the corresponding baseline LINE on three datasets. *Nodes in one concept are denoted with the same color.*

The visualization results are available in Fig. 2. Figure 2 (a) and (c) show the visualization results of LINE on `citeseer` and `dblp`, Fig. 2 (b) and (d) show the visualization results of LINE+GC on `citeseer` and `dblp`, and the results on `cora` are shown in Fig. 1. To enhance the visual effect, we merges the community with less than 10 nodes into one big community before training LINE and LINE+GC. This community can be regarded as a concept of *isolated nodes*.

It is clear in these visualization results that the graph concepts are well-captured in LINE+GC. Nodes who share common concepts (communities) are constrained to converge together, forming different concept centers. Indeed, LINE to some degree captures these community concepts: node representations from LINE with same colors tend to be closer. However, this effect is very limited, because there is no explicit constraints in LINE to preserve these concepts.

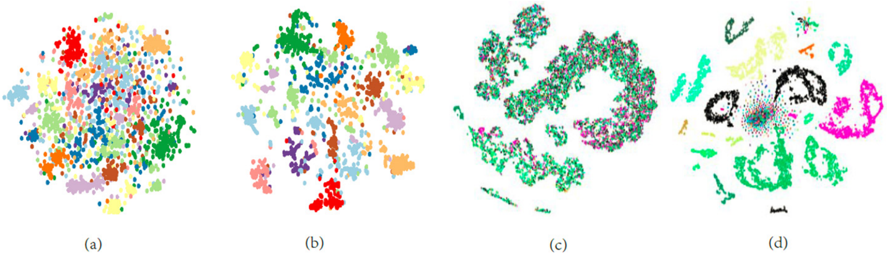


Fig. 2. Visualization results on different datasets. (a) and (b) are from LINE and LINE+GC on `citeseer` dataset; (c) and (d) are on `dblp` dataset. The visualization result on `cora` is shown in Fig. 1. (Color figure online)

Another interesting observation can be observed in the visualization results, that different graph concepts tend to have different variances. This is especially obvious in the `citeseer` dataset considering the concept of *isolated node*. These nodes, shown as blue dots in Fig. 2 (b), form a huge ellipse across the figure. This conforms the intuition that nodes in this *isolated concept* shares an central commonality – isolation – but with a large variance.

4.3 The Effect of α

We also demonstrate the effect of parameters including α – the weight of combining O_1' and O_2' – and d – the dimension of vector in our model. We train models on `cora` with different αs using both LINE+GC and Deepwalk+GC, and evaluate the representations under the same setup as Sect. 4.2. The results are shown in Fig. 3 (a) and (b). When α levels off to 0, LINE+GC and Deepwalk+GC degrade to their corresponding baselines. Meanwhile, when α gets too large, the node representations inside one concept converge to an identical vector, and the predictions are made purely on the concept a node belongs to. Accordingly, the

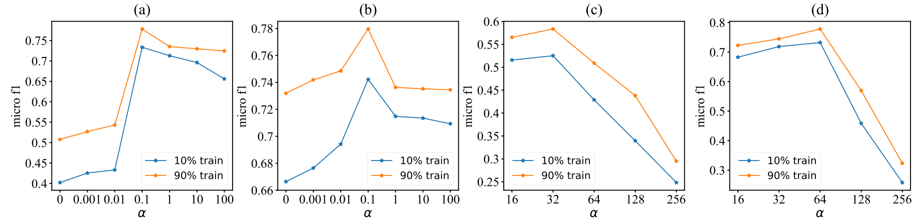


Fig. 3. The influences of parameters in our model, evaluated on `cora` dataset. (a) Micro f1s under different αs in LINE+GC. (b) Micro f1s under different αs in Deepwalk+GC. (c) Micro f1s under different ds in LINE. (d) Micro f1s under different ds in LINE+GC.

performances drop. Empirically, αs between $(0.01, 1)$ would generally be appropriate choices.

A evaluation with different ds for LINE and LINE+GC on `cora` is also conducted. The results are shown in Fig. 3 (c) and (d). When evaluating the performances of different ds, instead of training the model with millions of samples [17], we limits the number of training samples in proportion to the data scale: 200 epochs of data are trained for each d. Under this setup, a higher dimension of the node representations does not equal to a better performance. In fact, the performance start to drop at $d = 32$ for LINE. As empirically, our model LINE+GC converges a lot faster than LINE, it is able to generate effective representations with higher dimensions.

5 Conclusion and Future Work

5.1 Conclusion

In our paper, we first show that the optimization problems in all proximity-preserving models can be equivalently transformed to a Bayesian Inference problem, while a uniform prior is generally assumed in these models. We then elaborate this prior using underlying node concepts of a graph. Based on the graph concept prior, we derive the maximum likelihood loss for the probabilistic model.

Experiments show that merely by adjusting this prior, a 33% elevation on performance can be achieved for the classic link-preserving model LINE. Visualization results show that graph concepts with different variances are well captured in our model. Furthermore, as our model only modifies the prior, i.e. the final loss optimized in these proximity-preserving models, our method enjoys the same efficiency as these models, which are well-known as fast algorithms.

5.2 Future Work

Firstly, instead of combined with proximity-based models, our proposed objective can also be adapted to all kinds of machine learning algorithms on graphs by regarding the algorithms as different types of *node encoders*. Taking Graph Neural Networks as an example, a simple definition of u_i can be used as:

$$\begin{aligned} u_i &:= u_i^{(l)}, \\ u_i^{(h+1)} &= f(u_i^{(h)}, agg(\{u_j^{(h)} | j \in N(i)\})), \quad h < l. \end{aligned}$$

where $agg(\cdot)$ aggregates a set of node neighbors. Secondly, the graph concepts can be generalized to more types, such as a hierarchical structures concepts and sub-concepts. This can be easily implemented following the route introduced in our paper, by regarding sub-concepts as observations of the parent-concept and constrained to its center. Thirdly, our paper introduces an approach assuming deterministic concepts are available, however, this can be generalized to a probabilistic model which learns the probabilities of a node belonging to a concept. In addition, we are also exploring other priors besides the graph-concept-based Gaussian prior introduced in our paper to better capture the manifolds that graph nodes lies on.

References

1. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theor. Exp. **2008**(10), P10008 (2008)
2. Bojchevski, A., Günnemann, S.: Deep Gaussian embedding of graphs: unsupervised inductive learning via ranking. arXiv preprint https://arxiv.org/abs/1707.03815 (2017)
3. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems, vol. 29, pp. 3844–3852 (2016)
4. Du, L., Lu, Z., Wang, Y., Song, G., Wang, Y., Chen, W.: Galaxy network embedding: a hierarchical community structure preserving approach. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI 2018, pp. 2079–2085 (2018)
5. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, pp. 855–864. ACM (2016)
6. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems, vol. 30, pp. 1024–1034 (2017)
7. Henaff, M., Bruna, J., LeCun, Y.: Deep convolutional networks on graph-structured data. arXiv preprint https://arxiv.org/abs/1506.05163 (2015)
8. Huang, W., Zhang, T., Rong, Y., Huang, J.: Adaptive sampling towards fast graph representation learning. arXiv preprint https://arxiv.org/abs/1809.05343 (2018)
9. Kipf, T.N., Welling, M.: Variational graph auto-encoders. In: Proceedings of NIPS Bayesian Deep Learning Workshop (2016)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th International Conference on Learning Representations, ICLR 2017 (2017)
11. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Advances in Neural Information Processing Systems, vol. 27, pp. 2177–2185 (2014)

12. Li, Z., Zhang, L., Song, G.: GCN-LASE: Towards adequately incorporating link attributes in graph convolutional networks. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI 2019 (2019)
13. Li, Z., Zhang, L., Song, G.: SepNE: Bringing separability to network embedding. In: Proceedings of the 33rd AAAI's Conference on Artificial Intelligence, AAAI 2019 (2019)
14. Oh, S.J., Murphy, K., Pan, J., Roth, J., Schroff, F.: Modeling uncertainty with hedged instance embedding. arXiv preprint https://arxiv.org/abs/1810.00319 (2018)
15. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014, pp. 701–710. ACM (2014)
16. Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., Tang, J.: Network embedding as matrix factorization: unifying DeepWalk, LINE, PTE, and node2vec. In: Proceedings of the 11th ACM International Conference on Web Search and Data Mining, WSDM 2018, pp. 459–467 (2018)
17. Tang, J., Qu, M., Zhang, M., Mei, Q.: LINE: Large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, WWW 2015, pp. 1067–1077 (2015)
18. Lei, T., Jin, W., Barzilay, R., Jaakkola, T.: Deriving neural architectures from sequence and graph kernels. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017 (2017)
19. Tu, C., Zhang, W., Liu, Z., Sun, M.: Max-margin DeepWalk: discriminative learning of network representation. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI 2016 (2016)
20. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint https://arxiv.org/abs/1710.10903 (2017)
21. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, pp. 1225–1234 (2016)
22. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: Proceedings of the 7th International Conference on Learning Representations, ICLR 2019 (2019)
23. Yang, C., Liu, Z., Zhao, D., Sun, M., Chang, E.Y.: Network representation learning with rich text information. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI 2015, pp. 2111–2117 (2015)
24. Yang, S., Li, L., Wang, S., Zhang, W., Huang, Q.: A graph regularized deep neural network for unsupervised image representation learning. In: Proceedings of IEEE Conference on Computer Vision & Pattern Recognition, CVPR 2017 (2017)
25. Zhu, D., Cui, P., Wang, D., Zhu, W.: Deep variational network embedding in Wasserstein space. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2018, pp. 2827–2836 (2018)

Tuser^3: A Profile Matching Based Algorithm Across Three Heterogeneous Social Networks

Atika Mbarek[1,2](✉), Salma Jamoussi[1,2](✉), and Abdelmajid BenHamadou[1,2](✉)

[1] Multimedia InfoRmation Systems and Advanced Computing Laboratory, MIRACL,
University of Sfax, 3021 Sfax, Tunisia
mbarek.atika91@gmail.com, salma.jammoussi@isims.usf.tn,
Abdelmajid.benhamadou@isimsf.rnu.tn

[2] Digital Research Center of Sfax DRCS, 3021 Sfax, Tunisia

Abstract. Matching profiles can be defined as the process of determining whether two profiles in the same or different social networks are instances of the same user in real life or not. Because of the considerable increase in the number of created accounts in social networks, matching profiles across social networks has become a popular focus in a myriad of research works. Current methods in this field require accurate profile analysis to obtain a high user identification quality. However, such studies are restrictive since they do not consider the profile in its globality. In this work, we target the problem of user identification task based on their created profiles in social networks. Specifically, we conduct two main steps. First, we introduce a supervised model which employs the similarity of features for predicting matched profiles. Second, we propose Tuser^3 algorithm to search for the correct matched target profile of a source user on three heterogeneous social networks based on several user profiles features. Our experiments show the effectiveness of our method in matching profiles across the three target social networks.

Keywords: Matching profiles · Social networks · Machine learning · Tuser^3

1 Introduction

Due to the diverse functionalities of social network platforms, each online social network (OSN) offers different services to its users to share different types of information. Such information help in understanding the relations between users in social networks and characterizing their behaviors. Every user within an OSN is associated with a profile containing information that is relevant to him. A powerful way for identifying the user is to target his created profile in social networks such as Facebook, Twitter, YouTube, etc. However, such profile in one network may be insufficient to understand the users' interests. Furthermore, the information obtained solely from a single network is sometimes incomplete or missed. Therefore, a rich profile is very important to provide a high quality user identification. In this context, millions of users, who follow different lifestyles and belong to different demographic groups, regularly, share their interests on various

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 191–206, 2020.
https://doi.org/10.1007/978-3-030-65390-3_16

OSNs. Hence, matching profiles across social networks also serves as a powerful way in identifying users and creates a complete view of them. Such study can be implemented in many applications, such as user profiling identification, recommendation systems and cyber-security. Figure 1 provides an example of matching three different social networks. Each two networks will be matched independently. User X has two accounts in youtube and twitter, user Z has two accounts in Tumblr and Twitter and user Y has two accounts in Twitter. Here, two fundamental questions arise: can we link the two accounts of the user X in twitter and youtube? Can we find the matched profile of Z within the network Twitter knowing that we haven't any information about its matched profile? To answer these questions, some crucial challenges may occur:

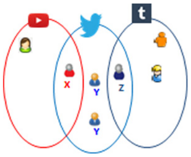


Fig. 1. Matching profiles example.

- **Social Networks Heterogeneity:** OSNs allow their users to share their content in many data formats, including textual data (Twitter), videos (Youtube), pictures (Instagram), blogs (Tumblr) and geolocations (Foursquare). Therefore, the user profile identification differs from one social network to another. Furthermore, some networks may contain rich information about profiles such as friends, followers and interests, while some others may not have such information. This unbalance between data makes it difficult to compare profiles across different social networks and compute their similarity based on unified available user's information.
- **Data Noise:** The quality of data is important in analyzing profiles in social networks. However, the user's noisily content such as spelling errors, abbreviations, non-standard words, false starts, repetitions, missing punctuations, non-standard and variety of languages may affect the result of user identity in social networks. The consequence of this noise is a reduction in the effectiveness of methods of linking profiles.
- **Missing Information:** users may simply forget or neglect to include information about their activities and interests; and some users simply do not reveal their real identity or delight in giving misleading information to amuse themselves. Such missing information is prevalent in user identity and its insufficiency can make it harder to provide useful linkage of users.

Motivated by the above challenges, we propose a method for matching users from multiple OSNs. Specifically, this work involves three main contributions: (i) we construct a supervised learning model which determines whether a pair of profiles matches. (ii) We propose the Tuser^3 algorithm that searches for the matched profile of a source one. (iii) We use the supervised learning model to test the output matched profiles returned by our algorithm and demonstrate the effectiveness of Tuser^3 algorithm. We finally conduct extensive experiments on datasets extracted from three different social networks. The remainder of the paper is organized as follows. We first discuss the related work in Sect. 2. We then present our matching profiles method in Sect. 3, followed by introducing the Tuser^3 algorithm in Sect. 4. We show experimental results in Sect. 5, before concluding our paper in Sect. 6.

2 Related Work

Matching profiles across social networks, also called User Identity Linkage (UIL), is a very challenging task. In fact, many state of the art literatures [1, 2] in this field have been studied in order to better understand why certain methods perform better than others, which methods are most suitable and for what type of data. Most of the existing profile matching approaches can be generalized into 4 categories:

2.1 Name-Based Approaches

Based on the assumption that benign users tend to make similar usernames on different OSNs [3–5], authors suggest methods to find matched profiles based on name search. Wang et al. tried to determine whether two usernames belong to the same user based on username similarity and username abbreviation [4]. They introduced a self-information vector model to quantify the similarity and proposed a dynamic programming algorithm to detect the initialism phenomenon between usernames. Another approach by Li et al. tried to connect Chinese user identities by linking their usernames [6]. They proposed a language mapping method that can translate different Chinese words of a given username into their corresponding Pinyin words. In another work, Liu et al. focus on the alias-disambiguation step that differentiates users with the same usernames [7]. They propose an unsupervised approach that uses n-gram probability to estimate rareness or commonness of a username. Furthermore, Zafarani et al. proposed a MOBIUS method [8] that identifies the user across sites in reference to the naming patterns of usernames. In the same context, [5] propose an algorithm for mapping users that starts by extracting candidate matching with similar names.

While the results of these methods are promising in matching profiles, they, however could not be powerful for some cases like fake profiles whose users do not reveal their real names. Another major limitation of these existing approaches is the possibility to find a lot of matched profiles with similar names of which only a few are true matched. Therefore, this could lead to many false matching profiles.

2.2 Account-Based Approaches

Mostly, a number of works suggest methods which include similarity between account features such as name, photo, email, biography, etc. In an earlier work, Vosecky et al. propose a method to identify users relying on nick name, email and date of birth [9]. Kucuk et al. analyzed the threat between an auxiliary OSN in which users share their real identities and an anonymous OSN by considering the publicly available attributes of users: username, location, gender and photo [10]. In [11], authors extracted the user's online digital footprints from publicly available information such as (username, display name, description, etc.) to match profiles. Furthermore, [12] proposed an algorithm that matches profiles by using available information of users. The algorithm applied a set of rules that compare the profile attributes (names, family names, usernames, etc.). In another work, [13] propose a method for matching profiles based on user's face photos. In a nutshell, their method uses face detection and comparison of face embeddings from the OSNs VKontakte and Instagram.

Almost, those methods are focusing on the easily accessible information about the account features. However, they have some disadvantages: Their works suffer from lack of relevant information that can be implicitly inferred such as interests, writing style, etc. Moreover, these techniques are heavily dependent on account features to be publicly available unlike many OSNs that may keep some sensitive information (e.g. age, gender or contact information) private.

2.3 UGCs-Based Approaches

Meanwhile, there were several efforts done on matching profiles based on the User Generated Contents (UGCs). Such information usually reflect the behavior properties such as when, where and what the user is posting. Authors in [14] propose a UGC-based user identification model based on spatial, temporal and content similarity of two UGCs. In addition, Stylometric features have also been used by researchers to link profiles across social networks [15]. In the same context, Keretna et al. analyzed short messages from Twitter and extracted linguistic features that can distinguish between the writing styles of different users [16]. In another work, Backes et al. studied whether anonymity within a single network can protect a user from being linked across sites [17]. Basically, they are focusing on the anonymity and likability threats to find a suitable representation of identities based only on public posts.

Along with textual data, social networks involve UGCs of various modalities such as photos or videos. Integration of such heterogeneous contents requires the development of efficient algorithms that can analyze the user's contents. Furthermore UGCs can be very sparse for certain users, who are not active in publishing their activities or those who made their profiles protected.

2.4 Network-Based Approaches

Network based methods are another promising way which has been used by several researches who exploit the graph structure of social networks for matching users. For instance, Man et al. proposed an anchor link prediction model that employs network embedding to predict the anchor links of two networks [18]. At the same time, Liu et al. studied the problem of mapping users by proposing a representation learning model to learn on an aligned network embedding for multiple networks [19]. In [20], authors proposed a deep neutral network based algorithm for user identity linkage which leverages the duality of mapping between any two social networks. Moreover, Koutra et al. focused on aligning bipartite graphs and proposed a gradient-descent based solution [21]. In other works [22, 23], researchers used neighborhood-based features to match profiles. They relied on different metrics such as common neighbors and Adamic/Adar score to measure the neighborhood similarities. Recently, Hongxu et al. proposed a framework that considers multi-level graph convolutions on both local network structure and hypergraph structure in a unified manner [30].

Although network structure is promising in tackling the matching profiles problem, it remains difficult for existing works to differentiate the real user identity from its neighborhoods. In addition, some network features can only be used when global networks are obtained, which is not practical in real world cases.

Motivated by the previous works mentioned above, we propose a new method for profile matching. The originality of this paper is not to focus on a particular type of features, but to target all the profile and reveal new possibilities for comparing profiles based on several extracted features. Apart from the previous works, we utilize name, account features and UGCs to solve the problem of matching profiles. In addition, our method provides a lot of additional information about users and proves that such information is useful for the user identification from different social networks.

3 Proposed Matching Profiles Process

The main contribution in this paper is matching profiles that belong to the same user in three heterogeneous social networks. We propose an algorithm that determines the matched profile of a targeted user in three social networks based on their attributes similarity. In particular, the algorithm is based on the use of several features that were implicitly inferred using different data mining methods and tools. These features allow us to enrich the user's identification and facilitate the final classification of profile pairs into matched and not matched.

3.1 Problem Formulation

The matching profiles process is formulated as a binary classification problem. There are two types of profile pairs: matching profile pairs and non-matching profile pairs. The main goal is to predict whether a pair of profiles belongs to the same user or not.

Given two networks: Source network N_S and target network N_T. We aim to match a source profile P_S within N_S, with its matched target profile P_T within N_T. We define that two profiles P_S and P_T are matched if they belong to the same real user. Formally, the matching process can be presented as a prediction function: f: $P_S \times P_T \rightarrow \{0, 1\}$:

$$f(P_S, P_T) = \begin{cases} 1, \ \textit{if } P_S \textit{ and } P_T \textit{ belong to the same user} \\ \quad 0, \ \textit{Otherwise} \end{cases} \tag{1}$$

Our proposed method can be generalized in two main steps as shown in Fig. 2. The first step involves three phases: Data collection, feature extraction and supervised model construction. The second step involves three phases including feature extraction, Tuser^3 algorithm and supervised model construction. In the data collection phase, we generate profile pairs from three OSNs: Twitter, Youtube and Tumblr. We then extract several types of features from each profile pair. The supervised model is trained based on the extracted features. Finally the trained model is used to predict whether a testing pair of profiles is matched or not. In the second step, we propose an algorithm called Tuser^3 which search for a source profile P_S within a source network N_S its matched target profile P_T within a target network N_T. The algorithm contains two main phases that will be clearly described in Sect. 4. Finally, the output of the algorithm is in the form of matched profile pairs that will be tested by the trained model to demonstrate the effectiveness of Tuser^3 in getting true matched profiles.

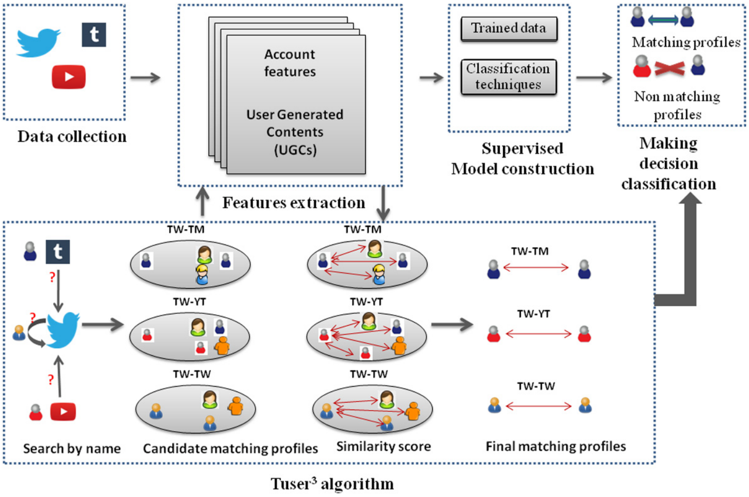


Fig. 2. Matching profiles architecture

3.2 Process of Profiles Gathering

3.2.1 Social Data Collection

Collecting profiles that belong to the same users in real world is a challenging task, even in the literature [1], there are so limited public datasets for matching profiles and even for the few public dataset, it does not consider the social networks that we target. Therefore, we create our own dataset to evaluate our work. Inevitably, the choice of a suitable network is an essential step in giving significant and rich information about users. For one reason or another, some OSNs have failed to become successful[1] because they are used by few people and do not contain rich information such as google+ and MySpace. However, there are relevant OSNs that provide rich features (e.g. Facebook) which become restrictive and put more limits on developer access. In this context, we believe that the most important information required to identify users is their UGCs. For this reason, we chose three popular OSNs[2] that contain rich information about users and are accessible by different APIs. Specifically, we target users from: Twitter, Youtube and Tumblr. Figure 3. Shows the availability of data in these networks.

[1] https://www.searchenginejournal.com/failed-social-media-sites/303421/.

[2] http://www.ebizmba.com/articles/social-networking-websites.

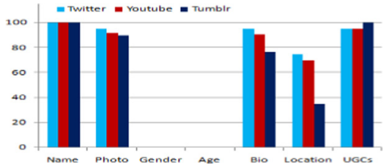


Fig. 3. Availability of data in the three social networks.

For the sake of more consistency and in order to get significant results, we extracted features only from the attributes that are common in the above mentioned networks. Although gender and age are totally missing in the three networks, UGCs are almost available except in some rare cases in which profiles are protected or users do not share any content. We did not consider such cases and we employ features whose availability is above 80% in the three networks (Name, photo, Bio and UGCs) so that we can match profiles with a high degree of confidence. Our methodology in collecting data is shown in Fig. 4.

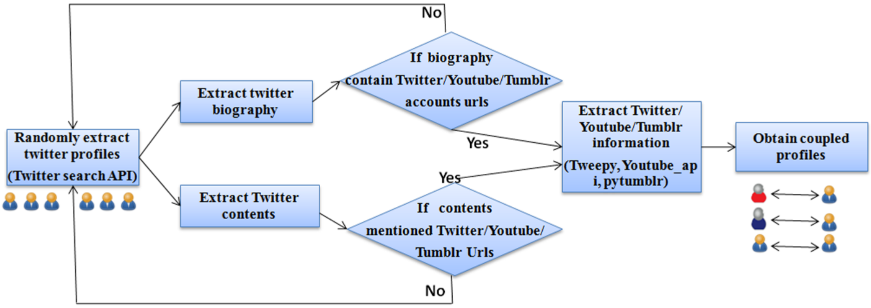

No
If biography contain Twitter/Youtube/Tumblr accounts urls
Extract twitter biography
Randomly extract twitter profiles (Twitter search API)
Yes
Extract Twitter/ Youtube/Tumblr information (Tweepy, Youtube_api, pytumblr)
Obtain coupled profiles
Yes
Extract Twitter contents
If contents mentioned Twitter/Youtube/ Tumblr Urls
No

Fig. 4. Methodology of collecting data

Specifically, in Twitter, some users make as detail in their biography[3] short links witch refer to their accounts in other sites. Motivated by this available information, we implemented a python code that uses Twitter Search Api to extract profiles whose account links are available. Furthermore, some users in twitter mentioned their identities in multiple networks through their UGCs. On top of that, we used tweepy[4], youtube_api[5] and PyTumblr[6] to perform data crawling from Twitter, Youtube and Tumblr. Additionally, we considered the dataset used by [24] which consists of 200 profile pairs from Twitter and Youtube. Our final dataset consisted of 1800 profile pairs in which 600 are matching and 1200 are non-matching. Each networks pair is considered a part: Twitter-Twitter (TW-TW), Twitter-Tumblr (TW-TM) and Twitter-Youtube(TW-YT). Our crawling strategy allows us to have accurate ground truth.

[3] short write up /'bio' /'about me' which the user provides about himself.

[4] http://docs.tweepy.org/en/latest/.

[5] https://pub.dev/packages/youtube_api.

[6] https://pypi.org/project/PyTumblr/.

3.2.2 Data Cleaning

As OSNs contain different types of data that are vast and noisy, data cleaning is a crucial step to get a good representation of features. Clearly, the quality of our extracted features will depend on how well we perform data cleaning. Therefore, we conduct a normalization step to clean noisy data from our collected dataset. This step includes removing spaces, useless stopwords, special characters, spelling errors, converting text into lower case, etc.

3.3 Features Extraction

The features extraction phase is very important to conduct the similarity between users. In this work, we exploit all available profile attributes and use them to infer implicit attributes that contribute much to the profile matching. The extracted attributes are used to compute the similarity between profile pairs and predict whether each pair belongs to the same real user or not. Formally, each pair of profiles belonging to a same user generated a similarity vector in the form:

$$V(P_S, P_T) = \left\{Sim(Attribute)_{(P_S,P_T)}\right\} \tag{2}$$

Where $Sim(Attribute)_{(P_S,P_T)}$ is the similarity between the attribute (e.g. photo) of the two profiles (P_S, P_T) in both networks $N_S \ and \ N_T$. The more similar the two profiles; the higher the probability that they belong to the same user. We provide in this section the details of the extracted features and the similarity metrics of each attribute. Regardless of semantics, we categorize features into Account features and UGCs.

3.3.1 Account Features

Generally, account features refer to the personal information about the user such as name, photo, biography, created profile, etc. Regardless of the heterogeneity of OSNs, we consider only attributes that are available in the three social networks that we target (see Fig. 3.). In particular, we consider the attributes name, photo and biography.

We use Match sequence algorithm [25] which compares two sequences of any type as long as the values are hashable. The algorithm returns the name similarity $Sim(Name)_{(P_S,P_T)}$ as real value between 0 and 1. We consider that the higher is the similarity of two usernames, the higher is the probability of being the same user.

The photo similarity is computed using FaceApi[7]. Given two photos of two profiles $P_S \ and \ P_T$; The tool returns the photo similarity $Sim(photo)_{(P_S,P_T)}$ as a binary value: 1(meaning the same face appears in two photos); 0 (totally different faces).

To calculate the biography similarity between two profiles $P_S \ and \ P_T$, we use the cosine similarity between the two extracted biographies from each profile.

$$Sim(bio)_{(P_S,P_T)} = \frac{\overrightarrow{bio_{P_S}}.\overrightarrow{bio_{P_T}}}{|bio_{P_S}||bio_{P_T}|} \tag{3}$$

[7] https://azure.micros.oftcom/en-us/services/cognitive-services/face/.

Where $\overrightarrow{bio_{P_S}}$ *and* $\overrightarrow{bio_{P_T}}$ are word-frequency vectors of the provided biographies by each profile pair P_S *and* P_T. The resulting similarity is real value which ranges from 0(meaning exactly different biography), to 1 (meaning exactly the same biography).

3.3.2 Features Based on UGCs

Generally, UGCs contain rich information that helps to understand the behaviors and interests of users. Obviously, such content may give great number of attributes that can characterize users. We therefore need to exploit the UGCs to generate more suitable attributes and enrich the user identification. In our case, UGCs refer to tweets in Twitter, videos in Youtube and blogs in Tumblr. Each content can consist of four types of features: Timeline features, emotional features, writing style and n-grams.

Timeline Features: Temporal information is relevant for determining the precise time during which the user is active on his profile. In fact, certain users usually post contents at night, while other users share posts at morning. In addition, some users are very active in the weekend unlike other users who usually are active in ordinary days. Therefore, exploiting the availability of posting time and as in [27], we extracted such timeline features as: frequency of posting per morning/afternoon/night, frequency of posting per summer/winter/autumn/spring/autumn, frequency of posting per ordinary day/weekend, etc. In total we extracted 12 timeline features. The similarity between a timeline feature can be represented as the difference in two posting time. We compute the difference of each timeline feature extracted from P_S *and* P_T as bellow:

$$Sim(Timeline)_{(P_S,P_T)} = |Freq(Timeline)_{P_S}| - |Freq(Timeline)_{P_T}| \tag{4}$$

Where $Freq(Timeline)$ is computed in the same way for each timeline feature, e.g. the frequency of posting at morning is computed by the sum of posts published at morning divided by the total number of posts. The resulting similarity gives a score in the range [0, 1]. Since we focus on pairwise distances, the distance is considered as the inverse of similarity, as the distance between timelines increases, similarity decreases.

Emotional Features: Due to the nature of OSNs, on which people share contents about their views on a variety of topics, complain, and express their sentiment regarding problems that encounter them, several emotional features may be extracted from the UGCs. These features give useful prediction of what the user thinks and feels. To analyze the emotional content of a profile, we focus on three relevant information:

Empath features: We utilize empath [29] Api as it was considered to be a powerful mechanism for user identification. The great benefit of this Api is its ability to analyze text across 200 gold standard topics and emotional categories (e.g. violence, science, government, etc.).For each topic, empathy returns a score in the range [0,1] which tells us how strongly it relates to the text. For example, given the topic “nature”, the Api returns a score of 0.1, which means that the published content of the user would be weakly related to the topic “nature”. In total, we extracted 71 empath features. Finally, we compute each empathy feature similarity $Sim(empathfeature)_{(P_S,P_T)}$ for each profile pair from the distance of the two computed topic scores.

Sentiment analysis: Sentiment analysis presents a growing area which has become a field of interest for many researches. Being able to extract sentiments from UGCs will highly increase the chances of identifying the sentimental side of users. For this purpose, we use textblob[8] to extract sentiments from UGCs. Specifically, we consider the polarity (positive, negative and neutral) and subjectivity score of contents.

Given the content to analyze, the sentiment analysis tool returns the probability for positive, negative, neutral sentiment and subjectivity in the text. For each sentiment feature is assigned as core between 0 and 1.The similarity $Sim(sentiment)_{(P_S,P_T)}$ was then taken as the difference between the two sentiment scores of a profile pair. The lower the difference in scores, the higher the similarity of the two sentiments.

Emojis: People in social networks usually use emojis and other characters that express particular meanings. Obviously, the more we specify the type of the used emojis, the more we get precise recognition of the emotional state of users. For this purpose, we create a dictionary for emojis features. The construction of the dictionary was based on the classification of emojis into nine classes including happy, sad, satisfied, gleeful, angry, surprised, disappointed, romantic and disgusted.

Writing Style Features: Extracting writing style from text is a very challenging task [13, 27]. As users in OSNs usually use written text to communicate, obtaining such features may highly increase the chances of distinguishing a user from another. In this context and as in [28], we extracted several writing style features that include different characteristics such as the frequency of using: special characters, digits, capital and small letters, short and long sentences, etc. In total, 32 features were extracted.

Similar to emotional features, the similarity of each writing style feature $Sim(Writingstyle)_{(P_S,P_T)}$ between each profile pair (P_S, P_T) is also computed as the distance of the computed writing style frequencies. The lower the distance between two writing styles frequencies, the higher their similarity.

N-grams: N-grams are one of relevant features that can identify users. Indeed, they can collectively reflect the topical interests of users. Furthermore, frequent words will also serve to know what the best user interest is. For instance, if the user is interested in music, normally, he usually posts words related to music (e.g. pop, rock, etc.). Therefore, for each user, we extract the most used frequent words, bigrams and trigrams. In order to compute the n-grams similarity as well as frequent words of a profile pair (P_S, P_T), we extract the common n-grams mostly used by (P_S, P_T), and divide them on the minimum of the two initial extracted n-grams. This measure is defined as:

$$Sim(n-grams)_{(P_S,P_T)} = \frac{|n-grams(P_S) \cap n-grams(P_T)|}{Min(|n-grams(P_S)|, |n-grams(P_T)|)} \tag{5}$$

[8] https://textblob.readthedocs.io/en/dev/.

4 Tuser^3 Algorithm

We propose the Tuser^3 algorithm (Twitter-user↔YouTube-user, Twitter-user ↔Tumblr-user, Twitter-user ↔ Twitter-user) which searches the matched target profile P_T of a source profile P_S. As shown in Fig. 2, Tuser^3 starts by searching the candidate matching profiles. A straightforward way for the matching process is to use the username as a first step to extract a list of candidate matching profiles. However, as mentioned in Sect. 2.1, depending only on username may lead to many false matching. For instance, the algorithm may return five matched profiles of a source profile with similar names; one is real matched and four are false matched. Thus, a filtering step is required to choose only real matched. For this purpose Tuser^3 is based on two main steps: (1) candidates discovery, (2) matched profiles selection.

4.1 Candidates Discovery

Based on related works in [3, 8, 11], authors suggest that when users make their names on different OSNs, they tend to make similar usernames, even if they change their usernames from a network to another, they make new usernames similar to the old ones. For rare cases like fake profiles, users tend to hide their real names and create new names in each network. In our work, we do not consider such cases. In fact, since names are often recorded with different spellings, applying exact matching leads to poor results [26]. Therefore, we define a matching name method that extracts matched target profiles with similar names of a source profile P_S. Specifically, for each P_S we extract all words combinations that make up the profile name as described in [24]. We further add the first half of the letters that make up the name (e.g. Cristiano → Crist). Finally, we search for profiles having in their names this list of combinations.

4.2 Matched Profiles Selection

Once the candidate matched profiles are extracted, we need a filtering step to select only real matched. In this step, the source profile P_S is compared to each candidate profile P_T among the list of candidate profiles based on a similarity score that refers to which extent $P_S\ and\ P_T$ are matched. Thus, we assign to each profile pair a similarity score $Score(P_S, P_T)$ which is computed as the average of all similarities values:

$$Score(P_S, P_T) = \frac{\sum Sim(Attribute)_{(P_S,P_T)}}{\sum Attribute} \tag{6}$$

We note that profile pairs with high similarity score are likely to be matched. Therefore, we define a threshold to decide which pairs are matched. Finally, the algorithm chooses only profile pairs that have a similarity score above the threshold.

4.3 Supervised Model

As mentioned above, we build a supervised model to tackle the profile matching problem. The goal of the supervised model is to learn the prediction function mentioned in Sect. 3.1. The latter is learned based on the similarity vectors for the profile pairs (P_S, P_T). Once the prediction function is learned, the evaluation can be performed on a testing set of profile pairs. Specifically, the matched profiles returned by Tuser[3] will be tested using the supervised model by giving it as input a profile pair (P_S, P_T) from a pairwise network (N_S, N_T) to be classified as matched or a non-matched. We employ different supervised classification techniques including: Random forest, Bayes Net, SMO, KNN and Adaboost. Here, the goodness of the classifier is evaluated based on three metrics: precision, recall and F-measure.

5 Evaluation

In this section, we evaluate our proposed method by using real data from the three OSNs mentioned above. We also study the impact of the extracted attributes to the profile-matching task.

5.1 Baselines Comparison

We compare our method with two previous works, which also require account attributes and UGCs. We note that there are many methods which used account features and UGCs which are also applicable for matching profiles as we mentioned in Sect. 2. In this paper, we compare our methods with the following previous works:

Name Match: The Name-Match method considers that two usernames belong to the same user if and only if they have exactly the same name.

Tuser^2: Tuser^2 [24] uses different attributes to match profiles. It differs from our method by the search for candidate profiles step as well as the number of attributes.

5.2 Supervised Model Evaluation

We describe a first evaluation that aims at testing our constructed supervised model based on all features similarity and classifiers listed above. We conduct ten-fold cross validation. The evaluation results are presented in Table 1.

Table 1. Classification results of matching profiles.

Classification techniques	Precision	Recall	F-measure
Random forest	90.3	90.4	90.3
Bayes Net	**98.5**	**98.5**	**98.5**
SMO	93.8	93.6	93.4
KNN	76	76.4	76.7
Adaboost	92.8	92.7	92.5

The results are significant and so are the classifiers. Specifically, BayesNet is best suited, in terms of recall, our work achieves 98.5%. Based on these results, we can conclude that our extracted features are of crucial importance for matching profiles.

5.3 Tuser^3 Performance Evaluation

Similar to [1, 20], we used MAP, AUC and precision@k as good metrics in the con-text of profile matching which can measure the performance of how well the algorithm can rank matching profiles higher than any non-matching profile. The higher the values of each of these measures, the better are the performance. We select 30 profiles from each network and search for their matched target profiles in twitter.

As presented in Table 2, Tuser^3 outperforms the two previous works in terms of MAP and AUC. Furthermore, Table 3 shows that our work achieves the highest precision in all values of k, which demonstrate the effectiveness of our algorithm to rank the matched target profile as high as possible. Another explanation is the richness of the used features which carries to high confidence of matching.

Table 2. Algorithm performance comparison for each social network pair.

	Match-name		Tuser2		Tuser3	
	MAP	AUC	MAP	AUC	MAP	AUC
TW-TM	0.32	0.23	0.4319	0.5021	**0.4905**	**0.5597**
TW-YT	0.4708	0.3768	0.3587	0.4317	**0.4996**	**0.5306**
TW-TW	0.2982	0.2512	0.2637	0.4654	**0.4369**	**0.5046**

Table 3. Precision@k comparison for each social network pair.

	Match-name				Tuser2				Tuser3			
	p@5	p@10	p@15	p@21	p@5	p@10	p@15	p@21	p@5	p@10	p@15	p@21
TW-TM	0.32	0.32	0.32	0.32	0.56	0.56	0.56	0.64	**0.6**	**0.64**	**0.64**	**0.72**
TW-YT	0.60	0.60	0.60	0.60	0.47	0.69	0.78	0.91	**0.65**	**0.78**	**0.86**	**0.91**
TW-TW	0.34	0.34	0.34	0.34	0.42	0.57	0.73	0.88	**0.53**	**0.69**	**0.80**	**0.88**

Finally, the output of tuser^3 is tested by the supervised model to determine whether the extracted profiles match or not. Table 4 shows the precision, recall and F-Measure of Tuser^3. In the same table, we also compare Tuser^3 with the baselines approaches. We observe that Tuser^3 achieves a high matching performance in terms of recall, precision and F-Measure in the tree network pairs. We also observe that tuser^3 notably provides significantly higher precision values compared to the baseline approaches.

Table 4. Tuser^3 evaluation based on the supervised model in the three social networks pairs.

		Match-name			Tuser2			Tuser3		
		Precision	Recall	F-Measure	Precision	Recall	F-Measure	Precision	Recall	F-Measure
KNN	TW-TM	35.7	33.54	34.06	80	50	66.7	**100**	**96.9**	**98.4**
	TW-YT	38.4	37.22	37.8	90.1	66.7	85	**95.33**	**82.9**	**90.6**
	TW-TW	35.7	33.54	34.06	80	50	66.7	**100**	**96.9**	**98.4**
Bayes Net	TW-TM	39.2	32	33.75	100	99.5	99.7	**100**	**99.4**	**99.7**
	TW-YT	38.4	35.61	37.1	100	100	100	**100**	**100**	**100**
	TW-TW	30.2	19.7	24.71	83.33	55.5	71.4	**100**	**95.8**	**97.9**
Random forest	TW-TM	38.4	30.22	30.8	80	66.7	70.2	**100**	**100**	**100**
	TW-YT	38.4	30.22	30.8	80	66.7	70.2	**100**	**100**	**100**
	TW-TW	39.66	31.1	32.7	90.4	79.2	88.4	**96.33**	**89.9**	**94.7**
SMO	TW-TM	39.2	24	27.4	93.4	82.8	90.6	**96.66**	**87.9**	**93.6**
	TW-YT	38.4	30.22	30.8	96.33	83.3	94.4	**100**	**93.6**	**96.7**
	TW-TW	39.66	33.8	34.2	100	97.8	98.9	**100**	**100**	**100**
Adaboost	TW-TM	39.2	25.33	26.6	100	92.4	96	**100**	**92.4**	**96**
	TW-YT	38.4	37.22	37.8	100	83.3	97.8	**100**	**96.4**	**98.2**
	TW-TW	35.7	28.3	31.1	73.33	41.7	58.8	**96.66**	**81.9**	**90.1**

5.4 Discussion

In the state of the art studies in the field of user profiles linkage, researches usually use the precision metric for evaluating their methods. However, the performance of their works is related to the assumption that one user has at most one account in an OSN which is not the case in real life. Intuitively, a good way for evaluating the performance of Tuser^3 is considering the recall in getting the matched profiles of source ones. Specifically, the performance of Tuser^3 can be well shown if we determine how many matched profiles the algorithm returns among a set of real matched profiles. However, in real word, a user may have multiple accounts in an OSN that we cannot simply determine. This makes the problem of considering the recall measure more complicated. For instance, we get five profiles that we know their matched accounts in Twitter in which two profiles have three matched accounts and three profiles have two matched accounts. We use those tested profiles as an input to Tuser^3 algorithm and compute the recall. The obtained recall is 60% that is considered as relevant value in the case of the exiting of more than one matched target profile of a source one.

6 Conclusion and Future Work

Profile matching methods serve as an important task in the identity resolution process. In this paper, we introduced a supervised model for the profile matching prediction. We further proposed Tuser^3 algorithm which consists of the discovery of the matched target profiles of source ones. The effectiveness of our proposed model was proved in three different social networks. For future directions, many new tasks in matching profiles require to be considered. Specifically, nowadays with the spread of Coronavirus disease to every inhabited continent, we plan to explore user profiles methods to identify people who were contaminated and then recovered based on their accounts in OSNs and reveal the precautions that they have already taken to fight this epidemic.

References

1. Shu, K., Wang, S., Tang, J., Zafarani, R., Liu, H.: User identity linkage across online social networks: a review. ACM SIGKDD Explor. Newslett. **18**(2), 5–17 (2017)
2. Farseev, A.: 360 user profile learning from multiple social networks for wellness and urban mobility applications Doctoral dissertation, National University of Singapore (2017)
3. Perito, D., Castelluccia, C., Kaafar, M.A., Manils, P.: How unique and traceable are usernames? In: Fischer-Hübner, S., Hopper, N. (eds.) PETS 2011. LNCS, vol. 6794, pp. 1–17. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22263-4_1
4. Wang, Y., Liu, T., Tan, Q., Shi, J., Guo, L.: Identifying users across different sites using usernames. In: ICCS, vol. 2016, pp. 376–385, January 2016
5. Panchenko, A., Babaev, D., Obiedkov, S.: Large-scale parallel matching of social network profiles. In: Khachay, M.Y., Konstantinova, N., Panchenko, A., Ignatov, D.I., Labunets, V.G. (eds.) AIST 2015. CCIS, vol. 542, pp. 275–285. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26123-2_27
6. Li, Y., Zhu, J., Zhou, Z., Zhou, B., Wu, X.: Connecting Chinese users across social media sites. In: 3rd International Conference on Material, Mechanical and Manufacturing Engineering (IC3ME 2015). Atlantis Press, August 2015
7. Liu, J., Zhang, F., Song, X., et al.: What's in a name? An unsupervised approach to link users across communities. In : Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, pp. 495–504 (2013)
8. Zafarani, R., Liu, H.: Connecting users across social media sites: a behavioral-modeling approach. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2013)
9. Vosecky, J., Hong, D., Shen, V.Y.: User identification across multiple social networks. In: 2009 First International Conference on Networked Digital Technologies, pp. 360–365. IEEE, July 2009
10. Kucuk, V., Ayday, E.: Profile matching across unstructured online social networks
11. Malhotra, A., Totti, L., Meira, Jr., W., Kumaraguru, P., Almeida, V.: Studying user footprints in different online social networks. In: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 1065–1070, August 2012
12. Bennacer, N., Nana Jipmo, C., Penta, A., Quercini, G.: Matching user profiles across social networks. In: Jarke, M., et al. (eds.) CAiSE 2014. LNCS, vol. 8484, pp. 424–438. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07881-6_29
13. User profiles matching for different social networks based on faces identification (2019)
14. Li, Y., et al.: Matching user accounts based on user generated content across social networks. Future Gener. Comput. Syst. **83**, 104–115 (2018)
15. Brounstein, T.R., et al.: Stylometric and temporal techniques for social media account resolution. No. SAND2017-2965C. Sandia National Lab. (SNL-NM), Albuquerque, NM (United States) (2017)
16. Keretna, S., Hossny, A., Creighton, D.: Recognizing user identity in twitter social networks via text mining. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics, pp. 3079–3082. IEEE, October 2013
17. Backes, M., et al.: On profile linkability despite anonymity in social media systems. In: Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society (2016)
18. Man, T., Shen, H., Liu, S., Jin, X., Cheng, X.: Predict anchor links across social networks via an embedding approach. In: IJCAI, vol. 16, pp. 1823–1829, July 2016
19. Liu, L., Cheung, W.K., Li, X., Liao, L.: Aligning users across social networks using network embedding. In: IJCAI, pp. 1774–1780, July 2016

20. Zhou, F., et al.: Deeplink: a deep learning approach for user identity linkage. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications. IEEE (2018)
21. Koutra, D., Tong, H., Lubensky, D.: Big-align: fast bipartite graph alignment. In: 2013 IEEE 13th International Conference on Data Mining. IEEE (2013)
22. Zhang, Y., Tang, J., Yang, Z., Pei, J., Yu, P.S.: Cosnet: connecting heterogeneous social networks with local and global consistency. In: KDD (2015)
23. Kong, X., Zhang, J., Yu, P.S.: Inferring anchor links across multiple heterogeneous social networks. In: CIKM (2013)
24. Mbarek, A., Jamoussi, S., Hamadou, A.B.: Tuser^2: A new method for twitter and youtube matching profiles. In: Nguyen, N.T., Chbeir, R., Exposito, E., Aniorté, P., Trawiński, B. (eds.) ICCCI 2019. LNCS (LNAI), vol. 11684, pp. 110–121. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28374-2_10
25. Musser, D.R., Nishanov, G.V.: A fast generic sequence matching algorithm. arXiv preprint arXiv:0810.0264(2008)
26. Christen, P.: A comparison of personal name matching: techniques and practical issues. In: Sixth IEEE International Conference on Data Mining-Workshops (ICDMW 2006), pp. 290–294. IEEE, December 2006
27. Mbarek, A., et al.: Suicidal profiles detection in Twitter. In: WEBIST (2019)
28. Basti, R., et al.: Arabic Twitter user profiling: application to cyber security. In: WEBIST (2019)
29. Fast, E., Chen, B., Bernstein, M.S.: Empath: understanding topic signals in large-scale text. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (2016)
30. Chen, H., Yin, H., Sun, X., Chen, T., Gabrys, B., Musial, K.: Multi-level graph convolutional networks for cross-platform anchor link prediction. arXiv preprint arXiv:2006.01963 (2020)

Encrypted Traffic Classification Using Graph Convolutional Networks

Shuang Mo[1], Yifei Wang[2], Ding Xiao[1](✉), Wenrui Wu[1], Shaohua Fan[1], and Chuan Shi[1]

[1] Beijing University of Posts and Telecommunications, Beijing, China
{moshuang,dxiao,wuwenrui,fanshaohua,shichuan}@bupt.edu.cn
[2] State Grid Jibei Information and Telecommunication Company, Beijing, China
wangyf341@163.com

Abstract. Traffic classification plays a vital role in the field of network management and network security. Because of the continuous evolution of new applications and services and the widespread use of encrypted communication technologies, it has become a difficult task. In this paper, we study the classification of encrypted traffic, where the purpose is to firstly distinguish between Virtual Private Networks (VPN) and regular encrypted traffic, and then classify the traffic into different traffic categories, such as file, email, etc. The available information in encrypted traffic classification is composed of two parts: the complex traffic-level features and the diverse network-side behaviors. To fully utilize these two parts of information, we propose an approach, called **E**ncrypted **T**raffic **C**lassification using **G**raph **C**onvolutional **N**etworks (**ETC-GCN**), which incorporates traffic-level characteristics with convolutional neural networks (CNN) and network-wide behavior with graph convolutional networks (GCN) in the communication network. We compare the proposed approach with existing start-of-the-art methods on four experiment scenarios, and the results demonstrate that ETC-GCN can improve the classification performance by considering the information of neighbor endpoints that communicated, and the internal features of the traffic together.

Keywords: Traffic classification · Encrypted traffic classification · Graph convolutional networks · Convolutional neural networks

1 Introduction

In recent years, traffic classification technology has received more and more attention due to the implementation of network quality of service (QoS) and network security principles. However, the continuous expansion of the network and innovation of communication technologies make the characteristics of the traffic complex and diverse. And encryption techniques, which encode data before it is sent to avoid information leakage, have become ubiquitous in nowadays' networks and serves as the basis for secure communication. It masks the characteristics of the data, making it difficult to distinguish. This continuous expansion, innovation, and concealment make classification on encrypted traffic become a challenging task. Generally, traffic classification can be

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 207–219, 2020.
https://doi.org/10.1007/978-3-030-65390-3_17

categorized into three types: traffic identification, traffic characterization, and detailed traffic classification in terms of different granularities [1]. Specific to the classification of encrypted traffic, the traffic identification distinguishes between encrypted traffic and VPN traffic, the traffic characterization divides the traffic into different application categories, and the detailed traffic classification classifies the traffic into specific applications. We study the problem of traffic identification and characterization of encrypted traffic.

Recent traffic classification methods can be divided into three types: port-based methods, payload-based methods, and flow statistics-based according to the features involved [2]. Furthermore, encrypted traffic classification is more challenging due to the following characteristics:

1. *Complexity of traffic-level features.* Port-based methods make traffic identification by comparing ports of applications or protocols defined by Internet Assigned Numbers Authority (IANA) [3]. But they are too simple to cope with applications using random ports and ports disguise. Payload-based methods exacted payload data from the application layer of the ISO/OSI reference model, also called data packet inspection (DPI), which focuses on matching packets with existing signatures or patterns [4]. However, DPI requires payload examination, which is not computationally efficient and cannot handle with encapsulated, encrypted traffic. Flow statistics-based methods firstly compute statistical features from the packets. After that, these features were fed into machine learning algorithms like C4.5 decision tree, Support Vector Machine (SVM), and K-nearest neighbor (KNN) [5–8]. But their performance heavily depends on feature engineering by domain experts and has poor generalization. Besides, some researchers focus on training models on raw data of traffic with deep learning techniques directly such as CNN, Stack Autoencoder(SAE), attention mechanism or Recurrent Neural Networks (RNN) to build an end-to-end model [9–14]. However, these methods only consider the internal characteristics of the traffic and ignore external information and face performance bottlenecks.
2. *Diversity of network-side behaviors.* Some researchers build a graph from processed traffic, with an edge between any two communicated IP addresses. After that, they feed the features of the graph into a K-means model to make classification [15]. In this way, the model can additionally extract information for communication and interaction with the neighbor communication hosts. That is, the host may send flows in the same type. However, the vertices of the graph are IP addresses, which are too coarse to classify. As is shown in Fig. 1, the diversity of network network-side behaviors that Host A is assigned only one IP address from the Internet but sends two traffic types of email, Peer-to-Peer (P2P), due to different network applications. And Host E uses the same port 53 to send and receive traffic, and as the sending end and receiving end, the traffic types sent are different. The characteristics of the same port as the sender and the receiver are different, which is the heterogeneity of the network-side behaviors. Ignoring these two pieces of information may interfere with traffic classification.

Therefore, we propose an **E**ncrypted **T**raffic **C**lassification using **G**raph **C**onvolutional **N**etworks (**ETC-GCN**), integrating traffic-level characteristics and

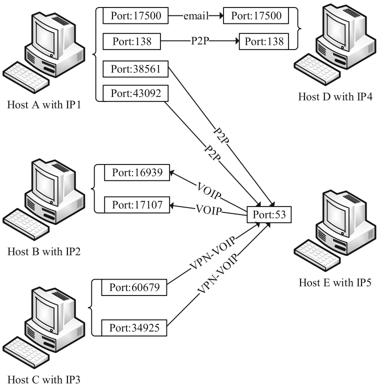


Fig. 1. An example of communication between endpoints on the Internet. An endpoint consists of an IP address and port.

network-wide behaviors in a uniform framework. In ETC-GCN, we firstly use a one-dimension CNN (1D-CNN) to learn the embeddings of raw features of traffic. And then, we build communication between endpoints as a heterogeneous graph with two types of endpoints: source and destination endpoints, which can handle the diversity and heterogeneity of the communication. After that, we model the network-wide behaviors in the graph by a heterogeneous GCN. In summary, the main contributions of this paper are:

- We present an innovative encrypted traffic classification method called ETC-GCN, which captures traffic-level characteristics by CNNs and network-wide behaviors by a heterogeneous GCN. To our knowledge, this is the first work to use GCN to utilize both side information in a unified framework for traffic classification.
- We evaluate ECT-GCN on four different experiment scenarios. It achieves outstanding classification results comparing to the baselines.

The rest of this paper organizes as follows. Section 2 describes the related works. Section 3 describes a formal definition of our problem. Section 4 introduces the details of our proposed method. Section 5 compares and analyzes the experimental results of our approach. And Sect. 6 concludes this paper and outlines future work.

2 Related Work

Traffic classification has been well studied in the last decades. Many classification methods have been proposed to exploit the internal characteristics of the traffic, and traditional traffic classification methods' effect is significantly reduced because of the vast usage of encrypted techniques [16]. To address this issue, some researchers try to use flow statistic features computed from packets and combine with machine learning methods. A C4.5 decision tree is used to classify flow statistics features of proxy traffic, which identifies proxy traffic in traffic log files without accessing communication endpoints [17]. Some researchers use C4.5 and KNN to study the effectiveness of flow-based time-related features [1]. However, machine learning-based methods heavily depend on feature engineering by domain experts and lack of generalizability.

Due to deep learning that can extract features from raw data, it has been used in traffic classification in recent years. Some researchers firstly preprocess traffic as image-struct data and feed the images into a 1D-CNN that makes classification [9]. Also, a combined model employing SAE and CNN is used to classify the encrypted network traffic in the granularity of both traffic characterization and application identification [12]. Besides, some researchers used RNNs, i.e., Long Short-Term Memory (LSTM), combing with the attention mechanism to model time-series networks on the ISCX2016 dataset [14]. However, these deep learning-based models only consider the internal characteristics of the flow, ignoring other potentially useful information and face bottlenecks in classification performance. Besides, some researchers propose a method called Graph-based classification, Graption, and they build a graph from processed traffic, where an edge between any two IP addresses that communicate. After that, they feed the attributes of the graph into a K-means model to make the classification [15]. However, the vertices of the graph are IP addresses, which do not consider the diversity and heterogeneity of network-side behaviors. Therefore, the network-side behaviors captured by the model are mixed and uncertain.

Compared to these methods, our proposed ETC-GCN firstly uses a 1D-CNN to learn traffic-level characteristics. And then, build source and destination endpoints communication in the network as a graph, whose vertices consist of IP address and port. This graph is heterogeneous because the same IP address and port can be used as both the sender and receiver of the flow. Therefore, we use heterogeneous GCN to learn network-side behaviors.

3 Problem Definition

A flow is defined as a sequence of packets with the same five-tuple for source IP, source port, destination IP, destination port, and transport-level protocol [18]. And our purpose is to classify flows at network communication, which can be defined as an edge classification problem in a directed bipartite graph with two types of vertices: source endpoints and destination endpoints and flow as the edge. So it has attributes on both vertices and edges.

We are inspired by the previous work of heterogeneous graph embedding work [22], which incorporates information from neighbors of a heterogeneous graph for edge classification. We build the behavior of endpoints communicating in the network as a heterogeneous graph $G(S, D, E)$, where S is the set of source endpoint nodes (vertices), which are represented by source IP address and source port. D is a set of destination endpoint nodes (vertices), which are composed of a destination IP address and destination port. And E is the set of network flows (edges) transmits from source endpoint to destination endpoint. An edge $e \in E$ from a source endpoint $s \in S$ to a destination endpoint $d \in D$ exists if s send a flow e to d. Also, given a vertex $v \in S \cup D$, let $N(v)$ be the one-hop neighbor set of vertices, i.e. $N(v) = \{v' \in S \cup D | (v, v') \in E\}$. $E(v)$ represents the edge connected to v. Let $S(e)$ and $D(e)$ indicate the source endpoint node and destination endpoint node of edge e. This heterogeneous graph is called **Communication Graph**. See Fig. 1 for a real-world example.

4 Methodology

4.1 Overview

In this paper, we propose a model to classify encrypted traffic called ETC-GCN. We firstly use a 1D-CNN to learn traffic-level characteristics and use GCN to learn network-side behaviors. The proposed method mainly consists of three steps, traffic data preprocessing, traffic transforming, and traffic classification. In the traffic data preprocessing step, we process the raw traffic data into an image-structure with fixed size to feed into CNN. After that, a 1D-CNN is used to transform traffic into embeddings, which can capture traffic-level characteristics. After that, a heterogeneous GCN is applied to learning the network-side behaviors with the embeddings and communication graph. Finally, a dense layer with softmax is used to classify the final embeddings of the traffic. Next, we will describe the above steps in detail.

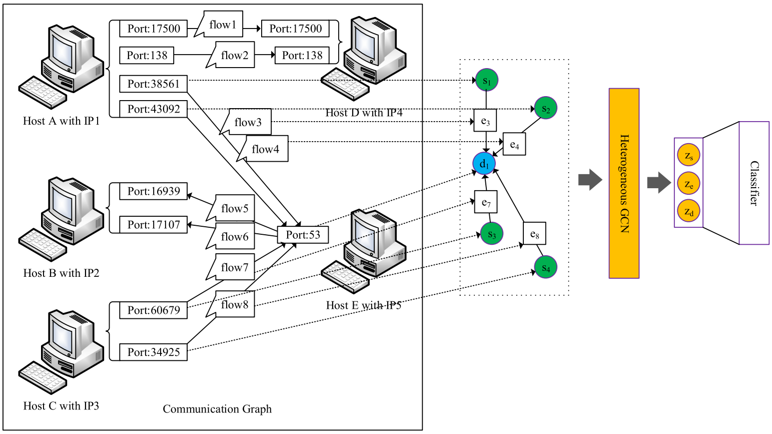


Fig. 2. A schematic of ETC-GCN which combines two graphs. Heterogeneous GCN acts on the Communication Graph and provides source and destination endpoints, flow embeddings z_s, z_d and z_e respectively.

4.2 Traffic Data Preprocessing

We firstly split the traffic data into multiple flows according to the flow definition, and flows are considered directional (forward and reverse directions). After that, for each flow, we pad or truncate the flow into an image-structure with fixed size of pixel, each pixel represents a byte in the traffic. It can be described as Eq. 1,

$$\mathbf{x} = x_1 \oplus x_2 \oplus \ldots \oplus x_n, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is a vector of traffic sample, $\oplus$ is the concatenation operator, and n is the number of pixels in the traffic image.

Besides, we extract the corresponding IP addresses and ports for each flow to build the communication graph and merge the corresponding port information defined by the IANA. In detail, the features of a port are shown in Table 1.

Table 1. Port features.

Feature	Description
port	The number of a port that is ranging from 0 to 65,535
type	The type of a port, where a system port is in range 0 to 1,023, user port is in range 1,024 to 49,151, and dynamic port is range 49,152 to 65,535
TCP	Whether support TCP protocol in the transport layer
UDP	Whether support TCP protocol in the transport layer
SCTP	Whether support SCTP protocol in the transport layer
DCCP	Whether support DCCP protocol in the transport layer

4.3 Feature Transforming

The raw features of the flow are difficult to learn by GCNs directly, so we transform them into embeddings with 1D-CNN before merging with source and destination endpoint features, which is shown as Eq. 2,

$$\mathbf{h}_e^0 = 1\mathrm{D} - \mathrm{CNN}(\mathbf{x}), \tag{2}$$

where $\mathbf{x}$ represents the vector of traffic image, and $\mathbf{h}_e^0$ is the initial embedding of the traffic.

4.4 Graph Convolutional Network

The GCN-based models are composed of multiple propagation layers, and the parameters are propagated through layers. And all nodes in each propagation layer will be updated at the same time. Generally, a propagation layer can be divided into two sub-layers. Given a graph $G = (V, E)$ with node $v \in V,\ (v, v') \in E$, node feature $\mathbf{x}_v = \mathbf{h}_v^0$ for $v \in V$. And $\mathbf{h}_v^l$ is the hidden vector of node v learned by l-th of the GCN. Then, for a GCN with an L layer, its aggregation sublayer and combination sublayer at l-th layer ($l = 1, 2, \ldots, L$) can be expressed as:

$$\mathbf{h}_{N(v)}^l = \sigma(\mathbf{W}^l \cdot AGG(\{\mathbf{h}_{v'}^{l-1}, \forall v' \in N(v)\})), \tag{3}$$

$$\mathbf{h}_v^l = COMBINE(\mathbf{h}_v^{l-1}, \mathbf{h}_{N(v)}^l), \tag{4}$$

where $N(v)$ is a set of v's neighbors, $AGG(\cdot)$ is a function that aggregates embeddings from neighbor nodes of v, $\mathbf{W}^l$ is a trainable matrix shared among all nodes of layer l. $\sigma(\cdot)$ is the activation function. $\mathbf{h}_{N(v)}^l$ represents the feature aggregated from the neighbors of node v at l-th layer. $COMBINE(\cdot)$ function is used to combine embeddings of itself and neighbors.

Graph Convolutional Networks on Communication Graph. To capture the information for communication and interaction with the neighbor endpoints in the heterogeneous communication graph, a heterogeneous GCN is used. In the GCN-based node classification task of the homogenous graph, the embeddings node in the last propagation layer is used as the input of the classifier. Instead, we use the edge embedding of the last propagation layer and the embeddings of the two vertices connected by this edge. We concatenate the above three embeddings and use them for edge classification tasks. As shown in Fig. 2, where $\mathbf{z}_e, \mathbf{z}_s$ and $\mathbf{z}_d$ denote the edge, source endpoint, and destination endpoint embedding, i.e., $\mathbf{z}_e = \mathbf{h}_e^L, \mathbf{z}_s = \mathbf{h}_{S(e)}^L, \mathbf{z}_d = \mathbf{h}_{D(e)}^L$.

Aggregation Sublayer. The aggregation sublayer of GCN treats all types of nodes equally and ignores the attributes of edges. To fit our Communication Graph, we define three aggregation functions for three kinds of entities (source endpoint, destination endpoint, and flow).

For a flow, i.e., an edge, its hidden state is updated by concatenating the hidden states of the previous edge itself and the two nodes it is connected. Therefore, the aggregation sublayer is defined as:

$$\mathbf{h}_e^l = \sigma(\mathbf{W}_E^l \cdot AGG_E^l(\mathbf{h}_e^{l-1}, \mathbf{h}_{S(e)}^{l-1}, \mathbf{h}_{D(e)}^{l-1})), \tag{5}$$

where

$$AGG_E^l(\mathbf{h}_e^{l-1}, \mathbf{h}_{S(e)}^{l-1}, \mathbf{h}_{D(e)}^{l-1}) = concat(\mathbf{h}_e^{l-1}, \mathbf{h}_{S(e)}^{l-1}, \mathbf{h}_{D(e)}^{l-1}). \tag{6}$$

For the endpoint node $s \in S$ and $d \in D$, in addition to information from neighboring nodes, the attributes of the edges connecting them are also collected. The aggregated neighbor embedding $\mathbf{h}_{N(s)}^l, \mathbf{h}_{N(d)}^l$ calculate as:

$$\begin{aligned} \mathbf{h}_{N(s)}^l &= \sigma(\mathbf{W}_S^l \cdot AGG_S^l(\mathbf{H}_{DE}^{l-1})), \\ \mathbf{h}_{N(d)}^l &= \sigma(\mathbf{W}_D^l \cdot AGG_D^l(\mathbf{H}_{SE}^{l-1})), \end{aligned} \tag{7}$$

where

$$\begin{aligned} \mathbf{H}_{DE}^{l-1} &= \{concat(\mathbf{h}_d^{l-1}, \mathbf{h}_e^{l-1}), \forall e = (s, d) \in E(s)\}, \\ \mathbf{H}_{SE}^{l-1} &= \{concat(\mathbf{h}_s^{l-1}, \mathbf{h}_e^{l-1}), \forall e = (s, d) \in E(d)\}. \end{aligned} \tag{8}$$

The two types of node maintain different parameters $(\mathbf{W}_S^l, \mathbf{W}_D^l)$ separately, and different aggregation functions (AGG_S^l, AGG_D^l).

As for the specific form of AGG_S^l and AGG_D^l, we adapt the attention mechanism:

$$AGG_S^l(\mathbf{H}_{DE}^{l-1}) = ATTN_S(\mathbf{h}_s^{l-1}, \mathbf{H}_{DE}^{l-1}), \tag{9}$$

$$AGG_D^l(\mathbf{H}_{SE}^{l-1}) = ATTN_D(\mathbf{h}_d^{l-1}, \mathbf{H}_{SE}^{l-1}).$$

$ATTN(.)$ is a function $f : \mathbf{h}_{key} \times \mathbf{H}_{val} \rightarrow \mathbf{h}_{val}$, which maps a feature vector $\mathbf{h}_{key}$ and the set of candidates' feature vector $\mathbf{H}_{val}$ to a weighted sum of elements in $\mathbf{H}_{val}$. The weights of the summation, i.e., attention values, are calculated by the scaled dot-production attention [19].

Combination Sublayer. After aggregating neighbor information, we follow a combination strategy as the previous work [20] for the source and destination endpoint nodes as:

$$\mathbf{h}_s^l = concat(\mathbf{V}_S^l \cdot \mathbf{h}_s^{l-1}, \mathbf{h}_{N(s)}^l), \tag{10}$$

$$\mathbf{h}_d^l = concat(\mathbf{V}_D^l \cdot \mathbf{h}_d^{l-1}, \mathbf{h}_{N(d)}^l),$$

where $\mathbf{V}_S^l$ and $\mathbf{V}_D^l$ denote trainable weight matrix for source endpoint S, and destination endpoint node D, the $\mathbf{h}_s^l$ and $\mathbf{h}_d^l$ are source and destination endpoints' hidden states of l-th layer.

4.5 Traffic Classification

The final embeddings of ETC-GCN are the concatenation of the embeddings learned from Communication Graph:

$$y = classifier(concat(\mathbf{z}_s, \mathbf{z}_d, \mathbf{z}_e)), \tag{11}$$

where $\mathbf{z}_s, \mathbf{z}_d$ and $\mathbf{z}_e$ denote the embeddings of $S(e), D(e)$ and e, which is determined through heterogeneous GCN on communication graph, respectively. The forward propagation process of our algorithm shows as Algorithm 1.

Algorithm 1: The forward propagation process of heterogeneous GCN on Communication Graph.

Input: Set of edges $E_b \subset E$, number of layers L, functions $S(E_b), D(E_b)$, which map E_b to the source endpoint nodes and the destination nodes linked, respectively. Communication Graph $G(S, D, E)$.

Output: The predicted label y of the flow.

```
$E^l \leftarrow E_b, S^l \leftarrow S(E_b), D^l \leftarrow D(E_b)$
for $l = L,...,1$ do
    $S^{l-1} \leftarrow S^l, D^{l-1} \leftarrow D^l$
    for $s \in S^l$ do
        $S^{l-1} \leftarrow S^{l-1} \cup N(s)$
    end
    for $d \in D^l$ do
        $D^{l-1} \leftarrow D^{l-1} \cup N(d)$
    end
end
for $l = 1,...,L$ do
    for $e \in E^l$ do
        $\mathbf{h}_e^l \leftarrow \sigma(\mathbf{W}_E^l \cdot AGG_E^l(\mathbf{h}_e^{l-1}, \mathbf{h}_{S(e)}^{l-1}, \mathbf{h}_{D(e)}^{l-1}))$
    end
    for $s \in S^l$ do
        $\mathbf{H}_{SE}^{l-1} \leftarrow \{concat(\mathbf{h}_s^{l-1}, \mathbf{h}_e^{l-1}), \forall e = (s, d) \in E(s)\}$
        $\mathbf{h}_{N(s)}^l = \sigma(\mathbf{W}_S^l \cdot AGG_S^l(\mathbf{H}_{SE}^{l-1}))$
        $\mathbf{h}_s^l = concat(\mathbf{V}_S^l \cdot \mathbf{h}_{S(e)}^{l-1}, \mathbf{h}_{N(s)}^l)$
    end
    for $d \in D^l$ do
        $\mathbf{H}_{DE}^{l-1} \leftarrow \{concat(\mathbf{h}_d^{l-1}, \mathbf{h}_e^{l-1}), \forall e = (s, d) \in E(d)\}$
        $\mathbf{h}_{N(d)}^l = \sigma(\mathbf{W}_D^l \cdot AGG_D^l(\mathbf{H}_{DE}^{l-1}))$
        $\mathbf{h}_d^l = concat(\mathbf{V}_D^l \cdot \mathbf{h}_{D(e)}^{l-1}, \mathbf{h}_{N(d)}^l)$
    end
end
for $e \in E^l$ do
    $\mathbf{z}_e = \mathbf{h}_e^L, \mathbf{z}_s = \mathbf{h}_{S(e)}^L, \mathbf{z}_d = \mathbf{h}_{D(e)}^L$
end
Classification: $y = classifier(concat(\mathbf{z}_s, \mathbf{z}_d, \mathbf{z}_e))$
```

5 Experiments

5.1 Dataset and Metrics

We conduct all experiments on a real-world encrypted dataset called ISCX2016 VPN-NonVPN, containing regular traffic over VPN with 12 categories: chat, email, VPN-chat, VPN-email, etc. However, the original dataset needs to be preprocessed since its unbalanced problem between different types. We randomly select 1000 flows from each type of flow. Table 2 describes the statistics result of the preprocessed dataset. The reason why the total number flows is less than 12,000 is that the sample of some types is less than 1000.

Table 2. Preprocessing results of the ISCX2016 dataset

#Source Endpoint Nodes	#Destination Endpoint Nodes	# Edge/Flow
6,398	4,666	10,599

Instead of labeling the traffic manually, given the prediction and its ground truth, we compute the measure metrics as Eq. 12,

$$\mathrm{ACC} = \frac{TP + TN}{TP + TN + FP + FN},\ P = \frac{TP}{TP + FP},\ \mathrm{R} = \frac{TP}{TP + FN},\ \mathrm{F}_1 = \frac{2PR}{P + R} \tag{12}$$

where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative in the confusion matrix, separately.

5.2 Baselines

We compare our method with baselines as following:

- C4.5 [1]: C4.5 is a machine learning method using flow-based time-related features to make classification.
- Xgboost [21]: Xgboost is an ensemble model based on the gradient boosting decision tree (GBDT) using flow-based time-related features to make classification.
- 1D-CNN [9]: 1D-CNN is an encrypted traffic classification method based on one-dimension CNN, which preprocesses each flow as a 784-pixel image.
- HAST [13]: HAST is a deep model combining with CNN and LSTM to model the spatial and temporal features of traffic data.

5.3 Settings

We implement the proposed model ETC-GCN on the hardware of 16 cores CPU * 2, memory 25G, and NVIDIA P100 GPU and the software of Red Hat 4.8.5 OS, TensorFlow 1.15.0. In our ETC-GCN model, for the heterogeneous GCN, we set the hidden size of both layers and the hidden size of the attention layer to 64. Besides, we set the momentum to 0.5 and the learning rate to 0.001 to optimize our model. For the 1D-CNN and HAST, we set parameters the same as in the papers.

5.4 Results and Analysis

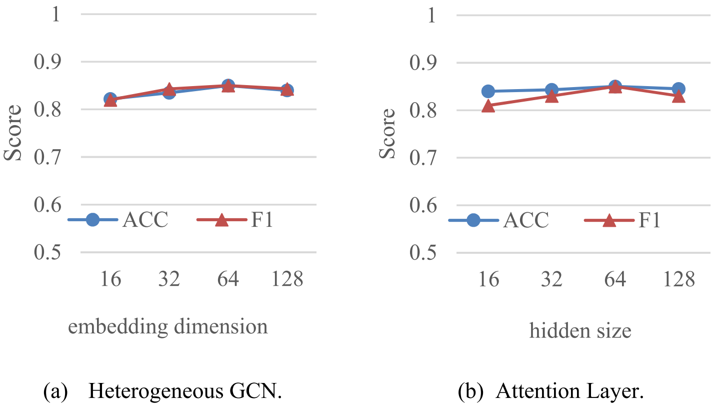
Performance Evaluation. We evaluate the results on the ISCX2016 dataset with the above baselines on different experiment scenarios as Exp 1, Exp 2, Exp 3, and Exp 4 to validate the performance in tasks of traffic identification and characterization. Exp 1 is a binary classification on VPN vs. Encrypted traffic. Exp 2 is a six-class classification on types of chat, email, file, P2P, streaming, and Voice Over Internet Phone (VOIP) of regular encrypted traffic. Exp 3 is another six-class classification on protocol encapsulated traffic. Exp 4 is another twelve-class classification on VPN or encrypted traffic with particular types such as chat, VPN-chat, etc. The results of encrypted traffic classification in the given encrypted traffic show in Table 3. We can see that in Exp 1, ETC-GCN achieves the best results. But in the multi-class classification task, 1D-CNN keeps the best, which means the features exacted from neighbors in the heterogeneous graph bring the noise for classification. For example, the same port will send out all VPN traffic, but the more detailed categories are different, such as chat and email. In summary, this result indicates that incorporating traffic-level characteristics and network-side behaviors contribute to binary classification.

Table 3. Comparison with the state-of-the-art methods

Method	ACC	P	R	F$_1$	ACC	P	R	F$_1$
	Exp 1				Exp 2			
C4.5	0.923	0.924	0.924	0.924	0.726	0.722	0.722	0.725
Xgboost	0.939	0.940	0.940	0.940	0.750*	0.750	**0.750**	0.750*
1D-CNN	0.994	0.994	0.994	0.994	**0.757**	**0.781**	0.729	0.741
HAST	0.999*	0.999*	0.999*	0.999*	0.717	0.717	0.717	0.717
ETC-GCN	**1.000**	**1.000**	**1.000**	**1.000**	0.750*	0.778*	0.732*	**0.754**
	Exp 3				Exp 4			
C4.5	0.841	0.841	0.842	0.841	0.746	0.747	0.747	0.747
Xgboost	0.876	0.877	0.877*	0.877	0.786	0.791	0.786	0.787
1D-CNN	**0.945**	**0.948**	**0.945**	**0.948**	**0.863**	**0.871**	0.837*	0.842*
HAST	0.940*	0.935	**0.945**	0.940	0.814	0.814	0.814	0.814
ETC-GCN	0.940*	0.941*	**0.945**	0.943*	0.850*	0.850*	**0.850**	**0.850**

In Table 3, we bold the best results and superscript * the runner-ups.

Parameter Experiments. In this section, we explore the effects of different parameters. The key parameters include the dimensions of embedding in heterogeneous GCN and the hidden size of the attention layer. All parameter experiments are performed on Exp 4 with ETC-GCN on range 16 to 128. The results are shown in Fig. 3. We can see from Fig. 3(a) that the accuracy and F1-score increase when the size is less than 64, and decrease when it is greater than 64, so the best the dimension of embedding in heterogeneous GCN is 64. For the hidden size of the attention layer in Fig. 3(b), it also rises and starts to fall at 64, so we can conclude that the model gets the best performance when the size is 64.



(a) Heterogeneous GCN.

(b) Attention Layer.

Fig. 3. Performance of different parameters.

6 Conclusions and Future Work

In this paper, we propose an **E**ncrypted **T**raffic **N**etwork using **G**raph **C**onvolutional **N**etwork (**ETC-GCN**) to solve the problem of encrypted traffic classification. ETC-GCN learns the traffic-level characteristic through 1D-CNN and the network-side behaviors by a heterogeneous GCN. And the experimental results prove the effectiveness of our method. In the future, we will extend our model to more traffic classification scenes, such as an IDS system, malware traffic identification.

Acknowledgement. This paper is funded by the State Grid Jibei Power Co. Ltd. Science and Technology Project “Research on Data Fusion and Visualization Key Technologies Based on Full-service Unified Data Center” (52018E18006N).

References

1. Draper-Gil, G., Lashkari, A.H., Mamun, M.S.I., et al.: Characterization of encrypted and VPN traffic using time-related. In: Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP), pp. 407–414 (2016)
2. Rezaei, S., Liu, X.: Deep learning for encrypted traffic classification: an overview. IEEE communications Mag. **57**(5), 76–81 (2019)
3. Bremler-Barr, A., David, S.T., Hay, D., et al.: Decompression-free inspection: DPI for shared dictionary compression over HTTP. In: 2012 Proceedings IEEE INFOCOM, pp. 1987–1995. IEEE (2012)

4. Deri, L., Martinelli, M., Bujlow, T., et al.: nDPI: open-source high-speed deep packet inspection. In: 2014 International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 617–622. IEEE (2014)
5. Alshammari, R., Zincir-Heywood, A.N.: Investigating two different approaches for encrypted traffic classification. In: 2008 Sixth Annual Conference on Privacy, Security and Trust, pp. 156–166. IEEE (2008)
6. Alshammari, R., Zincir-Heywood, A.N.: Machine learning based encrypted traffic classification: identifying SSH and Skype. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–8. IEEE (2009)
7. Dusi, M., Este, A., Gringoli, F., et al.: Using GMM and SVM-based techniques for the classification of SSH-encrypted traffic. In: 2009 IEEE International Conference on Communications, pp. 1–6. IEEE (2009)
8. Vlăduţu, A., Comăneci, D., Dobre, C.: Internet traffic classification based on flows' statistical properties with machine learning. Int. J. Netw. Manage. **27**(3), e1929 (2017)
9. Wang, W., Zhu, M., Wang, J., et al.: End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 43–48. IEEE (2017)
10. Chen, Z., He, K., Li, J., et al.: Seq2Img: a sequence-to-image based approach towards IP traffic classification using convolutional neural networks. In: 2017 IEEE International Conference on Big Data (Big Data), pp. 1271–1276. IEEE (2017)
11. Wu, K., Chen, Z., Li, W.: A novel intrusion detection model for a massive network using convolutional neural networks. IEEE Access **6**, 50850–50859 (2018)
12. Lotfollahi, M., Jafari Siavoshani, M., Shirali Hossein Zade, R., Saberian, M.: Deep packet: a novel approach for encrypted traffic classification using deep learning. Soft. Comput. **24**(3), 1999–2012 (2019). https://doi.org/10.1007/s00500-019-04030-2
13. Wang, W., Sheng, Y., Wang, J., et al.: HAST-IDS: learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE Access **6**, 1792–1806 (2017)
14. Yao, H., Liu, C., Zhang, P., et al.: Identification of encrypted traffic through attention mechanism based long short term memory. IEEE Trans. Big Data (2019)
15. Iliofotou, M., Kim, H., Faloutsos, M., et al.: Graption: A graph-based P2P traffic classification framework for the internet backbone. Comput. Netw. **55**(8), 1909–1920 (2011)
16. Cao, Z., Xiong, G., Zhao, Y., Li, Z., Guo, L.: A survey on encrypted traffic classification. In: Batten, L., Li, G., Niu, W., Warren, M. (eds.) ATIS 2014. CCIS, vol. 490, pp. 73–81. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45670-5_8
17. Aghaei-Foroushani, V., Zincir-Heywood, A.N.: A proxy identifier based on patterns in traffic flows. In: 2015 IEEE 16th International Symposium on High Assurance Systems Engineering, pp. 118–125. IEEE (2015)
18. Moore, A.W., Papagiannaki, K.: Toward the accurate identification of network applications. In: Dovrolis, C. (ed.) PAM 2005. LNCS, vol. 3431, pp. 41–54. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31966-5_4
19. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
20. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems, pp. 1024–1034 (2017)
21. Chen, T., Guestrin, C.: Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
22. Li, A., Qin, Z., Liu, R., et al.: Spam review detection with graph convolutional networks. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 2703–2711 (2019)

Representing EHRs with Temporal Tree and Sequential Pattern Mining for Similarity Computing

Suresh Pokharel(✉), Guido Zuccon(✉), and Yu Li(✉)

The University of Queensland, Brisbane, Australia
{s.pokharel,g.zuccon}@uq.edu.au, yuli@itee.uq.edu.au

Abstract. The ability to rapidly identify at scale patients that are similar based on their electronic health records (EHRs) is fundamental for a number of clinical informatics applications, such as clinical decision support, cohort selection, treatment recommendation, among others.

The effective representation of EHR data is paramount to effective computational similarity methods. Such representation would take into account the complex properties of EHR data including temporality and multivariaty. Of critical importance for this is the modelling of: (i) compound information – multiple medical events for a patient occur in order and may be at the same time, (ii) clinical patterns – frequent common sequential patterns that are associated with specific sequences of clinical events. To model these, in this paper we exploit the recently proposed Temporal Tree technique to capture compound information and we further apply sequential pattern mining (SPM) with gap constraint to discover more complex clinical patterns.

The effectiveness of the proposed EHR representation method is evaluated using a real EHR dataset, MIMIC III, based on two task types within an Intensive Care Unit setting: (i) similar patients retrieval (ii) sepsis prediction and mortality prediction. The empirical results show that representation of EHRs with Temporal Tree and SPM, used in conjunction with traditional similarity measures or more complex embedding methods, delivers significant improvements in effectiveness in the considered tasks.

Keywords: Electronic health records representation · Temporal Tree · Sequential pattern mining · Gap constraint · Patient similarity

1 Introduction

The availability of increasingly larger amounts of electronic health records (EHRs) has motivated the exploration of large scale artificial intelligence, data mining and machine learning methods aimed to provide effective clinical decision support [14,22]. Key to many of these methods is the representation of such patient data available in the form of EHRs. This is prominent for example in

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 220–235, 2020.
https://doi.org/10.1007/978-3-030-65390-3_18

the task of *similarity computing*, i.e. establishing how similar a pair of patients is based on the information recorded in their EHRs [20,21]. The computation of such a similarity requires to take into account treatments, symptoms, laboratory reports, vital signals, among other data types. The task of similarity computing, indeed, is central to other advance applications such as automatic stratification of patients [14], analysis of patient' clinical pathways [13], personalise healthcare [32], and identifying the relationship between diseases and co-morbidities [18]. Because of its central role, in this paper we investigate a new EHR data representation method in the context of similarity computing.

The effective representation of EHR data for similarity computing, however, is challenging due to the complex nature of EHRs. EHRs in fact are multivariate, temporal, heterogeneous, irregular, and sparse. In addition, because of these multivariate and temporal characteristics, many inherent relationships between clinical events are present in EHR data. These relationships take two forms: that of *compound information*, and that of *clinical patterns*.

EHR information is compounded when multiple clinical events appear at the same point in time (or within a short period of time). For example, in Fig. 1a, the following patient measurements are recorded within one hour[1]: <Systolic Blood Pressure(SBP):80>, <Respiratory Rate(RR):18>, <Glasgow Coma Scale(GCS):16> – because these clinical measurements are recorded within the same temporal time unit used for representation (one hour), they form a compound information.

Clinical patterns (CPs) occur when common sequential clinical events emerge over time from the EHR data. For example, Fig. 1a shows the clinical event sequence for SBP as <SBP:80, SBP:78,. . . >.

Clinical diagnoses are often made on the basis of observations at one point in time (compound information) and the trend of similar observations over a period of time (clinical patterns). Thus, the availability of an effective method for jointly modelling compound information and clinical patterns may be crucial for example for differential diagnosis and for reaching a deeper understanding of patient conditions – important factors when considering patient similarity or other advance clinical decision support tasks.

Previous methods have been proposed that address the above challenges, but only partially. Multivariate time series has been used to represent EHR data [6, 8], thus modelling the temporal and multivariate aspects; however this method does not consider the inherent relationships between the clinical events, such as events occurring within a short period of time [26]. Graph mining methods such as subgraph2vec [23] and deep graph [35] can be used to represent compound information through the neighbourhoods of a node but they do not allow for the representation of temporal information. Sequential Pattern Mining (SPM) methods [34,36] or SPM with gap constraint [24] are useful to discover clinical pattens, however they only use univariate data and cannot model compound information. Recently, the Temporal Tree technique [26] has been proposed to

[1] Assume one hour is the temporal time unit used for representation.

capture compound information in EHRs; however Temporal Tree cannot identify and represent clinical patterns.

To overcome the limitations of existing methods and address the identified challenges, in this paper we propose a novel method that builds upon the recent Temporal Tree technique by integrating Sequential Pattern Mining. The proposed method works as follows: First, the Temporal Tree representation is used for capturing compound information. Then, sequences of clinical events are generated from the Temporal Tree representation. Subsequently, SPM with gap constraint is applied for discovering the complex clinical patterns. In this process, clinical patterns are generated not only from sequences of univariate observations (e.g., observations of SBP over a period of time) but also, with the help of Temporal Tree, from sequences of compound information captured across multiple levels of the hierarchical structure (e.g., combined observations of SBP, RR, GCS over a period of time). By doing so, the discovered clinical patterns encode complex relationships between clinical events due to the ability to capture multiple observations at the same time as well as frequent patterns over a period of time. We evaluate the proposed Temporal Tree with Sequential Pattern Mining for EHR representation across multiple clinical tasks where similarity computing is paramount, and consider an array of similarity measures as representative instantiations of traditional methods (Jaccard, overlap, weighted-cosine) and state-of-the-art embedding techniques (pv-dbow, pv-dm, soft-cosine).

This paper puts forward the following contributions:

1. A novel method for representing EHR data that captures complex inherent relationships between clinical events. This method is based on the Temporal Tree technique and Sequential Pattern Mining with gap constraint.
2. An evaluation of the proposed representation method on clinical tasks where effective similarity computing is paramount: (i) similar patient retrieval (ii) sepsis prediction and patient mortality prediction.
3. A comparison of the effectiveness of the proposed representation method against state-of-the-art methods on real ICU data, showing that the proposed method provides significant improvements in effectiveness in the considered evaluation tasks.

2 Related Work

Patient Similarity. The problem of computationally establishing how similar two patients are based on their EHRs has been explored in a number of previous studies. For example, Sun et al. [31] used locally supervised metric learning to compute a patient similarity matrix. Miotto et al. [20] used unsupervised deep feature learning to derive a general purpose patient representation. Jia et al. [15] used diagnoses sets and converted the multi-label classification problem into a single-value regression problem to identify similar patients. A common drawback of these methods, however, is that they do not consider the inherent relationships between clinical events when computing similarity.

Other methods do tackle the problem of representing inherent relationships between clinical events. For example Wang et al. [33] first derived dynamic Bayesian networks (DBNs) from the EHRs for finding the correlation among variables, and then exploited the DBNs within a recurrent neural network architecture to generate a representation of each patient. These sequences were utilized to learn patient embeddings using *med2vec* [9]. However, this method does not consider compound information and is characterised by an overwhelming amount of parameters, rendering the learning process difficult and lengthy.

Patient Embeddings. Embedding techniques have been exploited to represent EHR data into lower dimensional vectors, where similar patients would be represented by similar embeddings. Zhang et al. [37], Choi et al. [10], and Glicksberg et al. [11] used *word2vec* [19] to construct a lower dimensional embedding, while Bajor et al. [5] used the document-level embedding approach [17] (also known as *doc2vec*). However, these methods do not explicitly model the inherent relationships between clinical events. The method of Pokharel et al. [26], Temporal Tree, which is at the basis of the method put forward in this paper, models compound information for representing a patient into an embedding; however it does not model clinical patterns.

Sequential Pattern Mining on EHRs. SPM [1] has found wide application for discovering frequent patterns from the EHR data. Wright et al. [34] used SPM to identify temporal relationships between drugs; these relationships were then exploited to predict the next prescribed medications. Similarly, Rjeily et al. [27] applied SPM, specifically Compact Prediction Tree plus (CPT+), for identifying heart failure patients. These previous examples, however, rely on SPM using univariate data only. In addition, they do not use the gap constraint in their SPM, which is important to identify clinical events that occur in close time proximity: these events are in fact likely to be more meaningful than distantly occurring events. Batal et al. [7] used pattern mining on multivariate temporal data for identifying patients who can have potential risk of heparin-induced thrombocytopenia. However they did not consider compound information and gap constraint.

3 Temporal Tree with Sequential Pattern Mining

In this paper we investigate the application of the Temporal Tree technique [26] to capture compound information and further propose to extend this method using SPM with gap constraint with the aim of discover and model more complex clinical patterns. Next, we detail the Temporal Tree technique (Sect. 3.1) and subsequently the use of SPM to generate clinical patterns (Sect. 3.2).

3.1 Temporal Tree

A Temporal Tree [26] is a temporal hierarchical structural network which is constructed based on the temporal co-occurrence of clinical events, and it allows us to represent the compound information present in EHR data. A Temporal

Tree is constructed for each patient. An example of a simple Temporal Tree is shown in Fig. 1b. Each branch from a root node of a $SubTree$ represents an event type such as laboratory events, prescriptions, etc. In this paper, we use a single event type, i.e., quick Sequential Organ Failure(qSOFA) variables (see Sect. 4.2 for more details), but multiple event types are possible.

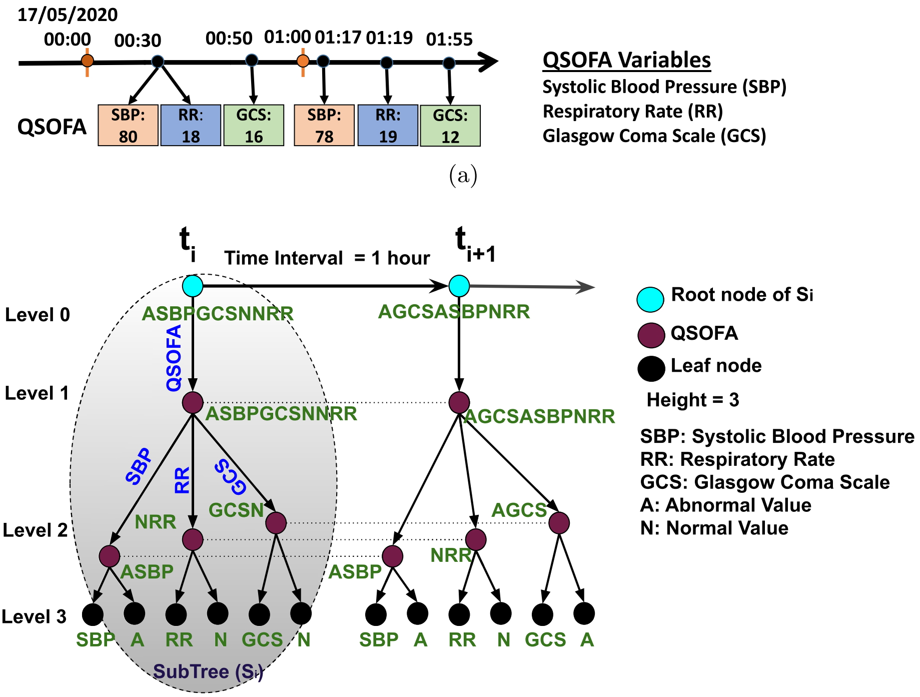
Compound information in the Temporal Tree is generated based on the local neighbourhood relationships between clinical events and is represented in a hierarchical form. Here, the leaf nodes represent the actual clinical events that appear at the respective timestamps and non-leaf nodes represent the compound information which is generated by the relabelling process. For relabelling, the Weisfeiler-Lehman graph kernels re-labelling method [29] is used. Note that during the relabelling process, the label of a parent node is generated from its children nodes by sorting them first and then concatenating them e.g., in Fig. 1b $GCSN$ is generated from GCN and N rather than $NGCS$.

Generation of Compound Information Sequences. Compound information sequences are generated from a Temporal Tree by Breath First Search (BFS) traversal. To avoid the unmeaningful labels and to capture the clinical patterns for each variables as well as compound information separately, we modify the original formulation of Temporal Tree [26] when generating a clinical sequence as follows. (1) We only consider qSOFA variables, hence we have only one branch from the root node of level 0. As a result, the labels of level 0 and level 1 are the same and thus the labels of level 0 are ignored when generating the sequence. Similarly, we also ignore the level 3 because if we generate the sequences from level 3, then they only contain the repetition of the same level (e.g., SBP, RR, GCS, A, N) which is not meaningful for distinguishing patients. (2) We generate many sequences from level 1 and level 2 as indicated by the horizontal doted line in Fig. 1b – this is unlike in the original Temporal Tree where one sequence for each level was generated. For example, in Fig. 1b we generate three different sequences from level 2 (for each variable: SBP, RR, GCS). Then, for each patient, all the generated sequences are concatenated to form a single clinical event sequence.

3.2 Clinical Patterns

Clinical Sequence. Each patient is considered as a sequence of compound information as described in Sect. 3.1. Formally, let Σ be a set of symbols (compound information) and $|\Sigma|$ denote its cardinality. A clinical sequence S is defined as a temporally ordered list of clinical events and is written as $S = \{e_1, e_2, \ldots, e_l\}$ where $e_i \epsilon \Sigma$ is the symbol at position i. $D = \{S_1, S_2, \ldots, S_N\}$ is a dataset of N sequences.

Subsequence. Let $S_1 = \{e_1, e_2, \ldots, e_m\}$ and $S_2 = \{\acute{e}_1, \acute{e}_2, \ldots, \acute{e}_n\}$ be two sequences over Σ. Then, S_1 is a subsequence of S_2 (denoted by $S_1 \subseteq S_2$ and also referred to as S_2 contains S_1) if there exists a one-to-one mapping $\phi : [1, m] \rightarrow [1, n]$, such that $S_1[i] = S_2[\phi(i)]$ and for any two positions i, j in S_1, $i < j \Rightarrow \phi(i) < \phi(j)$ [36].



(a)

(b)

Fig. 1. (a) An example of EHR data for an ICU patient (b) Temporal Tree representation, where qSOFA are shown as example events. The horizontal dotted line is not part of Temporal Tree: it instead indicates the traversal strategy being used.

General SPM instantiations are set to discover all present patterns without modelling the gap between the symbols. We argue however that in the case of clinical event sequences, this gap does matter. During a patient's stay in ICU, for example, a patient condition is very unstable and one of the key goals of an ICU doctor is to bring the patient to a stable condition. Treatments are thus provided based on the immediate condition of a patient being observed; further observations are made thereof, adjusting or changing the treatment regime. Thus, the closer two clinical events are (close gap), the more meaningful and strong their relationship is. To model this, we consider sequential patterns under gap constraint satisfaction.

Gap Constraint. A gap (denoted by Δ) is a positive integer, $\Delta > 0$. Let a clinical sequence be $S = \{\acute{e}_1, \acute{e}_2, \ldots, \acute{e}_n\}$ and an occurrence $o = \{i_1, i2, \ldots, i_n\}$ of a subsequence X of S. If $i_{k+1} \leq i_k + \Delta$ $(\forall i_k \epsilon [i, n-1])$, then o satisfies the Δ-gap constraint. If there is at least one occurrence o of X that satisfies the Δ-gap constraint, then X satisfies the Δ-gap constraint [24].

SPM with Gap Constraint. Given a clinical sequential dataset D, a gap constraint Δ and a minimum support threshold (denoted by $\delta \in [0, 1]$), sequential

Compound Information	Symbol
ASBP	a
NSBP	g
NRR	b
ARR	h
GCSN	c
AGCS	d
ASBPGCSNNRR	e
AGCSASBPNRR	f
ARRASBPGCSN	i
AGCNNRRNSBP	j
AGCSARRNSBP	k

(a)

Patient	Symbols
P1	{a, a, b, b, c, d}
P2	{a, g, h, b, c, d}
P3	{g, a, h, b, d, d}

(b)

Patient	Symbols
P1	{a, a, b, b, c, d, e, f}
P2	{a, g, h, b, c, d, i, j}
P3	{g, a, h, b, d, d, k, f}

(c)

CP	Symbols	Support
X1	{a}	1.00
X2	{b}	1.00
X3	{c}	0.67
X4	{d}	1.00
X5	{f}	0.67
X6	{g}	0.67
X7	{h}	0.67
X8	{b, c}	0.67
X9	{c, d}	0.67
X10	{h, b}	0.67

(d)

Patient	CPs
P1	{X1, X2, X3, X4, X5, X8, X9}
P2	{X1, X2, X3, X4, X6, X7, X8, X9, X10}
P3	{X1, X2, X4, X5, X6, X7, X10}

(e)

Fig. 2. (a) An example of compound information; (b) sequence of clinical events without applying Temporal Tree; (c) clinical sequence generated by Temporal Tree using BFS traversal; (d) from table c, ten clinical patterns are discovered using SPM with gap constraint, given that $\varDelta = 1$ and $\delta = 0.6$; (e) sequence of clinical patterns for each patient.

pattern discovery deals with finding all the subsequences(X), along with their corresponding supports (σ), such that $\sigma(X, \varDelta) \geq \delta$.

Example 1. Figure 2 shows an example of generating the clinical pattens for each patient using Temporal Tree and SPM with gap constraint.

Generating the clinical patterns has the following advantages: (1) they capture the inherent complex temporal and multivariate relationships between clinical events. For example, $X9$ is a pattern of a patient having *normal GCS* and then *abnormal GCS*. In real data, we observe a large number of more complex patterns than the example $X9$. (2) They increase the accuracy of the representation. (3) By considering both singleton clinical events and clinical patterns, we can increase the vocabulary size which results in better feature representation.

Example 2. Listing 1.1 shows an example of discovered clinical patterns using a real dataset. Here, $X101$ represents that a patient has *normal GCS* and *RR*, but *abnormal SBP* at the same time. Likewise $X974$ represents a clinical pattern of the type: *abnormal RR*, *abnormal RR*, *abnormal RR*, *normal RR*.

Listing 1.1. Examples of CPs. 0:normal and 1:abnormal. ^ represents occuring of respective events at the same time.

```
X101: {0_gcs^0_rr^1_sbp}
X974: {1_rr, 1_rr, 1_rr, 1_rr, 0_rr}
X4773: {0_gcs^0_rr^0_sbp, 0_gcs^0_rr^0_sbp, ..., 0_gcs^0_rr^1_sbp}
```

4 Evaluation Methodology

4.1 Dataset and Patient Cohort Selection

We use a publicly available de-identified real ICU dataset, MIMIC III [16], to evaluate the proposed approach. We consider each ICU admission as referring to a unique patient. Patients were selected according to the following criteria: (i) adults (patients aged 16 years or more), (ii) have at least one value recorded for each qSOFA variable (see Sect. 4.2 for more details), (iii) have been admitted to ICU for the first time – re-admitted patients are excluded because it is likely a patient is re-admitted for the same condition, and thus the data would show a high correlation, (iv) top-3 most frequent first[2] diagnoses only. This is because the use of all diagnoses available would result in a largely sparse similarity matrix to be used for evaluation. To avoid this, we filter the patients using such a criteria. Note that this is not a limitation of the proposed method, but an empirical setting chosen to maintain reliability in the evaluation. The filtered subset contains a total of $5,274$ patients. Note the dataset presents a bias towards survival patients (did not die during hospital stay); mortality: 642, survival: $4,632$. Similarly, the dataset contains $1,783$ patients that have developed sepsis and $3,491$ with no sepsis.

4.2 Features Selection

The features are selected based on the qSOFA criteria. The qSOFA score [4] is a simplified version of Sequential Organ Failure Assessment (SOFA) Score as an initial way to assess patients at high risk of poor outcome with respect to infection/sepsis. The advantage of qSOFA is its simplicity compared to SOFA which requires numerous lab tests, more time and is more expensive. qSOFA can be repeated serially and it can be applied outside the ICU setting as an initial way to identify patients at risk. qSOFA uses three variables to test the abnormality of organs according to the following criteria: Low Blood Pressure (SBP $\leq$100 mmHg), High Respiratory Rate (RR $\geq$ 22 breaths/min) and Altered Mentation (GCS $\leq$ 13) [28].

[2] Each patient may have multiple diagnoses: we only consider the first diagnosis when filtering the data to create the subset for evaluation. The used primary icd9_code are: "41401","0389" and "51881".

Table 1. Similarity computation methods considered by our empirical evaluation.

Methods	Formula	Remarks
Jaccard	$\frac{\lvert X \cap Y\rvert}{\lvert X\rvert + \lvert Y\rvert - \lvert X \cap Y\rvert}$	Given X and Y are the two lists
Overlap	$\frac{\lvert X \cap Y\rvert}{min(\lvert X\rvert, \lvert Y\rvert)}$	Given X and Y are the two lists
Weighted-cosine	$\frac{\sum_i^N X_i Y_i}{\sqrt{\sum_i^N X_i^2}\sqrt{\sum_i^N Y_i^2}}$	Given the N dimensional vectors X and Y
Embedding [17]	$\frac{\sum_i^N X_i Y_i}{\sqrt{\sum_i^N X_i^2}\sqrt{\sum_i^N Y_i^2}}$	Given X and Y are low dimensional vectors generated by using embedding technique [17] (either pv-dbow or pv-dm)
Soft-cosine [30]	$\frac{\sum_{i,j}^N s_{ij} X_i Y_j}{\sqrt{\sum_{i,j}^N s_{ij} X_i X_j}\sqrt{\sum_{i,j}^N s_{ij} Y_i Y_j}}$	Given the N dimensional vectors X and Y where $s_{i,j} = similarity(feature_i, feature_j)$ $s_{i,j}$ is calculated by using pv-dbow
Optimized Soft-cosine	$\frac{\sum_i^L \sum_j^M s_{ij} X_i Y_j}{\sqrt{\sum_{i,j}^L s_{ij} X_i X_j}\sqrt{\sum_{i,j}^M s_{ij} Y_i Y_j}}$	Where L,M are the unique features of X and Y. Since, $L, M << N$ then time reduces from $O(N^2)$ to $O(LM)$.

We set the time intervals to one hour and if more than one event is found within an interval, we take the average value of the events. Further, missing data is restored using linear interpolation. Numeric values are converted into categorical values (A: $Abnormal$, N: $Normal$) using the qSOFA criteria. We also set time intervals for Temporal Tree to one hour because the clinical events that occur in close temporal proximity often have a stronger relationship than events that occur far apart (at least during an admission in ICU).

4.3 Baselines

In the experiments, we apply the proposed representation to both traditional (jaccard, overlap, weighted-cosine) as well as state-of-the-art embedding based similarity methods (pv-dbow, pv-dm, soft-cosine), see Table 1.

- *Jaccard:* Similarity is computed based on the number of common clinical patterns shared by two patients (each patient is represented as a set of clinical patterns) over the size of the union for two patients.
- *Overlap:* Similarity is computed based on number of common clinical patterns shared by two patients (each patient is represented as a set of clinical patterns) over the size of the smaller set of the two patients.
- *Weighted-cosine:* The clinical patterns are weighted according to term frequency (tf)-inverse document frequency(idf). Then, the similarity is calculated based on the cosine angle between the two patient vectors.
- *PV-DM* [17]*:* State-of-the-art embedding based method where concatenation (or average) of a sequence vector along with surrounding CPs is used to predict a target CP. The sequence vector represents the missing information from the current context and can act as a memory of the topic (in the original paper, topic refers to topic of a paragraph) of the sequence of CPs. Note that each patient is represented as a sequence of CPs as described in Sect. 3.2.
- *PV-DBOW* [17]*:* State-of-the-art embedding based method where the sequence vector is trained to predict the CPs in a small window. Unlike the

PV-DM, this model ignores the surrounding CPs in the input, but force the model to predict CPs which are randomly sampled from the sequence of CPs in the output. Note that each patient is represented as a sequence of CPs as described in Sect. 3.2.
- *Soft-cosine* [30]*:* The tf-idf schema is used for assigning weights to clinical patterns and pv-dbow is used for calculating the similarity between clinical patterns. We have used the optimized soft-cosine formula as shown in Table 1 to reduce computational complexity.

As for comparison, we consider the following EHRs representations (i) raw features - clinical events without considering Temporal Tree and the modelling of clinical patterns (ii) Temporal Tree [26] - originally proposed without modelling of clinical patterns (iii) Temporal Tree with SPM - the proposed representation method in this paper where Temporal Tree with modelling of clinical patterns are considered. We apply all the above measures to these three considered representations.

4.4 Evaluation Tasks

The effectiveness of the proposed representation method is evaluated in two contexts: (i) similar patients retrieval (ii) prediction models.

Similar Patients Retrieval. We cast the similarity computing problem into an information retrieval problem where the task is to retrieve the patients that are similar to a query patient. For this, we follow the previous similar works by Gottlieb et al. [12] and Pokharel et al. [26]. Similar to them, we use the International Classification of Diseases, Ninth Revision (ICD-9) codes as gold standard to measure patient similarity. The gold standard similarity between two patients is calculated based on the number of diagnoses shared along with the respective ranking of diagnoses. This is achieve by using the *SimIndex* function [25]. For a query patient, the ranking of similar patients retrieved by the system is evaluated in terms of the following parameters: (1) *Mean Square Error (MSE):* computes the error made by the system when retrieving similar patients, compared to the gold standard. (2) *Normalized Discounted Cumulative Gain (nDCG):* Discounted Cumulative Gain (DCG) computes a weighted sum of the degree of relevancy while ranking the retrieved similar patients by the system. And the nDCG is DCG normalized by the ideal DCG - in our case, ideal DCG is DCG measure from patient similarity matrix which is obtained by using gold standard. (3) *Precision:* we follow the work of Gottlieb et al. [12] where they only consider the top two diagnoses (highest priority) and don't use *simIndex* function. So, relevance of the query patient is defined as follows: if a retrieved patient contains any of the two diagnosis of the query patient, then the patient is considered as relevant to the query patient. We restrict the number of retrieved patients for a given patient query to $k = 1, 5, 10, 20$.

Prediction Models. Intuitively, similar patients are likely to exhibit similar mortality and sepsis risks – thus effective representation methods would exhibit similar features for similar patients and thus be effective for prediction tasks such as (i) sepsis prediction (ii) in ICU patient mortality prediction. We use these two important ICU tasks as a down-stream application of the proposed EHR data representation method.

For sepsis prediction, we consider the sepsis information (sepsis or not-sepsis) for a patient as the class labels, thus becoming a binary classification problem. For obtaining the sepsis information, we follow the work of Angus et al. [3] which is common practice in hospitals for sepsis patient identification.

For mortality prediction, we consider the mortality information (survive or not-survive) for a patient at the end of their ICU stay as the class labels, thus becoming a binary classification problem.

For both prediction tasks, we use k-Nearest Neighbourhood (kNN) for classification as it is an intuitive similarity-based approach that can directly rely on the representations studied in this paper. We evaluate the classification effectiveness according to $f1\_micro$, $f1\_macro$ and Area Under the Receiver Operating Characteristic Curve (AUC). We apply 5-folds cross validation: in each fold (training:four portions, testing:one portion), the training dataset is further divided into sub-training and sub-validation dataset with 80:20 ratio to determine the value of k (which gives the maximum accuracy); k is varied in the range [0,20] with step 1; thus obtaining k is used for evaluation by using training and testing dataset. The whole process is repeated five times and the effectiveness is averaged to weed out bias due to the random partition of the training data.

5 Analysis of Empirical Results

5.1 Experimental Criteria Setup

Three main parameters govern our experiments: the minimum threshold (δ), the gap constraint ($\varDelta$) for discovering clinical patterns, and the number of embedding dimensions (ed). We set $\delta = 0.05$ which is a common value for this parameter, $\varDelta = 2$ because in the case of ICU, the clinical events that are close to each other in time are more meaningful than others, and $ed = 50$ following Altszyler et al. [2] who suggested 50 dimensions are appropriate for a medium-size dataset like the one we consider.

5.2 Similar Patients Retrieval Task

Tables 2, 3 and 4 report the performance of different methods for similar patient retrieval in terms of $nDCG$, $Precision$ and MSE. Note that all differences between methods are statistically significant (t-test with Bonferroni correction). In the rest of the paper, the suffixes indicate the following strategies: $\_raw$: without use of Temporal Tree and clinical patterns, $\_tt$: with use of Temporal Tree only, $\_tt\_spm$: with use of Temporal Tree with clinical patterns.

Table 2. *nDCG*

Methods	k = 1	k = 5	k = 10	k = 20
jaccard_raw	0.242	0.306	0.323	0.360
jaccard_tt	*0.454*	0.399	0.418	0.411
jaccard_tt_spm	0.446**	***0.455*****	***0.460*****	***0.467*****
overlap_raw	0.178	0.261	0.249	0.304
overlap_tt	0.178	0.269	0.277	0.325
overlap_tt_spm	***0.481*****	*0.437***	*0.426***	*0.420***
wt-cosine_raw	0.425	0.413	0.412	*0.415*
wt-cosine_tt	0.418	0.411	0.411	0.414
wt-cosine_tt_spm	*0.447***	*0.452***	*0.455***	*0.461***
pv-dbow_raw	0.393	0.397	0.401	0.407
pv-dbow_tt	0.403	0.406	0.409	0.414
pv-dbow_tt_spm	*0.439***	*0.445***	*0.449***	*0.455***
pv-dm_raw	0.359	0.370	0.377	0.387
pv-dm_tt	*0.392*	0.394	0.398	0.405
pv-dm_tt_spm	0.391**	*0.397***	*0.402***	*0.408***
soft-cosine_raw	*0.476*	*0.450*	*0.443*	*0.438*
soft-cosine_tt	0.474	0.448	0.438	0.434
soft-cosine_tt_spm	0.431**	0.432**	0.434*	0.435†

Table 3. *precision*

Methods	k = 1	k = 5	k = 10	k = 20
jaccard_raw	0.371	0.460	0.487	0.552
jaccard_tt	0.693	0.569	0.612	0.582
jaccard_tt_spm	***0.693*****	***0.689*****	***0.685*****	***0.679*****
overlap_raw	0.277	0.407	0.362	0.472
overlap_tt	0.276	0.408	0.421	0.506
overlap_tt_spm	*0.638***	*0.598***	*0.588***	*0.580***
wt-cosine_raw	0.618	0.603	0.600	0.595
wt-cosine_tt	0.615	0.604	0.600	0.594
wt-cosine_tt_spm	*0.691***	*0.683***	*0.677***	*0.669***
pv-dbow_raw	0.614	0.604	0.598	0.593
pv-dbow_tt	0.627	0.616	0.608	0.602
pv-dbow_tt_spm	*0.681***	*0.674***	*0.667***	*0.661***
pv-dm_raw	0.561	0.565	0.566	0.566
pv-dm_tt	0.608	0.597	0.592	0.589
pv-dm_tt_spm	*0.611***	*0.605***	*0.602***	*0.596***
soft-cosine_raw	0.657	0.624	0.614	0.603
soft-cosine_tt	0.657	0.624	0.610	0.600
soft-cosine_tt_spm	*0.668*†	*0.653***	*0.643***	*0.630***

Table 4. *MSE*

Methods	k = 1	k = 5	k = 10	k = 20
jaccard_raw	0.546	0.142	0.085	*0.052*
jaccard_tt	***0.308***	0.129	0.094	0.054
jaccard_tt_spm	0.315**	*0.121***	*0.078**	*0.052*†
overlap_raw	0.621	0.157	0.076	0.050
overlap_tt	0.619	0.150	0.083	0.053
overlap_tt_spm	*0.346***	*0.114***	*0.073*†	*0.048*†
wt-cosine_raw	0.363	0.133	0.084	*0.052*
wt-cosine_tt	0.365	0.137	0.087	0.054
wt-cosine_tt_spm	*0.316***	*0.120***	*0.080***	*0.052*†
pv-dbow_raw	0.370	0.143	0.091	0.058
pv-dbow_tt	0.362	0.141	0.090	0.056
pv-dbow_tt_spm	*0.324***	*0.125***	*0.083*	*0.052***
pv-dm_raw	0.410	0.146	0.089	0.057
pv-dm_tt	0.374	*0.137*	*0.087*	*0.056***
pv-dm_tt_spm	*0.371***	0.166**	0.108	0.066
soft-cosine_raw	0.334	0.110	***0.061***	***0.036***
soft-cosine_tt	*0.332*	***0.107***	0.066	0.040
soft-cosine_tt_spm	0.334†	0.125**	0.081†	0.052**

Effectiveness in terms of *nDCG*, *Precision* and *MSE* for the considered patient similarity approaches. **, *, † indicates statistical significance difference with $p < 0.01$, $p < 0.05$, $p > 0.05$ obtained when comparing the similarity method with and without *tt_spm* strategy.

Table 5. Sepsis prediction

Model	F1_micro	F1_macro	AUC
jaccard_raw	0.724 ± 0.01	0.674 ± 0.023	0.672 ± 0.018
jaccard_tt	0.759 ± 0.003	0.702 ± 0.002	0.690 ± 0.002
jaccard_tt_spm	*0.810 ± 0.002*	*0.776 ± 0.002*	*0.765 ± 0.002*
overlap_raw	0.680 ± 0.008	0.518 ± 0.069	0.566 ± 0.043
overlap_tt	*0.688 ± 0.009*	0.525 ± 0.046	0.561 ± 0.025
overlap_tt_spm	0.626 ± 0.004	*0.609 ± 0.004*	*0.624 ± 0.005*
wt_cosine_raw	0.736 ± 0.005	0.677 ± 0.006	0.668 ± 0.005
wt_cosine_tt	0.740 ± 0.005	0.680 ± 0.005	0.670 ± 0.004
wt_cosine_tt_spm	***0.813 ± 0.002***	***0.782 ± 0.002***	***0.772 ± 0.002***
pv-dbow_raw	0.732 ± 0.003	0.664 ± 0.002	0.656 ± 0.001
pv-dbow_tt	0.751 ± 0.003	0.685 ± 0.004	0.674 ± 0.003
pv-dbow_tt_spm	*0.803 ± 0.003*	*0.769 ± 0.003*	*0.758 ± 0.002*
pv-dm_raw	0.732 ± 0.002	0.700 ± 0.003	0.700 ± 0.003
pv-dm_tt	*0.740 ± 0.003*	*0.706 ± 0.003*	*0.704 ± 0.003*
pv-dm_tt_spm	0.722 ± 0.003	0.620 ± 0.004	0.619 ± 0.003
soft_cosine_raw	0.737 ± 0.004	0.677 ± 0.005	0.668 ± 0.004
soft_cosine_tt	0.741 ± 0.001	0.681 ± 0.001	0.671 ± 0.001
soft_cosine_tt_spm	*0.776 ± 0.003*	*0.755 ± 0.003*	*0.762 ± 0.003*

Table 6. Mortality prediction

Model	F1_micro	F1_macro	AUC
jaccard_raw	0.888 ± 0.002	0.581 ± 0.014	0.563 ± 0.009
jaccard_tt	0.891 ± 0.001	0.639 ± 0.014	0.607 ± 0.013
jaccard_tt_spm	***0.906 ± 0.002***	*0.713 ± 0.004*	*0.669 ± 0.004*
overlap_raw	0.879 ± 0.001	0.483 ± 0.022	0.508 ± 0.011
overlap_tt	0.881 ± 0.002	0.514 ± 0.022	0.525 ± 0.012
overlap_tt_spm	*0.883 ± 0.002*	*0.654 ± 0.009*	*0.626 ± 0.009*
wt_cosine	0.899 ± 0.001	0.697 ± 0.001	0.658 ± 0.002
wt_cosine_tt	0.902 ± 0.001	0.698 ± 0.003	0.657 ± 0.003
wt_cosine_tt_spm	*0.904 ± 0.002*	*0.713 ± 0.008*	*0.672 ± 0.009*
pv-dbow_raw	0.89 ± 0.002	0.675 ± 0.003	0.643 ± 0.002
pv-dbow_tt	0.893 ± 0.001	0.679 ± 0.007	0.644 ± 0.007
pv-dbow_tt_spm	*0.903 ± 0.001*	***0.723 ± 0.004***	***0.687 ± 0.003***
pv-dm_raw	0.874 ± 0.002	0.634 ± 0.006	0.612 ± 0.005
pv-dm_tt	*0.886 ± 0.002*	*0.636 ± 0.008*	*0.607 ± 0.005*
pv-dm_tt_spm	*0.888 ± 0.001*	*0.579 ± 0.008*	0.562 ± 0.005
soft_cosine_raw	0.899 ± 0.002	0.696 ± 0.006	0.657 ± 0.006
soft_cosine_tt	0.903 ± 0.002	0.701 ± 0.007	0.659 ± 0.007
soft_cosine_tt_spm	*0.893 ± 0.001*	*0.634 ± 0.006*	*0.602 ± 0.005*

Effectiveness measure in terms of $f1\_micro$, $f1\_macro$, AUC for different prediction tasks. Standard deviation is provided and represents the variation obtained across different rounds of tuning of the learnt classifier.

The following observations can be made based on the empirical results:

1. In general, all similarity methods obtain higher effectiveness when representing EHRs with Temporal Tree and even better effectiveness when using in addition SPM.
2. Surprisingly, jaccard performs better than embedding based methods when EHRs are represented with Temporal Tree and SPM.
3. The soft-cosine method gives mixed result when using Temporal Tree and SPM.
4. The best effectiveness is obtained by jaccard (in terms of $nDCG$ for $k = 5, 10, 20$, $precision$ for $k = 1, 5, 10, 20$) and overlap (in terms of in terms of $nDCG$ for $k = 1$) when applying Temporal Tree and SPM. In the case of MSE, the best effectiveness is obtained by jaccard (for $k = 1$) and soft-cosine (for $k = 5$) with Temporal Tree only and soft-cosine with raw features (for $k = 10, 20$).
5. Overall, Temporal Tree with SPM is found to be highly effective for representing EHRs (note, for MSE, the lower the better).

5.3 Prediction Tasks

Tables 5 and 6 reports the effectiveness of prediction methods in terms of $f1\_micro$, $f1\_macro$ and AUC for sepsis prediction and mortality prediction, respectively. From the results, the following observations can be made:

1. The most effective methods have been those that used when Temporal Tree and Temporal Tree with SPM for representing EHRs.
2. pv-dm, an embedding based method, gives mixed results when using Temporal Tree with SPM.
3. Weighted cosine is the most effective method for sepsis prediction, while jaccard (for $f1\_micro$) and pv-dbow (for $f1\_macro$ and AUC) are the most effective for the task of mortality prediction. All these methods perform when Temporal Tree with SPM is used to represent EHRs.

6 Conclusion and Future Work

In this paper we introduce a novel method for representing patient EHR data based on Temporal Tree with gap constraint with sequential pattern mining. The ability to holistically represent EHR data is paramount to effective patient similarity computation, which forms the basis of many methods in clinical decision support. Key to our method is the modelling of complex clinical patterns which exist within EHRs.

To demonstrate the proposed method, we perform an empirical evaluation that exploits our method within traditional as well as recent embedding based techniques for patient similarity computation. The empirical results show that the proposed method, Temporal Tree with sequential pattern mining with gap constraint, is an effective representation to be exploited when computing patient similarity from EHR. In future work, we plan to extend our method by investigating its capabilities in making the representation and the similarity computation explainable so as to increase clinicians' understanding of the results produced by the computational methods.

References

1. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings of the 11th International Conference on Data Engineering, pp. 3–14. IEEE (1995)
2. Altszyler, E., Ribeiro, S., Sigman, M., Slezak, D.F.: The interpretation of dream meaning: resolving ambiguity using latent semantic analysis in a small corpus of text. Conscious. Cogn. **56**, 178–187 (2017)
3. Angus, D.C., Linde-Zwirble, W.T., Lidicker, J., Clermont, G., Carcillo, J., Pinsky, M.R.: Epidemiology of severe sepsis in the united states: analysis of incidence, outcome, and associated costs of care. Crit. Care Med. **29**(7), 1303–1310 (2001). Society of Critical Care Medicine
4. Angus, D.C., et al.: A framework for the development and interpretation of different sepsis definitions and clinical criteria. Crit. Care Med. **44**(3), e113 (2016)
5. Bajor, J.M., Mesa, D.A., Osterman, T.J., Lasko, T.A.: Embedding complexity in the data representation instead of in the model: A case study using heterogeneous medical data. arXiv preprint arXiv:1802.04233 (2018)
6. Batal, I., Fradkin, D., Harrison, J., Moerchen, F., Hauskrecht, M.: Mining recent temporal patterns for event detection in multivariate time series data. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 280–288 (2012)

7. Batal, I., Valizadegan, H., Cooper, G.F., Hauskrecht, M.: A pattern mining approach for classifying multivariate temporal data. In: 2011 IEEE International Conference on Bioinformatics and Biomedicine, pp. 358–365. IEEE (2011)
8. Batal, I., Valizadegan, H., Cooper, G.F., Hauskrecht, M.: A temporal pattern mining approach for classifying electronic health record data. ACM Trans. Intell. Syst. Technol. (TIST) **4**(4), 63 (2013)
9. Choi, E., et al.: Multi-layer representation learning for medical concepts. In: Proceedings of the 22nd ACM SIGKDD, pp. 1495–1504 (2016)
10. Choi, Y., Chiu, C.Y.I., Sontag, D.: Learning low-dimensional representations of medical concepts. AMIA Jt. Summits Transl. Sci. Proc. **2016**, 41 (2016)
11. Glicksberg, B.S., et al.: Automated disease cohort selection using word embeddings from electronic health records. In: PSB, pp. 145–156. World Scientific (2018)
12. Gottlieb, A., Stein, G.Y., Ruppin, E., Altman, R.B., Sharan, R.: A method for inferring medical diagnoses from patient similarities. BMC Med. **11**(1), 194 (2013)
13. Huang, Z., Dong, W., Duan, H., Li, H.: Similarity measure between patient traces for clinical pathway analysis: problem, method, and applications. IEEE J. Biomed. Health Inform. **18**(1), 4–14 (2014)
14. Jensen, P.B., Jensen, L.J., Brunak, S.: Mining electronic health records: towards better research applications and clinical care. Nat. Rev. Genet. **13**(6), 395 (2012)
15. Jia, Z., Zeng, X., Duan, H., Lu, X., Li, H.: A patient-similarity-based model for diagnostic prediction. Int. J. Med. Inform. **135**, 104073 (2020)
16. Johnson, A.E., et al.: MIMIC-III, a freely accessible critical care database. Sci. Data **3**, 160035 (2016)
17. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196 (2014)
18. Li, L., et al.: Identification of type 2 diabetes subgroups through topological analysis of patient similarity. Sci. Transl. Med. **7**(311), 311ra174 (2015)
19. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
20. Miotto, R., Li, L., Kidd, B.A., Dudley, J.T.: Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. Sci. Rep. **6**, 26094 (2016)
21. Miotto, R., Weng, C.: Case-based reasoning using electronic health records efficiently identifies eligible patients for clinical trials. J. Am. Med. Inform. Assoc. **22**(e1), e141–e150 (2015)
22. Musen, M.A., Middleton, B., Greenes, R.A.: Clinical decision-support systems. In: Shortliffe, E.H., Cimino, J.J. (eds.) Biomedical Informatics, pp. 643–674. Springer, London (2014). https://doi.org/10.1007/978-1-4471-4474-8_22
23. Narayanan, A., Chandramohan, M., Chen, L., Liu, Y., Saminathan, S.: subgraph2vec: Learning distributed representations of rooted sub-graphs from large graphs. arXiv preprint arXiv:1606.08928 (2016)
24. Nguyen, D., Luo, W., Nguyen, T.D., Venkatesh, S., Phung, D.: Sqn2Vec: learning sequence representation via sequential patterns with a gap constraint. In: Berlingerio, M., Bonchi, F., Gärtner, T., Hurley, N., Ifrim, G. (eds.) ECML PKDD 2018. LNCS (LNAI), vol. 11052, pp. 569–584. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-10928-8_34
25. Pokharel, S., Li, X., Zhao, X., Adhikari, A., Li, Y.: Similarity computing on electronic health records (2018)

26. Pokharel, S., Zuccon, G., Li, X., Utomo, C.P., Li, Y.: Temporal tree representation for similarity computation between medical patients. Artif. Intell. Med. **108**, 101900 (2020)
27. Rjeily, C.B., Badr, G., Al Hassani, A.H., Andres, E.: Predicting heart failure class using a sequence prediction algorithm. In: 2017 4th International Conference on Advances in Biomedical Engineering (ICABME), pp. 1–4. IEEE (2017)
28. Seymour, C.W., et al.: Assessment of clinical criteria for sepsis: for the third international consensus definitions for sepsis and septic shock (sepsis-3). JAMA **315**(8), 762–774 (2016)
29. Shervashidze, N., Schweitzer, P., van Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-lehman graph kernels. J. Mach. Learn. Res. **12**, 2539–2561 (2011)
30. Sidorov, G., Gelbukh, A., Gómez-Adorno, H., Pinto, D.: Soft similarity and soft cosine measure: similarity of features in vector space model. Computación y Sistemas **18**(3), 491–504 (2014)
31. Sun, J., Wang, F., Hu, J., Edabollahi, S.: Supervised patient similarity measure of heterogeneous patient records. ACM SIGKDD Expl. Newsl. **14**(1), 16–24 (2012)
32. Utomo, C.P., Kurniawati, H., Li, X., Pokharel, S.: Personalised medicine in critical care using Bayesian reinforcement learning. In: Li, J., Wang, S., Qin, S., Li, X., Wang, S. (eds.) ADMA 2019. LNCS (LNAI), vol. 11888, pp. 648–657. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-35231-8_47
33. Wang, Y., Chen, W., Pi, D., Boots, R.: Graph augmented triplet architecture for fine-grained patient similarity. World Wide Web **23**(5), 2739–2752 (2020). https://doi.org/10.1007/s11280-020-00794-y
34. Wright, A.P., Wright, A.T., McCoy, A.B., Sittig, D.F.: The use of sequential pattern mining to predict next prescribed medications. J. Biomed. Inform. **53**, 73–80 (2015)
35. Yanardag, P., Vishwanathan, S.: Deep graph kernels. In: Proceedings of the 21th ACM SIGKDD, pp. 1365–1374. ACM (2015)
36. Zaki, M.J., Meira Jr., W., Meira, W.: Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge University Press, Cambridge (2014)
37. Zhang, J., Kowsari, K., Harrison, J.H., Lobo, J.M., Barnes, L.E.: Patient2Vec: a personalized interpretable deep representation of the longitudinal electronic health record. IEEE Access **6**, 65333–65346 (2018)

Research of Medical Aided Diagnosis System Based on Temporal Knowledge Graph

Fanfei Song[1,2], Bin Wang[1](✉), Yifan Tang[2,3], and Jing Sun[1]

[1] School of Computer Science and Engineering, Northeastern University,
Shenyang 110004, China
fanfeisong@foxmail.com, binwang@mail.neu.edu.cn, sunjing@stumail.neu.edu.cn

[2] Neusoft Corporation, Shenyang 110179, China
tang-yf@neusoft.com

[3] Neusoft Research of Intelligent Healthcare Technology, Co. Ltd.,
Shenyang 110179, China

Abstract. With the advent of medical big data era, medical knowledge graph has received extensive attention. The traditional knowledge graph prediction methods are mostly aimed at static data, which is not suitable for medical diagnostic data with dynamical variation characteristics. Take the example of pulmonary embolism in the clinical medicine domain, it is a typical high-risk lethal disease, and its course of disease has the characteristics of rapid deterioration over time. Therefore, it is necessary to consider the course of disease over time to predict the complications of pulmonary embolism and propose a reasonable diagnosis and treatment recommendation, it brings huge challenge to traditional knowledge graph prediction methods. For this reason, this paper proposes a deep learning method based on embedded representation, by using GRU to introduce temporal information into the knowledge graph and using TransR to ensure the structure property of the knowledge graph, to improve the accuracy and arithmetic performance of intelligent inference of the knowledge graph. The medically aided diagnosis system we have developed has been clinically implemented in several hospitals, and the effectiveness, reliability and stability of the system have been verified through practical application.

Keywords: Temporal knowledge graph · Knowledge representation learning · Translation Model TransR · GRU (Gated Recurrent Unit) · Temporal medical knowledge graph

1 Introduction

Knowledge graph is one of the semantic web technologies, and it is a graph-based data structure, to describe various entities, concepts and the incidence relation between these entities and concepts in the real world [1]. The concept of knowledge graph was formally proposed by Google in May 2012 [2], the

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 236–250, 2020.
https://doi.org/10.1007/978-3-030-65390-3_19

purpose is to improve the capabilities of search engines, improve the quality of search results and enhance the user experience. At present, the well-known general knowledge graphs include Google's Knowledge Graph, IBM's Watson Health, Sogou's Knowledge Cube (www.sogou.com), YAGO DBpedia and so on. They have the characteristics of large scale of data, wide scope of coverage, and containing a great quantity of common sense information. In recent years, Knowledge Graph has made a great progress in vertical areas of medicine, has very good development prospects in the fields of medical quality control, medical auxiliary diagnosis, disease risk assessment, and automated medical knowledge question answering. The typical medical knowledge graph including IBM's Washton Health, Oxford University's LynxKB knowledge graph for pharmacy [3], such as the knowledge map of traditional Chinese medicine at the Institute of Traditional Chinese Medicine Information, Chinese Academy of Medical Sciences [4] etc. However, the main knowledge sources of these medical knowledge graphs are public medical literature, pharmacopoeia, and a small amount of static electronic case data marked by experts but without considering the timing characteristics of clinical medical data.

1.1 Challenges

With the rapid development of medical information technology, major hospitals have accumulated a large amount of medical data for diagnosis and treatment. Taking electronic medical records in real medical scenarios as an example, the medical records contain rich clinical factual knowledge, such as disease knowledge including basic overview and diagnosis of diseases factors, diagnostic factors include symptoms, signs, abnormal examinations, past medical history, personal medical history and other factors; treatment knowledge includes such as drug treatment plan, surgical treatment plan, evaluation and prediction, etc. [5] term knowledge mainly includes symptoms, signs, disease diagnosis and so on, and there are many connections between various medical entities. In this paper, the non-computable knowledge that originally existed in the text data and clinical data is expressed in the form of knowledge graph, and it is transformed into the form of time series data that can be understood and calculated by machine, and using knowledge graph and knowledge inference to reveal the relationship between entities more accurately.

Medical knowledge inference is to further dig hidden information on the basis of the existing medical knowledge base, to enrich, complete and expand the existing knowledge base. In the medical knowledge graph, knowledge inference can help doctors improve disease diagnosis and treatment methods, to avoid medical errors and etc. [6]. The link prediction of knowledge graph [7] is an important application of knowledge graph learning and inference. Its main task is to predict the possible relationships between entities, which can realize the discovery and completion of missing information in the knowledge graph. Since the actual electronic medical record data is generally of low quality, there might be the condition of missing or incorrect relationships between entities. Through the link prediction of knowledge graph in clinical domain, the missing information could be completed and incorrect relationships could be corrected, which has very practical clinical application value [8].

At present, mature relation extraction models utilize attention mechanism, graph convolutional networks (GCNs), adversarial training, reinforcement learning, deep residual learning, and transfer learning [9]. However, these prediction models are good performed basically only at common sense general models, but not changed over time. In the clinical field, there is often a slow or fast evolution process of disease onset, these temporary factual time series knowledge can be converted into a time-series knowledge graph according to timing sequence. However, most of the existing knowledge inference is mainly for static data but without considering the large amount of time series information contained in the medical time series knowledge graph, and it is impossible to make accurate inference for medical knowledge graph with timing sequence.

1.2 Contributions

In order to solve this problem, while constructing this article based on authoritative medical knowledge such as clinical path, we proceeded from the actual diagnosis and treatment data of the hospital electronic medical records, and worked closely with the hospital's expert team to conduct knowledge sorting, data collation, established timing knowledge graph in clinical field, and to propose a learning model base on GRU circulatory neural networks to construct timing knowledge graph in clinical field, to realize the inference prediction for timing knowledge graph in clinical field.

In terms of knowledge representation, we compared the current commonly used translation models.

(1) TransE [10] maps starting entities, relationships, and pointing entities to vectors in the same space. It can handle one-to-one relationships well but cannot satisfy a large number of complex relationships in medical atlases.
(2) TransH [11] can solve the problems of one-to-many and many-to-many, the premise is that the entities and relationships are in the same space, and they cannot meet the data structure characteristics of medical atlas.
(3) An entity in the TransR [12] model is a combination of multiple attributes. Different relationships can focus on different attributes of the entity. TransR maps the entity to the relationship space through the relationship matrix and then minimizes the distance of the triplet.

This feature is more suitable for the characteristics of multiple semantic spaces in medical clinical data relationships. LSTM [13] and GRU are solutions to the short-term memory problem. They have internal mechanisms called "gates" that can regulate the flow of information. Compared with LSTM, using GRU can achieve considerable results, and it is easier to train and improve training efficiency greatly.

Through comparative experiments, this paper finally verifies that the system has a good effect on the inference prediction of timing medical knowledge graph.

Through the comparison experiment of the above models, this system finally confirmed that the TransR+GRU model has a good effect on the inference prediction of temporal medical knowledge graph.

2 Related Work

Knowledge representation learning [14] is a representation learning for entities and relationships in the knowledge base. The function in oriented representation of the semantic information of entities and relationships by projecting entities or relationships into a low-dimensional vector space [15]. Since knowledge representation learning can significantly improve the computational efficiency, effectively alleviate the sparsity of data, and realize the fusion of heterogeneous information, so for the construction, reasoning and application of the knowledge base are of great significance.

At present, the representative models of knowledge representation commonly used in academia are: distance model [16], single-layer neural network model [17], Energy Based Model [18], Bilinear Model [19], Tensor Neural Network Model [20], matrix decomposition model and translation model, etc. Knowledge-aware applications include natural language understanding (NLU), question answering, recommendation systems, and miscellaneous real-world tasks, which inject knowledge to improve representation learning.

In addition, YIDUCLOUD (https://www.yiducloud.com.cn/) etc. are also popular in the clinical domain knowledge graph in the industry. However, these methods are currently only applicable to static knowledge graphs. It is not suitable for the representation learning of medical time series knowledge graph with dynamic changing characteristics.

Regarding the application of time series information in knowledge graph representation learning, Leblay and Chekol [21] investigated temporal scope prediction over time-annotated triple, and simply extended existing embedding methods, Melisachew WC and Giuseppe P [22] studied the predictive model on the deterministic knowledge graph. However, most of the existing research is focused on the research of the general knowledge graph with time sequence, and there is a lack of medical time sequence research that is particularly prominent in the vertical field of clinical medicine.

3 Methodology

In order to facilitate the reader's understanding, this paper uses pulmonary embolism as a case to introduce the overall framework of the clinical domain temporal sequence knowledge graph auxilliary diagnosis and prediction model, and on this basis, gives the specific training process of the model.

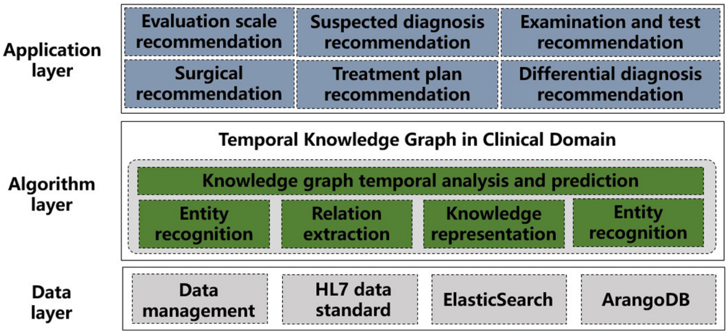

Application layer
Evaluation scale recommendation
Suspected diagnosis recommendation
Examination and test recommendation
Surgical recommendation
Treatment plan recommendation
Differential diagnosis recommendation
Algorithm layer
Temporal Knowledge Graph in Clinical Domain
Knowledge graph temporal analysis and prediction
Entity recognition
Relation extraction
Knowledge representation
Entity recognition
Data layer
Data management
HL7 data standard
ElasticSearch
ArangoDB

Fig. 1. Overall system architecture

3.1 Model Overall Framework

The medical temporal knowledge graph in this paper is based on clinical actual EMR data, and is constructed according to the data structure of the specialty database in a bottom-up way. Among them, the date input by the model is the triple sequence data of the medical temporal domain knowledge graph, and the output date is the prediction result of the relationship between the entities. The overall system architecture is as follows (Fig. 1):

The temporal knowledge graph in the clinical domain is composed of a data layer, an algorithm layer and an application layer, where the data layer contains the patient's personal information, family medical history, time to visit, chief complaint information, test indicators, medication and other information. Each entity has its own attributes and attribute values, for example: the medical card "LN-SY2021-1" belongs to the patient entity information, therefore, "LN-SY2021-1" is instantiated as the patient's physical medical card. The attribute value of the patient name "A" corresponding to "LN-SY2021-1" is taken as

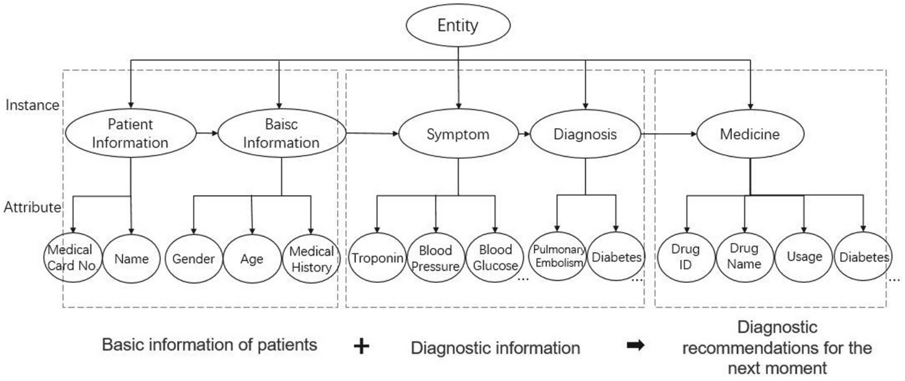


Fig. 2. Medical temporal knowledge graph data structure model

the attribute value of the patient name. The schematic diagram of the formal composition of data is shown in Fig. 2.

In the algorithm layer, the GRU assisted decision-making and reasoning service obtains the electronic medical record chief complaint, current medical history, past history, personal history and other information, and through the electronic medical record structural analysis, enters the knowledge graph for entity matching. After matching, in the application layer, the diagnosis and treatment path developed over time according to the course of disease is analyzed and calculated by the graph sequential inference engine to recommend knowledge such as suspected diagnosis, differential diagnosis, examination items, evaluation scale, treatment plan, etc. for the doctor.

The model training process is:

(1) The triplet x^n is submitted through TransR [12] for dimensionality reduction, and then its vectorized processing result is taken as input.
(2) After entering the GRU layer, we calculated the examination items or disease judgments required at the time of $(t+1)$ by using the GRU sequence characteristics according to the evolutionary order of disease changes over time $t-1$ to t, that is, auxiliary diagnostic inference.

As a result of pulmonary embolism in disease diagnosis process has a number of numerical test indicators, patient also need daily monitoring to prevent them in hospital, so we use temporal knowledge graph of pulmonary embolism triples sequence $X^{(i)}$ as input, after the auxiliary decision-making reasoning can predict the patients' physical process may need to have the next time the entity detection project Y, such as ultrasonic cardiogram, Y is the output. Please refer to Table 1 for the specific process.

Table 1. Prediction model of time-series knowledge graph in medical domain

Variables	Description	Values
X$^{(1)}$	Sequence of Troponin	(patient, troponin_test|1, value) (patient, troponin_test|2, Value)
X$^{(2)}$	Sequence of Blood Pressure	(patient, b_pressuren_test|1, value) (patient, b_pressuren_test|2, value)
...	...	...
X$^{(n)}$	Sequence of Blood Glucose	(patient, b_glucose_test|1, value) (patient, b_glucose_test|2, value)
Y	Complication with Pulmonary Embolism	(patient, yes/no complication diagnosis, Pulmonary Embolism)

$X^{(i)}$ in Table 1 represents the attributes corresponding to various entity relationships of the patient entity, the calculation formula is as follows:

$$X^{(i)} = \{X_1^i, X_2^i, ..., X_n^i\} \tag{1}$$

The whole reasoning process can be expressed as formula:

$$P\{X, Y\} = \{X^1, X^2, ..., X^n, Y\} \tag{2}$$

3.2 Model Vectorization Representation Based on TransR

This paper uses the TransR model to embed the triplets (Ei, R, Ej) in the medical clinical time-series knowledge graph G into the low-dimensional space. Due to the large number of semantic many-to-many relationships in the clinical field, such as multiple patient entities and examination relationships exist between many different detection index entities. Some of these patients belong to cardiology entities, some belong to respiratory entities, and some of the examination relationships are electrocardiogram examination relationships, and some are ultrasound examination relationships..., a patient entity defined as a complex of multiple attributes, different relationships focus on different attributes of entities, different relationships have different semantic spaces, entities and relationships are represented as vectors in the semantic space Ri, and each relationship corresponds to a specific Relation space Rj.

Figure 3 is an example, Patient A has a test relationship with troponin, a numerical test indicator. After the test relationship is projected, it is embedded in vector coordinates. Therefore, patient A and troponin establish a test vector conversion relationship.

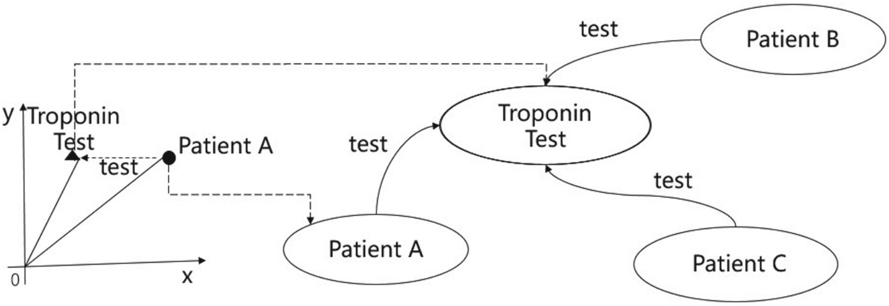


Fig. 3. TransR model vector conversion example

When TransR converts each tuple(h, r, t), the entity in the entity relationship is first projected by r into a projection matrix Mr, and then the entity vector is expressed as a subspace of the entity projection relationship r, and hr and tr are obtained.

$$\begin{aligned} h_r &= hM_r \\ h_t &= tM_r \end{aligned} \tag{3}$$

$$f_r(h, r) = ||h_r + r - t_r||_2^2 \tag{4}$$

TransR model supports the processing of different entities embedded in different semantic Spaces, which is consistent with the characteristics of multiple semantic Spaces in medical clinical data relationships.In the TransR model, the differentiation of similar entities in solid space was effectively solved by projecting onto each entity vector space, and then the transformation from head to tail entities was established, and the loss function was defined, so as to further solve the differentiation of many-to-many relationships in the medical clinical knowledge graph.

3.3 Reasoning Method of Medical Assistant Diagnosis Based on GRU

In this paper, we feed TransR processed triples into GRU, which not only maintains the original semantics, but also maintains the timing characteristics of the input data. Therefore, the combination of triplet and GRU after TransR can fully and accurately infer the dependence relationship between sequences by utilizing the superposition and enhancement of historical information in the medical clinical sequence diagram sequence.

GRU is a good variant of LSTM network. Compared with LSTM network, GRU has simpler structure and better effect. Therefore, GRU is also a kind of current very manifold network, which can solve the long dependence problem in RNN network.In order to make the demonstration more complete, in this section, we show the reasoning process of GRU through Fig. 4.

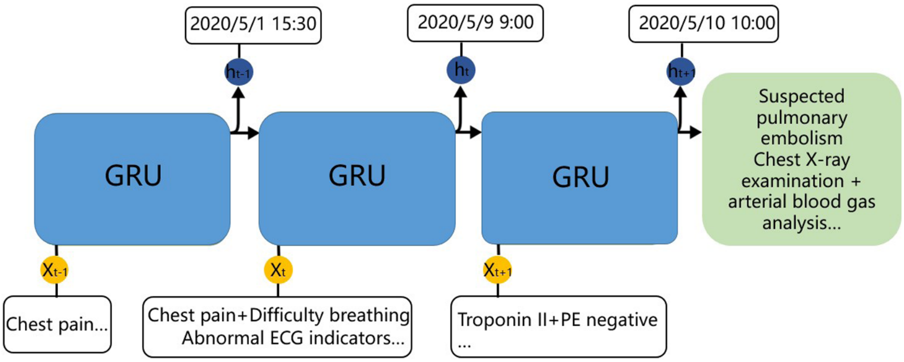

2020/5/1 15:30
2020/5/9 9:00
2020/5/10 10:00
ht-1
ht
ht+1
GRU
GRU
GRU
Suspected pulmonary embolism Chest X-ray examination + arterial blood gas analysis...
Xt-1
Xt
Xt+1
Chest pain...
Chest pain+Difficulty breathing Abnormal ECG indicators...
Troponin II+PE negative ...

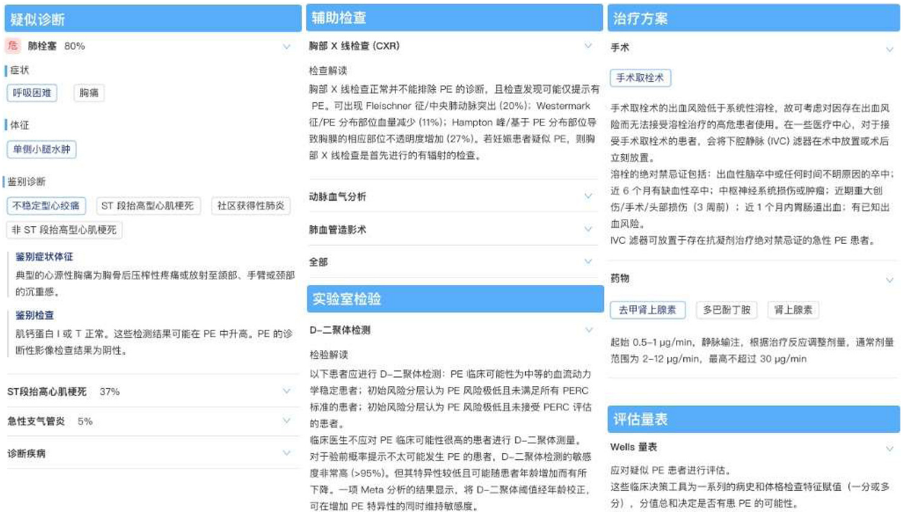
Fig. 4. The reasoning process of the GRU method

Aiming at the sequence of triples existing in the time period $(t-1,t)$in the clinical domain time series knowledge graph as input, through the operation of the GRU [23] method, the auxiliary diagnosis result at time $t+1$ can be inferred.

In summary, GRU preserves important features through the gate function, thus ensuring that they are not lost during long-term propagation. With the help of the GRU method, we ensured the integrity of the disease course information with temporal series data characteristics, and finally got the conclusion of auxiliary diagnosis.

3.4 System Display

Figure 5 is the system's clinical process auxiliary reasoning effect diagram. The system can support suspected diagnosis recommendation, symptom recommendation, examination recommendation, differential diagnosis recommendation, treatment plan recommendation (drug recommendation and surgical procedure recommendation) and evaluation scale recommendation function.



疑似诊断

危 肺栓塞 80%

症状

呼吸困难 胸痛

体征

单侧小腿水肿

鉴别诊断

不稳定型心绞痛 ST 段抬高型心肌梗死 社区获得性肺炎 非 ST 段抬高型心肌梗死

鉴别症状体征

典型的心源性胸痛为胸骨后压榨性疼痛或放射至颌部、手臂或颈部的沉重感。

鉴别检查

肌钙蛋白 I 或 T 正常。这些检测结果可能在 PE 中升高。PE 的诊断性影像检查结果为阴性。

ST段抬高心肌梗死 37%

急性支气管炎 5%

诊断疾病

辅助检查

胸部 X 线检查 (CXR)

检查解读

胸部 X 线检查正常并不能排除 PE 的诊断，且检查发现可能仅提示有 PE。可出现 Fleischner 征/中央肺动脉突出 (20%)；Westermark 征/PE 分布部位血量减少 (11%)；Hampton 峰/基于 PE 分布部位导致胸膜的相应部位不透明度增加 (27%)。若妊娠患者疑似 PE，则胸部 X 线检查是首先进行的有辐射的检查。

动脉血气分析

肺血管造影术

全部

实验室检验

D–二聚体检测

检验解读

以下患者应进行 D–二聚体检测：PE 临床可能性为中等的血流动力学稳定患者；初始风险分层认为 PE 风险极低且未满足所有 PERC 标准的患者；初始风险分层认为 PE 风险极低且未接受 PERC 评估的患者。

临床医生不应对 PE 临床可能性很高的患者进行 D–二聚体测量。

对于验前概率提示不太可能发生 PE 的患者，D–二聚体检测的敏感度非常高 (>95%)。但其特异性较低且可能随患者年龄增加而有所下降。一项 Meta 分析的结果显示，将 D–二聚体阈值经年龄校正，可在增加 PE 特异性的同时维持敏感度。

治疗方案

手术

手术取栓术

手术取栓术的出血风险低于系统性溶栓，故可考虑对因存在出血风险而无法接受溶栓治疗的高危患者使用。在一些医疗中心，对于接受手术取栓术的患者，会将下腔静脉 (IVC) 滤器在术中放置或术后立刻放置。

溶栓的绝对禁忌证包括：出血性脑卒中或任何时间不明原因的卒中；近 6 个月有缺血性卒中；中枢神经系统损伤或肿瘤；近期重大创伤/手术/头部损伤（3 周前）；近 1 个月内胃肠道出血；有已知出血风险。

IVC 滤器可放置于存在抗凝剂治疗绝对禁忌证的急性 PE 患者。

药物

去甲肾上腺素 多巴酚丁胺 肾上腺素

起始 0.5–1 μg/min，静脉输注，根据治疗反应调整剂量，通常剂量范围为 2–12 μg/min，最高不超过 30 μg/min

评估量表

Wells 量表

应对疑似 PE 患者进行评估。

这些临床决策工具为一系列的病史和体格检查特征赋值（一分或多分），分值总和决定是否有患 PE 的可能性。

Fig. 5. Medical temporal sequence knowledge graph system renderings

Figure 6 is a collection of samples of aortic embolism disease. Through the graphic display of specific examination items, the doctor can clearly inquire about the patient's disease and the recommended examination items for the disease.

The suspected diagnosis recommendation function is calculated by the graph reasoning engine, which can predict and calculate multiple suspected diagnosis results based on the current patient's medical record information and guide the doctor to confirm the diagnosis. The diagnosis results include disease name, disease code, disease pathology and epidemiological details, diagnosis recommendation basis and supporting literature.

The inspection and test items are recommended for doctors to indicate the current inspection and test items required to support the diagnosis of the patient, including the project name, project result interpretation, and important level for the doctor's clinical work reference.

The treatment plan is recommended to give multiple groups of treatment plans for diagnosis. Each treatment plan includes the type of plan, the method of the plan such as chemotherapy, surgery or drug combination and the description of the plan.

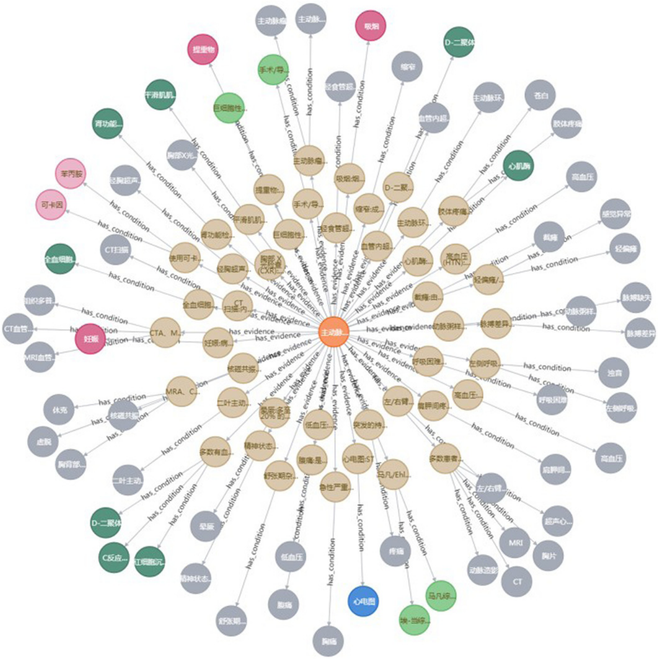


Fig. 6. Sample Atlas of Aortic Embolism Disease

The evaluation scale recommends the functional recommendation, the name and content of the evaluation scale and the scoring items involved in the relevant diagnosis.

The surgical operation recommendation includes the surgical name, surgical code, surgical indications and contraindications.

4 Experiments

4.1 Dataset

We conducted a comprehensive study and obtained 5,627 EHRs focusing on patient diagnosed with pulmonary embolism and other high risk diseases (acute aortic dissection, cardiac tamponade, tension pneumothorax, acute coronary syndrome) from emergency department of an real hospital (due to data security, the name of the hospital cannot be disclosed). These records encompassed emergency-patient encounters presenting from January 2018 to December 2019. It was manually annotated using the schema to train the NLP information extraction model for building the clinical knowledge graph. The NLP model uses deep learning technology to automate the annotations of free text EHR annotations into standardized dictionaries and clinical features, so that clinical information can be further processed for diagnostic classification.

The electronic medical record section includes chief complaint, history of present illness, physical examination, and laboratory tests. The clinical knowledge graph consists of 32,000 entities and 14 relationship types. The knowledge graph includes a total of 188,710 triples. In this paper, several algorithms were implemented for further analysis which includes Rescal tensor decomposition, DNN , Trans-E, Trans-D [24] and Trans-R combined with GRU and LSTM as common inference prediction models. In order to verify the model, A total of 5,627 EHRs was selected as final dataset for validation.

4.2 Evaluation

The performance of each classifier was measured using the 5-fold cross validation. Several measurements are used to evaluate classifiers and they are accuracy, recall, precision and F1-score. We adopt the following well known equations, where TP, FN, TN, FP refer to the numbers of true positives, false negatives, false negatives and false positives respectively.

Accuracy is the proportion of correct predictions among the total number of cases examined and given by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Recall is the number of true positives divided by the number of true positives and the number of false negatives and given by:

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

Precision is the number of true positives divided by the number of true positives and the number of false positives and given by:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

The F_1 score is a balanced measurement used to assess the effectiveness of the performance. Its definition is given by:

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{8}$$

4.3 Results and Discussion

In this chapter, the accuracy of model tests with different algorithms is compared, and the combination mode of TransR with GRU is compared with other combination models.

Algorithm Model Accuracy Comparison. In this chapter, the accuracy of model tests with different algorithms is compared, and the combination mode of

TransR with GRU is compared with other combination models. In this chapter, the accuracy of model tests with different algorithms is compared, and the combination mode of TransR with GRU is compared with other combination models. In this chapter, the accuracy of model tests with different algorithms is compared, and the combination mode of TransR with GRU is compared with other combination models.

First of all, this paper applies this model and other 7 reference model methods in the temporal knowledge graph of the clinical domain. Table 2 shows the accuracy comparison of these methods.

As can be seen from Table 2 the accuracy of the model used in this paper is 13.10% higher than that of TransD with GRU method in the second place, and 13.13% higher than that of TransR with LSTM in the third place. However, the advantages of using only DNN and Rescal models are not particularly obvious. Experiments show that the proposed TransE with GRU method has the highest accuracy. It also scored highest in recall rates and F1-scores.

Thus, it can be concluded that model prediction ability can be improved by using clinical sequence knowledge graph and extracting features such as semantic richness and temporal. The result of embedding and temporal algorithm combination is generally higher than that of traditional DNN and Rescal methods.

Table 2. Comparison of accuracy of various clinical finding.

Algorithm	Accuracy	Recall	Precision	F1-score
DNN	0.400	0.320	0.381	0.348
Rescal	0.409	0.373	0.402	0.387
Trans-E with LSTM	0.506	0.569	0.506	0.535
Trans-E with GRU	0.542	0.604	0.537	0.569
Trans-Dwith LSTM	0.551	0.658	0.542	0.594
Trans-D with GRU	0.624	0.714	0.605	0.655
Trans-R with LSTM	0.622	0.711	0.604	0.653
Trans-R with GRU	0.755	0.800	0.734	0.766

Comparison of Accuracy of Various Clinical Finding. Table 3 shows the predicted results of 2,714 patients diagnosed with pulmonary embolism in 5,627 medical records. According to the main complaint classification of patients, it can be divided into chest pain, palpitations, dyspnea, abdominal pain, dizziness and others.

The overall diagnostic prediction classification accuracy can reach 0.324, of which there are a maximum of 1623 cases with chest pain as the main complaint, with an accuracy of 0.377, which is better than the overall accuracy level, and the accuracy of the main complaint of dizziness can reach 0.474.

The temporal knowledge graph calculation method can help patients make decision to a certain extent by mining the continuous changes in the patient's

Table 3. Accuracy of typing in patients with pulmonary embolism.

Chief complaint	Accuracy
Chest pain	0.377 (613/1623)
Palpitation	0.401 (65/162)
Dyspnea	0.165 (23/139)
Abdominal pain	0.227 (23/101)
Dizziness	0.474 (28/59)
Other	0.203 (128/630)
Total	0.324 (880/2714)

disease period and using the deep learning algorithm to calculate the patient's most likely disease classification.

5 Conclusions and Future Work

This paper proposes an auxiliary diagnostic reasoning system based on a time-series medical knowledge graph in the clinical domain. Through the fusion of medical knowledge and data, we explore and find the best medical plan, aiming to provide clinical decision support for medical staff. Through data, model and other assistance to complete important information prompt, patient status analysis and decision reasoning. In future work, we consider conducting more research in the vertical medical field. Based on the currently constructed medical diagnosis and treatment knowledge base, it integrates deep learning methods based on embedded representation to achieve composite reasoning of local and global reasoning, and achieve multi-mode the fusion reasoning of the state knowledge graph network further improves the accuracy and operation performance of the system.

Acknowledgment. The work is partially supported by the Fundamental Research Funds for the Central Universities (No. N171602003), and Liaoning Distinguished Professor (No. XLYC1902057).

References

1. Zenglin, X., Sheng, Y., He, L., Wang, Y.: Review on knowledge graph techniques. J. Univ. Electron. Sci. Technol. **45**(4), 589–606 (2016)
2. Pujara, J., Miao, H., Getoor, L., Cohen, W.: Knowledge graph identification. In: Alani, H., et al. (eds.) ISWC 2013. LNCS, vol. 8218, pp. 542–557. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41335-3_34
3. Sulakhe, D., et al.: Lynx: a database and knowledge extraction engine for integrative medicine. Nucleic Acids Res. **42**(D1), D1007–D1012 (2014)
4. Jia, L., et al.: Construction of traditional Chinese medicine knowledge graph. J. Med. Inf. (8), 51–53 (2015)

5. Yang, X., Wang, B., Yang, K., Liu, C., Zheng, B.: A novel representation and compression for queries on trajectories in road networks (extended abstract). In 35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, 8–11 April 2019, pp. 2117–2118. IEEE (2019)
6. Li, X., Liu, Y., He, L., Liu, B., Zhang, Y.: Research review of knowledge graph and its application in TCM field. Chin. J. Inf. Tradition. Chin. Med. **24**(7), 129–132 (2017)
7. Baader, F., Sertkaya, B.: Usability issues in description logic knowledge base completion. In: Ferré, S., Rudolph, S. (eds.) ICFCA 2009. LNCS (LNAI), vol. 5548, pp. 1–21. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01815-2_1
8. Yang, X., Li, C.: Secure XML publishing without information leakage in the presence of data inference. In: Nascimento, M.A., Tamer Özsu, M., Kossmann, D., Miller, R.J., Blakeley, J.A., Bernhard Schiefer, K. (eds.) Proceedings of the Thirtieth International Conference on Very Large Data Bases, VLDB 2004, Toronto, Canada, 31 August 31–3 September 2004, pp. 96–107. Morgan Kaufmann (2004)
9. Liu, Z., Sun, M., Lin, Y., Xie, R.: Knowledge representation learning: a review. J. Comput. Res. Dev. **53**(2), 247–261 (2016)
10. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held 5–8 December 2013, Lake Tahoe, Nevada, United States, pp. 2787–2795 (2013)
11. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Brodley, C.E., Stone, P. (eds.) Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, Québec, Canada, 27–31 July 2014, pp. 1112–1119. AAAI Press (2014)
12. Dai, S., Liang, Y., Liu, S., Wang, Y., Shao, W.: Learning entity and relation embeddings with entity description for knowledge graph completion. In: Proceedings of 2018 2nd International Conference on Artificial Intelligence: Technologies and Applications, pp. 202–205 (2018)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
14. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: Burgard, W., Roth, D. (eds.) Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, 7–11 August 2011. AAAI Press (2011)
15. Yang, X., Wang, Y., Wang, B., Wang, W.: Local filtering: improving the performance of approximate queries on string collections. In: Sellis, T.K., Davidson, S.B., Ives, Z.C. (eds.) Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, 31 May–4 June 2015, pp. 377–392. ACM (2015)
16. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. In: Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held 5–8 December 2013, Lake Tahoe, Nevada, United States, pp. 926–934 (2013)

17. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: A semantic matching energy function for learning with multi-relational data - application to word-sense disambiguation. Mach. Learn. **94**(2), 233–259 (2014)
18. Lecun, Y., Chopra, S., Ranzato, MM.A., Huang, F.J.: A tutorial on energy-based learning. Raia Hadsell (2006)
19. Lin, T.-Y., RoyChowdhury, A., Maji, S.: Bilinear CNN models for fine-grained visual recognition. In: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, 7–13 December 2015, pp. 1449–1457. IEEE Computer Society (2015)
20. Scarselli, F., Gori, M., Chung Tsoi, A., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Trans. Neural Netw. **20**(1), 61–80 (2009)
21. Leblay, J., Chekol, .W.: Deriving validity time in knowledge graph. In: Champin, P.-A., Gandon, F.L., Lalmas, M., Ipeirotis, P.G. (eds.) Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon, France, 23–27 April 2018, pp. 1771–1776. ACM (2018)
22. Chekol, M.W., Pirrò, G., Schoenfisch, J., Stuckenschmidt, H.: Marrying uncertainty and time in knowledge graphs. In: Singh, S.P., Markovitch, S., (eds.) Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, California, USA, 4–9 February 2017, pp. 88–94. AAAI Press (2017)
23. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation, pp. 1724–1734. ACL (2014)
24. Ji, G., He, S., Xu, L., Kang, L., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In Meeting of the Association for Computational Linguistics International Joint Conference on Natural Language Processing (2015)

TOP-*R* Keyword-Aware Community Search

Xiaoxu Song[1,2], Bin Wang[1](✉), Jing Sun[1], and Rong Pu[1]

[1] School of Computer Science and Engineering, Northeastern University,
Shenyang 110004, China
songxiaoxu112@163.com, binwang@mail.neu.edu.cn, sunjing@stumail.neu.edu.cn,
purong0519@126.com

[2] College of Software, Shenyang University of Technology, Shenyang 110870, China

Abstract. Community search discovers closely connected subgraphs that involve a set of query vertices and cover all query keywords in a social network. In this paper, we propose a novel query type, namely TOP-*R* keyword-aware community search (TKACS), that retrieves Top-*R* keyword-aware communities that are most familiar with the query keywords. Consider an example, a scholar aims to find a research team that is similar to his research direction. The proposed query takes into account not only sufficient closeness among the members in the keyword-aware community and the textual description of objects (such as AI and Data mining) but also the similarity between the keyword-aware community and the query keywords. Moreover, keyword-aware communities are ranked based on the number of matches between the properties of each member and query keywords. To address this limitation, we propose two efficient algorithms for processing novel queries. Last but not least, we conduct extensive experiments for evaluating the performance of our methods.

Keywords: Community search · Keyword-aware community · TKACS · *K*-core

1 Introduction

The problems of community search naturally exist in many real-life networks such as social network, event collaboration, and communication networks. In these applications, the labels and keywords of the vertices usually describe the attribute information of the vertices. Many existing researches aim to find all the communities from a social network [2,4,7,9–11,15,18,19,23,24,32]. The problems of community search naturally exist in many real-world networks such as social network, event collaboration, and communication networks

The work is partially supported by the National Natural Science Foundation of China (Nos. U1736104, 62072088), and the Fundamental Research Funds for the Central Universities (No. N171602003), Ten Thousand Talent Program (No. ZX20200035), Liaoning Distinguished Professor (No. XLYC1902057).

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 251–265, 2020.
https://doi.org/10.1007/978-3-030-65390-3_20

[5,6,12,14,20,29,30,33]. In recent years, many researches focus on the graph keyword search [16,17,25,27,28,31,34].

However, the criterion for evaluating the optimal community is to have the maximum similarity. [8,13,33] may not be the optimal choice for all users, because each user has different personal preferences. Real-application not only requires finding multiple keyword-aware communities but also every keyword-aware community needs to establish a densely connected structure.

Consider an example, a scholar looks for keyword-aware communities with relation to his/her research. Figure 1 depicts a paper citation network containing authors and their relationship. Each author has textual information in the form of keywords extracted from his/her paper topic such as AI or graph. Given the TKACS query as $Q = (\{$"AI", "Datamining", "Graph"$\}, R = 2, k = 3)$, we aim to find the interdisciplinary papers working on the research of AI and DB. The keyword-aware communities satisfy a 3-core structure and are closer to the query keywords. The query returns two communities with the highest keywords similarity. The two communities are $C_1 = \{u_2, u_3, u_4\}$ and $C_2 = \{u_7, u_8, u_9\}$. Although C_1 is more similar to C_2, users are more interested in "AI".

In this paper, we first formalize the TOP-R keyword-aware community search (TKACS) problem for an attributed graph. TKACS aims to find Top-R keyword-aware communities with a k-core structure while achieving the strongest query keyword closeness. The main challenge compared to traditional Top-R queries [26] is that consider community similarity and k-core structure simultaneously.

To address the TKACS problem, we first develop an efficient and scalable algorithm, namely, the keyword-based algorithm (KA). KA searches for all objects containing the query keywords. Then we find the keyword-aware communities that satisfy the k-core structure through the k-core decomposition method. Finally, we compute the similarity score of keyword-aware communities to the query keywords to find the R best keyword-aware communities. Nevertheless, to search for vertices containing query keywords, it is necessary to traverse all vertices in the dataset. In this way, the time cost of the algorithm is very large. To further improve the performance of our algorithm, we develop another algorithm called inverted index based algorithm (IIA). Through the inverted index, we directly find the vertices containing the query keywords. The major contributions are summarized as follows:

- We formalize the TOP-R keyword-aware community search (TKACS). To the best of our knowledge, this is the first effort to define and address the TKACS problem.
- We propose a novel algorithm, namely, keyword-based algorithm (KA) for efficient query processing of the TKACS problem.
- We further propose an improved algorithm, namely, inverted index based algorithm (IIA). In contrast to the KA, we directly find the vertices containing the query keywords. The time cost has been greatly improved.
- We conduct extensive experiments to demonstrate the effectiveness and efficiency of our proposed algorithms.

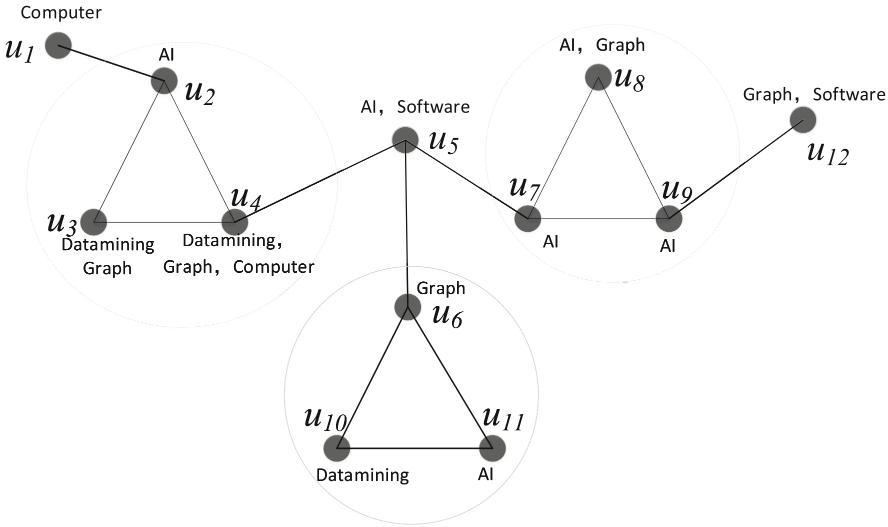


Fig. 1. Motivating example.

2 Related Work

2.1 Community Search

Community search discovers closely connected subgraphs that involve a set of query vertices and cover all query keywords in a social network. In [5], Wanyun Cui proposed a novel model and several algorithms for searching overlapping communities. The algorithm provides a query vertice and searches for overlapping communities useful to that vertice. Since searching for the optimal community in the whole graph is very costly, [6] proposed a local search strategy to add qualified vertices to the community. [12] firstly proposed minimal steiner maximum-connected subgraph which finds a smallest subgraph with the maximum connectivity that contains a set of vertices. In [14], Xin Huang proposed a community search using the k-truss model to find the largest k-truss subgraph that contains a set of query vertices within the minimal radius. [20] devised a search algorithm to solve the problem of the novel p-influential community model with k-core structure in the entire graph.

2.2 Graph Content Search

In recent years, many researches focus on the graph keyword search. In practical applications, people not only consider social relations and geographic location, they also focus on personal interests and attributes. In [27], Jaewon Yang proposed an effective algorithm which detects overlapping communities with vertice attributes by building a community model. However, this study may not be the optimal choice for all users, because each user has different personal preferences. [25] proposed a new generative model that allows one person to participate in multiple communities and a community has multiple topics. In [17], Mehdi Kargar proposed the problem of finding p-cliques which aims to search for a set of attribute vertexes that covers all the giving keywords and the Euclidean distance between all vertexes is no greater than p. [3] proposed a new contextual community model for attributed community search. The community model only needs one query context, and the returned community is both structural and attribute cohesive for the provided query context. [21] proposed a novel attribute community search, called vertex center attribute community search. The community search finds the closer communities with the highest attribute scores.

3 Preliminaries and Problem Formulation

In this section, we first describe some terms and notations of our research. Then we introduce definitions of k-core. Finally, we propose a keyword-aware community model and problem definition.

Table 1. Summary of notation

Notation	Definition
$G(V,E)$	A social network undirected and weight graph
u, v	A user in G
(u, v)	An user edge in G
$\varepsilon(v)$	The property set of v
$N(v)$	The set of neighbors of v in G
$deg(v)$	The degree number of v in G
L	An use gives a set of keyword $(l_1, l_2, \ldots, l_n)$
$G_{kp}(V_{kp}, E_{kp})$	A social network subgraph which is a keyword-aware community with k-core structure
$sim(u)$	The textual similarity between vertice u and keyword set L
$sim(G_{kp})$	The textual similarity between community G_{kp} and keyword set L

In this model, we connect the user relationship with the vertice attribute. Consider an undirected attributed graph $G = (V, E, \Sigma)$, where the set of vertices V denotes vertices and the set of edges E denotes the vertice relations in V, for any two acquaintance users, they $exit(u, v) \in E$. Each vertice is associated with a set of attributes, denoted as $\varepsilon(u) = p_1, p_2, \ldots, p_n$, which describe several properties of the vertice. That is, $\Sigma(u_i) = \{u_1 : (p_1, p_2, \ldots, p_n), u_2 :$

$(p_1, p_2, \ldots, p_m), \ldots\}$, where u_i is a vertice name and p_j is an attribute of u_i. The user gives a keyword l, if any attribute value of vertice u_i is equal to the keyword l, we say vertice u_i contains the keyword l, denoted by $l \in \Sigma(u)$. For example, vertice u represents a person named "Mike" who is interested in "painting", "badminton", "international chess" and "online game". The label of "Mike" is Σ(Mike) = { Mike: ("painting", "badminton", "international chess", "online game") }. For an l, we use $V(u) = \{u \in V : l \in \Sigma(u)\}$ to denote that l belongs to the attribute set of vertice u.

Definition 1 *(K-core) [1]. Given a graph G and an integer k, the maximal connected subgraph $G' = (V', E')$ which is in G is a k-core if $degG'(v) \geq k$ $(\forall v \in V^{'})$.*

Definition 2 *(Coreness) [22]. If a vertice v belongs to a k-core but does not belong to any $(k+1)$-core, we call k the coreness of v.*

Definition 3 *(Keyword-aware community). Given a set of attributes $L = (l_1, l_2, \ldots, l_n)$, a preferred community is a maximal connected subgraph that each vertice u contains a set attributes $\varepsilon(u) = p_1, p_2, \ldots, p_n$ and at least one attribute of every vertice belongs to L.*

Definition 4 *(Textual similarity). Given a set of keywords $L = (l_1, l_2, \ldots, l_n)$, each vertice u contains a set attributes $\varepsilon(u) = p_1, p_2, \ldots, p_n$. The textual similarity of the vertice u is defined as follows:*

$$sim(u) = \frac{L \cap \varepsilon(u)}{L \cup \varepsilon(u)} \tag{1}$$

Problem Statement. (TOP-R keyword-aware community search (TKACS)). The query aims to find Top-R keyword-aware communities which are most familiar with the query keywords. The communities satisfy the following conditions:

- every community G_{kp} is a k-core maximal connected subgraph;
- community similarity is defined as follows:

$$sim(G_{kp}) = \frac{1}{n} \sum_{1 \leq i \leq n} \frac{L \cap \varepsilon(u_i)}{L \cup \varepsilon(u_i)}; \tag{2}$$

- $\exists p_j \in L$, the property $(p_1, p_2, \ldots, p_n)$ of u_i in community.

4 The Proposed Algorithms

In this section, we introduce an algorithm for solving the TKACS problem. The TKACS aims to find Top-R communities with the highest text similarity where every community is a k-core maximal connected subgraph and each vertice properties contain at least one keyword given by the user. However, in practical problems, it is difficult to find a keyword-aware community where each vertice contains at least one keyword given by the user and every community satisfies k-core at the same time. The algorithm is divided into two parts. In the first part, we propose an algorithm, namely Keyword-based Algorithm (KA), which first finds all the vertices containing the query keywords. Then we do k-core decomposition to find all k-core maximum connected subgraphs, that is, k-core based keyword-aware community. Next, we further propose an optimization algorithm, Inverted Index based Algorithm (IIA), which speeds up to find the vertices containing the query keywords, and builds the property of the vertices into an inverted index. Since the improvement of IIA can directly find the vertices containing the query keywords instead of traversing all vertices in the entire graph like KA. The query time of IIA can be greatly reduced. Then we do k-core decomposition to find a k-core based keyword-aware community. The second part of the algorithm is to calculate the similarity of all k-core based keyword-aware communities according to the similarity formula of the keyword-aware community, and select Top-R keyword-aware communities with a k-core structure for the user.

4.1 Keyword-Based Algorithm

The naive approach to find the TOP-R keyword-aware community search requires searching for all objects containing the query keywords and satisfying the k-core structure, then calculating the textual relevance of finding the objects to the query keywords, to compute the similarity score of keyword-aware community. These steps are repeated for every keyword-aware community to find the R best keyword-aware communities.

The pseudo-code of the keyword-based algorithm is presented in Algorithm 1. At first, we initialize the graph G to establish the relationship between the vertices and the label of each vertice. The vertices and the query keywords are stored in V and L respectively (line 1). Then we check whether the property of every vertice in the set V contains the query keywords. If the property of vertice u contains the query keywords, we store u to graph $G^{'}$ and establish a new subgraph $G^{'}$(lines 2–6). We iteratively delete all vertices with degree less than k (line 7). Then we use graph breadth-first search method to divide the unconnected vertices in $G^{''}$ into several connected subgraphs and store the subgraphs in $G_{kf(i)}$. First, we put a vertice v in $V^{''}$ into the queue (line 11). After that, the first vertice in the queue Q is assigned to s every time (line 13) and delete s from $V^{''}$ (line 14). We set s to be visited (line 15) and traverse neighbor vertices of vertice s (line 16). If the neighbor vertice m is not visited (line 17), m enqueues into queue Q (line 18). Until all vertices in the queue Q are visited, we store visited vertices to $G_{kf(i)}$ as a keyword-aware community. We iteratively access all vertices in $V^{''}$ until every vertice is assigned to a keyword-aware community. We calculate the similarity of each keyword-aware community separately (line 21). Then we find Top-R keyword-aware communities with the highest similarity (line 22).

```
Algorithm 1: Keyword based Algorithm
Input: $G$, keyword, $k$, $R$
Output: Top-$R$ keyword-aware communities with $k$-core structure
$V \leftarrow$ all vertices in the graph $G$; $L \leftarrow$ keyword;
for each $u \in V$ do
    for each $p \in u.property$ do
        if $p \in L$ then
            $G^{'} \leftarrow u$;
            break;
iteratively delete all vertices with degree less than $k$ in the $G^{'}$;
$G^{''} \leftarrow G^{'}$; $i$=1; initial an empty queue $Q$;
while each $v \in V^{''}$ do
    $i = i + 1$;
    $Q$.enqueue($v$);
    while $Q$ do
        $s$=$Q$.dequeue();
        $V^{''}$=$V^{''}$-$s$;
        visited.append($s$);
        for each $m \in Neighbors(s)$ do
            if $m$ not in visted then
                $Q$.enqueue($m$);
    $G_{kf(i)} \leftarrow$ visited;
    visited.clear();
    $sim_i$=similiarity($L$, $G_{kf(i)}$);
$\Gamma$ ←find Top-$R$ $sim_i$;
return $\Gamma$;
```

4.2 Inverted Index Based Algorithm

To efficiently accelerate the search, we propose an Inverted Index based Algorithm (IIA). The pseudo-code of the Inverted Index based Algorithm is presented in Algorithm 2. At first, we initialize the graph G to establish the relationship between the vertices and the label of each vertice. The vertices and the query keywords are stored in V and L respectively (line 1). We build up an inverted index and find the vertices containing the query keywords from the inverted index. If the property of vertice u contains the query keywords, we store u to graph $G^{'}$ and establish a new subgraph $G^{'}$ (lines 2–3). We iteratively delete all vertices with degree less than k (line 4). Then we use graph breadth-first search method to divide the unconnected vertices in $G^{''}$ into several connected subgraphs and store the subgraphs in $G_{kf(i)}$. First, we put a vertice v in $V^{''}$ into the queue (line 8). After that, the first vertice in the queue Q is assigned to s every time (line 10) and delete s from $V^{''}$ (line 11). We set s to be visited (line 12) and traverse neighbor vertices of vertice s (line 13). If the neighbor vertice m is not visited

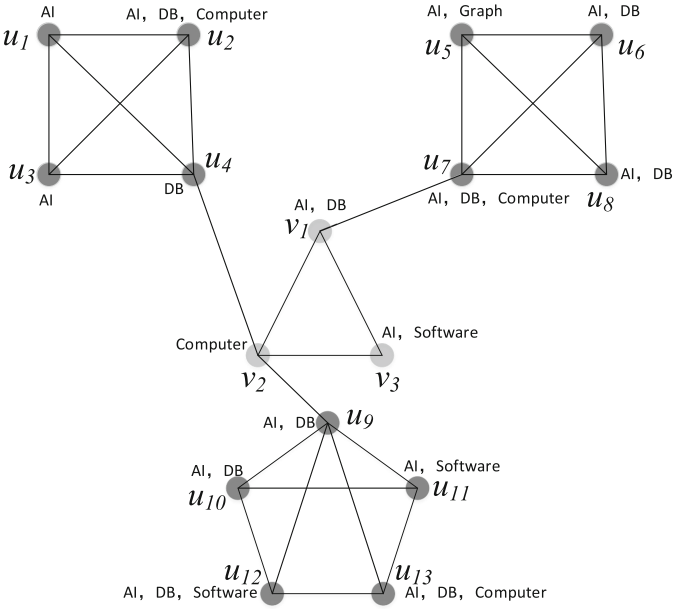


Fig. 2. An example for KA.

(line 14), m enqueues into queue Q (line 15). Until all vertices in the queue Q are visited, we store visited vertices to $G_{kf(i)}$ as a keyword-aware community. We iteratively access all vertices in $V^{''}$ until every vertice is assigned to a keyword-aware community. We calculate the similarity of each keyword-aware community separately (line 18). Then we find Top-R keyword-aware communities with the highest similarity (line 19).

4.3 Complexity Analysis

In this section, we analyze the time complexity of the two algorithms proposed in this paper. Given a TKACS query and the graph G, $G_{kf(i)}$ is the keyword-aware community. The time complexity of KA algorithm is $O(|V| \times |P| \times |K| + |V^{'}| + |E^{'}| + |V^{''}| + |V^{''}| \times |P| \times |K|)$. Let $|V|$ be the number of vertices in G. The number of properties for each vertice is $|P|$. The number of query keywords is $|K|$. $|V^{'}|$ represents the number of vertices in $G^{'}$. $|E^{'}|$ represents the number of edges in $G^{'}$. $|V^{''}|$ represents the number of vertices in $G^{''}$. As shown in Algorithm 1, Keyword based Algorithm is firstly to iteratively compute whether the property of all vertices contains the query keywords. The time complexity of this process is $O(|V| \times |P| \times |K|)$. We find the vertices where the core value is greater than k in $V^{'}$. The time complexity of this process is $O(|V^{'}| + |E^{'}|)$. Then we compute the number of keyword-aware communities in $V^{''}$. The time complexity of this process is $O(|V^{''}|)$. Finally, we calculate the similarity of the keyword-aware community and the query keywords. The time complexity of this process is $O(|V^{''}| \times |P| \times |K|)$.

```
Algorithm 2: Inverted Index based Algorithm
Input: $G$, keyword, $k$, $R$
Output: Top-$R$ keyword-aware communities with $k$-core structure
$V \leftarrow$ all vertices in the graph $G$; $L \leftarrow$ keyword;
Building up inverted index;
$G^{'} \leftarrow$find the vertices containing $L$ from inverted index;
iteratively delete all vertices with degree less than $k$ in the $G^{'}$;
$G^{''} \leftarrow G^{'}$; $i$=1; initial an empty queue $Q$;
while each $v \in V^{''}$ do
    $i = i + 1$;
    $Q$.enqueue($v$);
    while $Q$ do
        $s$=$Q$.dequeue();
        $V^{''} = V^{''}$-$s$;
        visited.append($s$);
        for each $m \in Neighbors(s)$ do
            if $m$ not in visted then
                $Q$.enqueue($m$);
    $G_{kf(i)} \leftarrow$ visited;
    visited.clear();
    $sim_i$=similiarity($L$, $G_{kf(i)}$);
$\Gamma \leftarrow$find Top-$R$ $sim_i$;
return $\Gamma$;
```

Another algorithm is Inverted Index based Algorithm, which uses an inverted index to speed up the query. The complexity of the IIA is $O(|K| \times \overline{V_k} + |V^{'}| + |E^{'}| + |V^{''}| + |V^{''}| \times |P| \times |K|)$, where $\overline{V_k}$ represents every keyword containing the average number of vertices. As shown in Algorithm 2, we first search for the vertices containing L from the inverted index. The time complexity of this process is $O(|K| \times \overline{V_k})$. We find the vertices where the core value is greater than k in $V^{'}$. The time complexity of this process is $O(|V^{'}| + |E^{'}|)$. Then we compute the number of keyword-aware communities in $V^{''}$. The time complexity of this process is $O(|V^{''}|)$. Finally, we calculate the similarity of the keyword-aware community and the query keywords. The time complexity of this process is $O(|V^{''}| \times |P| \times |K|)$. In contrast to KA, the IIA query is much faster than KA due to the acceleration of the inverted index.

5 Analysis of Experimental Results

In the section, we have conducted experimental studies using one real social graph dataset, and we study the performance of our algorithm by comparing it with two algorithms. We first introduce the experimental settings.

5.1 Experimental Settings

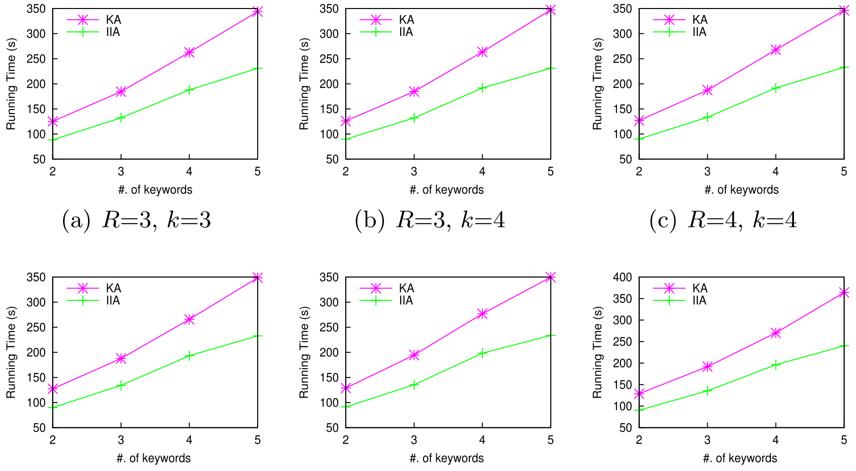
Setup. All the algorithms are implemented in Python. All programs are performed on CPU Intel i7-4790 (3.60 GHz). The operating system is Microsoft Windows 7.

Table 2. Parameters of Dataset

Parameter	Range	Default
k	3, 4, 5, 6	4
The number of keywords	2, 3, 4, 5	3
R	2, 3, 4, 5	3

Datasets. We used one social graph dataset DBLP to evaluate our proposed algorithms. In the DBLP dataset, every vertice in the graph has labels and the relationship between the vertices is represented by edges. The entire graph has 977288 vertices and 3432273 edges.

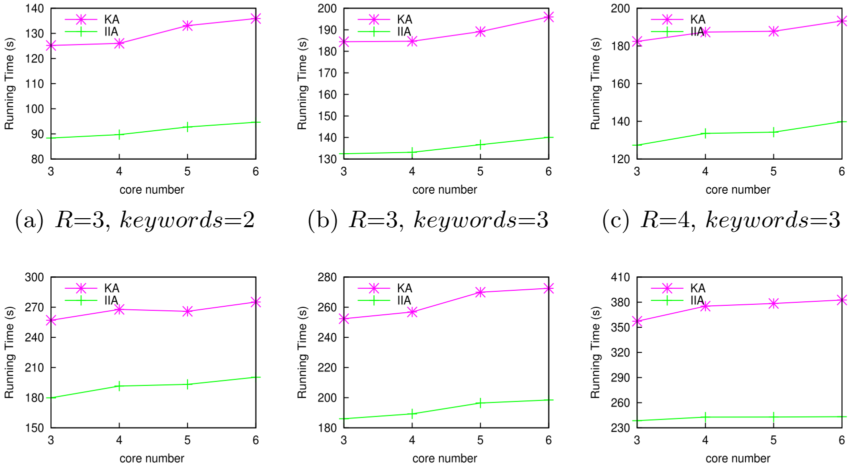
Parameters. The experiments are conducted using different settings on 3 parameters: k (the core value of keyword-aware community), the number of the query keywords, and R (the most similar R keyword-aware communities). The parameters range are shown in Table 2. We set the range of k from 3 to 6. We set the range of *keywords* from 2 to 5. We vary R from 2 to 5. According to different query requests, we choose different parameter values.



(a) R=3, k=3 (b) R=3, k=4 (c) R=4, k=4

(d) R=4, k=5 (e) R=5, k=5 (f) R=5, k=6

Fig. 3. Performance varying the number of keywords



(a) R=3, $keywords$=2 (b) R=3, $keywords$=3 (c) R=4, $keywords$=3

(d) R=4, $keywords$=4 (e) R=5, $keywords$=4 (f) R=5, $keywords$=5

Fig. 4. Performance varying the core number

5.2 Effectiveness

In this section, we verified the effectiveness of the two algorithms KA and IIA for the TKACS problem in dataset DBLP. According to the different needs of users, we find the keyword-aware communities that meet the query conditions. We first fix R and k values and set the number of keywords from 2 to 5 in Fig. 3. Then we fix R and the number of keywords and set core numbers from 3 to 6. We use a real example to illustrate the effectiveness of the query below.

Given the TKACS query as $Q = (\{$“datamining”, “graphtheoretic”, “geosocial”, “privacypreserving”$\}$, $R = 3$, $k = 4)$, we aim to find top-3 keyword-aware communities with the 4-core structure that include papers working on the research of the database. The two algorithms return the top-3 same keyword-aware communities. The similarity of top-3 keyword-aware communities is 0.0732, 0.0675, 0.0666 respectively. The running time of the IIA algorithm is much faster than the KA algorithm. The two algorithms are capable to find the top-3 keyword-aware communities of papers working on the same topic such that they have a densely connected structure and all of them are close to the query keywords.

5.3 Efficiency

In this section, we evaluate the efficiency of the proposed two algorithms. The time cost of the two algorithms is compared by setting different parameters in dataset DBLP. We propose two algorithms to address the TKACS problem. The first algorithm is keyword-based algorithm (KA), which searches the vertices with the query keywords and keyword-aware communities with a k-core structure. The second algorithm is the inverted index based algorithm (IIA), which speeds up to find the vertices containing the query keywords, and builds the property of the vertices into an inverted index.

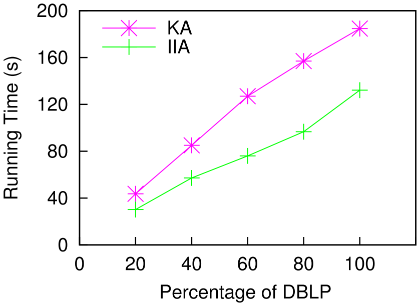


Fig. 5. Performance varying the graph size.

Effect of Varying the Number of Keywords. Figure 3 shows the performance varying the number of keywords by two proposed algorithms in dataset DBLP. We fix the R and core value to evaluate the effect of time cost. We alter the number of keywords from 2 to 5. In the six cases, the time cost of the IIA algorithm always runs faster than the KA algorithm. As the number of keywords increases, the time cost of two algorithms also increases. The reason is that the number of vertices to be processed has increased. In Figs. 3(a) and (b), the core number does not greatly affect the time cost. The same conclusion is also reflected in Figs. 3(c), (d), (e) and (f). When R and core numbers increase simultaneously, the increase in running time is not particularly obvious. The reason is that the execution time of k-core decomposition is very little and keyword-aware community similarity needs to calculate the similarity of all communities.

Effect of Varying the Core Number. We vary the core number from 3 to 6. The time costs of executing the queries in DBLP are shown in Fig. 4. We fix R and the number of keywords, the time cost of two algorithms for the TKACS queries increases with a larger core number. The reason is that the number of iterations of k-core decomposition increases in finding a larger core number. In Figs. 4(a) and (b), the time cost of two algorithms for the TKACS queries increases with a larger number of keywords. This is because the remaining graph for consideration becomes larger. The same conclusion is also reflected in Figs. 4(c), (d), (e) and (f). When R and the number of keywords increase simultaneously, the increase in running time is particularly obvious. The reason is that the execution time of the remaining graph becomes larger, and the number of iterations increases.

Varying the Graph Size. Figure 5 shows the scalability of our algorithms. The parameters in the experiment are default values. We choose the size of the graph, ranging from 20% to 100% in DBLP. With a larger graph size, the cost time of all the algorithms increases. In all graph size settings, the time cost of the IIA algorithm always runs faster than the KA algorithm.

6 Conclusion

In this paper, we define a user query, namely, TOP-R keyword-aware community search (TKACS), to obtain Top-R keyword-aware communities with a densely connected structure that achieving the strongest keyword closeness. To solve this problem, we propose efficient algorithms and indexes. We conduct extensive experiments over a real dataset, and the results demonstrate the efficiency of the proposed algorithms.

References

1. Batagelj, V., Zaversnik, M.: An O(m) algorithm for cores decomposition of networks. arXiv preprint arXiv:cs/0310049 (2003)
2. Chen, H., Jin, H.: Finding and evaluating the community structure in semantic peer-to-peer overlay networks. Sci. China Inf. Sci. **54**(7), 1340–1351 (2011). https://doi.org/10.1007/s11432-011-4296-6
3. Chen, L., Liu, C., Liao, K., Li, J., Zhou, R.: Contextual community search over large social networks. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE), pp. 88–99. IEEE (2019)
4. Chen, Y., Xu, J., Xu, M.: Finding community structure in spatially constrained complex networks. Int. J. Geogr. Inf. Sci. **29**(6), 889–911 (2015)
5. Cui, W., Xiao, Y., Wang, H., Lu, Y., Wang, W.: Online search of overlapping communities. In: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, pp. 277–288 (2013)
6. Cui, W., Xiao, Y., Wang, H., Wang, W.: Local search of communities in large graphs. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, pp. 991–1002 (2014)
7. Expert, P., Evans, T.S., Blondel, V.D., Lambiotte, R.: Uncovering space-independent communities in spatial networks. Proc. Natl. Acad. Sci. **108**(19), 7663–7668 (2011)
8. Fang, Y., Cheng, R., Luo, S., Hu, J.: Effective community search for large attributed graphs. Proc. VLDB Endow. **9**(12), 1233–1244 (2016). https://doi.org/10.14778/2994509.2994538. http://www.vldb.org/pvldb/vol9/p1233-fang.pdf
9. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3–5), 75–174 (2010)
10. Girvan, M., Newman, M.E.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. **99**(12), 7821–7826 (2002)
11. Guo, D.: Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP). Int. J. Geogr. Inf. Sci. **22**(7), 801–823 (2008). https://doi.org/10.1080/13658810701674970
12. Hu, J., Wu, X., Cheng, R., Luo, S., Fang, Y.: Querying minimal Steiner maximum-connected subgraphs in large graphs. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 1241–1250 (2016)
13. Huang, X., Lakshmanan, L.V.S.: Attribute-driven community search. Proc. VLDB Endow. **10**(9), 949–960 (2017). https://doi.org/10.14778/3099622.3099626. http://www.vldb.org/pvldb/vol10/p949-huang.pdf
14. Huang, X., Lakshmanan, L.V., Yu, J.X., Cheng, H.: Approximate closest community search in networks. arXiv preprint arXiv:1505.05956 (2015)

15. Islam, M.R., Kabir, M.A., Ahmed, A., Kamal, A.R.M., Wang, H., Ulhaq, A.: Depression detection from social network data using machine learning techniques. Health Inf. Sci. Syst. **6**(1), 8 (2018). https://doi.org/10.1007/s13755-018-0046-0
16. Kacholia, V., Pandit, S., Chakrabarti, S., Sudarshan, S., Desai, R., Karambelkar, H.: Bidirectional expansion for keyword search on graph databases. In: Böhm, K., Jensen, C.S., Haas, L.M., Kersten, M.L., Larson, P., Ooi, B.C. (eds.) Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, 30 August–2 September 2005, pp. 505–516. ACM (2005). http://www.vldb.org/archives/website/2005/program/paper/wed/p505-kacholia.pdf
17. Kargar, M., An, A.: Keyword search in graphs: finding r-cliques. Proc. VLDB Endow. **4**(10), 681–692 (2011). https://doi.org/10.14778/2021017.2021025. http://www.vldb.org/pvldb/vol4/p681-kargar.pdf
18. Khan, B.S., Niazi, M.A.: Network community detection: A review and visual survey. CoRR abs/1708.00977 (2017). http://arxiv.org/abs/1708.00977
19. Lancichinetti, A., Fortunato, S.: Limits of modularity maximization in community detection. CoRR abs/1107.1155 (2011). http://arxiv.org/abs/1107.1155
20. Li, R.H., Qin, L., Yu, J.X., Mao, R.: Influential community search in large networks. Proc. VLDB Endow. **8**(5), 509–520 (2015)
21. Liu, Q., Zhu, Y., Zhao, M., Huang, X., Xu, J., Gao, Y.: VAC: vertex-centric attributed community search. In: 2020 IEEE 36th International Conference on Data Engineering (ICDE), pp. 937–948. IEEE (2020)
22. Ma, Y.L., Yuan, Y., Zhu, F.D., Wang, G.R., Xiao, J., Wang, J.Z.: Who should be invited to my party: a size-constrained k-core problem in social networks. J. Comput. Sci. Technol. **34**(1), 170–184 (2019)
23. Plantié, M., Crampes, M.: Survey on Social Community Detection. In: Ramzan, N., van Zwol, R., Lee, J.S., Clüver, K., Hua, X.S. (eds.) Social Media Retrieval. Computer Communications and Networks, pp. 65–85. Springer, London (2013). https://doi.org/10.1007/978-1-4471-4555-4_4
24. Ruan, Y., Fuhry, D., Parthasarathy, S.: Efficient community detection in large networks using content and links. In: Schwabe, D., Almeida, V.A.F., Glaser, H., Baeza-Yates, R., Moon, S.B. (eds.) 22nd International World Wide Web Conference, WWW 2013, Rio de Janeiro, Brazil, 13–17 May 2013, pp. 1089–1098. International World Wide Web Conferences Steering Committee/ACM (2013). https://doi.org/10.1145/2488388.2488483
25. Sachan, M., Contractor, D., Faruquie, T.A., Subramaniam, L.V.: Using content and interactions for discovering communities in social networks. In: Mille, A., Gandon, F.L., Misselis, J., Rabinovich, M., Staab, S. (eds.) Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, 16–20 April 2012, pp. 331–340. ACM (2012). https://doi.org/10.1145/2187836.2187882
26. Tsatsanifos, G., Vlachou, A.: On processing top-k spatio-textual preference queries. In: Alonso, G., e al. (eds.) Proceedings of the 18th International Conference on Extending Database Technology, EDBT 2015, Brussels, Belgium, 23–27 March 2015, pp. 433–444. OpenProceedings.org (2015). https://doi.org/10.5441/002/edbt.2015.38
27. Yang, J., McAuley, J.J., Leskovec, J.: Community detection in networks with node attributes. In: Xiong, H., Karypis, G., Thuraisingham, B.M., Cook, D.J., Wu, X. (eds.) 2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, 7–10 December 2013, pp. 1151–1156. IEEE Computer Society (2013). https://doi.org/10.1109/ICDM.2013.167

28. Yang, T., Jin, R., Chi, Y., Zhu, S.: Combining link and content for community detection: a discriminative approach. In: IV, J.F.E., Fogelman-Soulié, F., Flach, P.A., Zaki, M.J. (eds.) Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July 2009, pp. 927–936. ACM (2009). https://doi.org/10.1145/1557019.1557120
29. Yang, X., Li, C.: Secure XML publishing without information leakage in the presence of data inference. In: (e)Proceedings of the 30th International Conference on Very Large Data Bases, VLDB 2004, Toronto, Canada, 31 August–3 September 2004, pp. 96–107 (2004). https://doi.org/10.1016/B978-012088469-8.50012-7. http://www.vldb.org/conf/2004/RS3P2.PDF
30. Yang, X., Wang, B., Yang, K., Liu, C., Zheng, B.: A novel representation and compression for queries on trajectories in road networks. IEEE Trans. Knowl. Data Eng. **30**(4), 613–629 (2018). https://doi.org/10.1109/TKDE.2017.2776927
31. Yang, X., Wang, Y., Wang, B., Wang, W.: Local filtering: improving the performance of approximate queries on string collections. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, 31 May–4 June 2015, pp. 377–392 (2015). https://doi.org/10.1145/2723372.2749445
32. Zhang, Z., et al.: Inductive structure consistent hashing via flexible semantic calibration. IEEE Trans. Neural Netw. Learn. Syst. (2020)
33. Zhang, Z., Huang, X., Xu, J., Choi, B., Shang, Z.: Keyword-centric community search. In: 35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, 8–11 April 2019, pp. 422–433. IEEE (2019). https://doi.org/10.1109/ICDE.2019.00045
34. Zhou, Y., Cheng, H., Yu, J.X.: Graph clustering based on structural/attribute similarities. Proc. VLDB Endow. **2**(1), 718–729 (2009). https://doi.org/10.14778/1687627.1687709. http://www.vldb.org/pvldb/vol2/vldb09-175.pdf

Online Community Identification over Heterogeneous Attributed Directed Graphs

Zezhong Wang[1,2,4](✉), Xiangmin Zhou[3], Yuliang Ma[2], and Xun Yi[3]

[1] State Key Laboratory of Robotics, Shenyang Institute of Automation,
Chinese Academy of Sciences, Shenyang 110016, China
zezhong_wang@sina.cn

[2] School of Computer Science and Engineering, Northeastern University,
Shenyang, China

[3] School of Science, RMIT University, Melbourne, VIC 3000, Australia

[4] Institutes for Robotics and Intelligent Manufacturing,
Chinese Academy of Sciences, Shenyang 110169, China

Abstract. The creating of communities has resulted in the astonishing increase in many areas. Especially in the area of social networks, it has wide applications in the domains such as product recommendation, setting up social events, online games etc. The applications are relied on effective solutions for retrieving communities online. In this way, a great deal of research has been conducted on yielding communities. Unfortunately, the state-of-the-art community identity methods which aim to find out communities containing the query nodes, only consider topological structure, but ignore the effect of nodes' attribute, direction between nodes, and nodes' information across heterogeneous graphs, lead to communities with poor cohesion. Thus, we address the problem of discovering communities online, across heterogeneous directed attributed graphs. We first propose an online method to match pairs of users in heterogeneous graphs and combine them into a new one. Then we propose IC-ADH, a novel framework of retrieving communities in the new directed attributed graph. Extensive experiments demonstrate the effectiveness of our proposed solution across heterogeneous directed attributed graphs.

Keywords: Online community identification · User identity linkage · Attributed directed graphs

1 Introduction

With the rising popularity of social media, identifying users communities online is of great use for personal tasks online. A large amount of research effort have been undertaken to get communities online. However, existing work simply assumes that searching communities is finding compact subgraphs in one undirected graph without attributes, which may not be accurate in many social application scenarios. For identifying communities online over heterogeneous directed

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 266–280, 2020.
https://doi.org/10.1007/978-3-030-65390-3_21

attributed graphs, several key issues need to be addressed. The first issue is how to identify communities customized for a query request. This is vital, as getting all the communities in the large graph is a waste of time and doesn't make sense for the user. We need to propose a method of identifying communities based on query request, which is suitable for quick or online retrieval of communities. The second issue is how to take full advantage of users' information in all the social networks. It is well known that an individual always simultaneously register in several social networks. A real natural person can choose different attributes in different online social network sites such as age, gender, location, education background, contact information, etc. Thus, we are supposed to propose a method to filter information and integrate information in heterogeneous graphs. The third issue is how to get compact subgraphs based on multiple kinds of user's information, such as topological structure and attributes of nodes in the graph. Direction of users' relationship is meaningful and unsymmetrical. For example, a celebrity could have a huge number of fans, but follow none. Thus, direction of relationship could reveal characteristics of users. If we simply replace the directed relationship with undirected edges, we could get communities with low cohesiveness and even wrong ones [1]. Meanwhile, ignoring the attributes of users could cause the communities with low effectiveness. Under this circumstance, we need to construct a model, which captures direction of users' relationship and users' attributes. To overcome these problems, we design a novel approach that exploits the users' relationship and attributes over multiple heterogeneous social networks to identify communities online. We first match users accounts in different social networks and combine the social networks into a new one. Then, we identify users' communities in the graph. The contributions of the article are highlighted as follows:

- We propose a solution to link users' accounts of different social network in a short time, and get more accurate attributes based on the matched users. In this way, we take full use of users' relationship and attributes and enrich the users' information greatly.
- We design a novel k-ADcore model based on topological structure and nodes' attributes over heterogeneous graphs, which measures the "goodness" of communities online.

The remainder of the paper is organized as follows. We briefly review the related work in Sect. 2. We introduce problem definition in Sect. 3. In Sect. 4, we describe how to processing users' information over heterogeneous graphs. In Sect. 5, we describe the method for retrieval of communities. In Sects. 6 and 7, we show the approach's cost analysis and report extensive experimental results. Finally, Sect. 8 concludes the paper.

2 Related Work

In this section, we review the existing research on two problems closely related to our work, including the community search and user identity linkage across heterogeneous social networks.

2.1 Community Search

Community search methods aim to identify communities online. Global method [2], which yields a subgraph containing query vertex, is the first algorithm in this area. Local method [3] enhance efficiency of communities' retrieval, by using local expansion techniques. We will compare the two kind of solutions in our experiments. Most work focus on topological structure and construct multiple models to retrieve communities, including k-clique [4], k-truss [5] and spectral cluster [6]. However, these algorithms overlook the meaning of relationships' direction, and work on undirected graphs, which could lead to low effectiveness of communities. Recently, some methods are proposed to focus on directed graphs [7]. Yixiang [7] propose D-core and a new minimum degree measure to get communities in directed graph. Nevertheless, it ignores the attributes of users. Some others [8,9] work on attributed graphs. However, the methods overlook the information across heterogeneous social networks, which is of great significance to learn users. As we will see, performing identifying communities online method on attributed directed heterogeneous graphs is better than methods, which only consider one or two characters of the three.

2.2 User Identity Linkage

To link users over heterogeneous social networks, research should extract users' features. User identity feature could be classified into three kinds, content [10–12], network components [13–16] and profile [17–20]. Oana [12] proposes a method to extract features based on content, which applies a logistic regression classifier match pairs of users. However, the methods based on content can not perform well, when the users are not active. In addition, a great deal of research works on network components. The paper [15] use a graph theoretic model based on neighborhood-based network structure to match users in heterogeneous social networks. Nevertheless, the features extracted based on networks, could be very noisy. Thus, this kind of algorithm is not suitable for our method. The paper [20] is based on profile to link users in different social networks, which uses distance-based profile feature. It is easy for researchers to get these features, but users could selectively fill up with the information, which leads to wrong matches. We search communities online, so we should find some way to get matched users in a short time and avoid these problems above.

3 Problem Definition

This section provides a formal problem definition and describes our proposed approach briefly. Our solution works on directed attributed heterogeneous graphs, which is very different from others. Thus, it is of great importance to define $heterogeneous social network$.

Definition 1 ***(Heterogeneous social network).*** *A heterogeneous social network can be modeled as a attributed directed graph G = (V, E, A, N), where V represents vertices in the graph, E is the set of all the edges in the graph, which represents the relationships between the vertices, A is the set of vertices' attributes and N is the set of all vertices' name strings.*

Most community search methods adopt minimum degree measurement. We also choose minimum degree to measure cohesiveness of communities, with no exception. However, our method works on directed graphs. Thus, we should define $min - degree$, which is suitable for our method.

Definition 2 ***(min-degree).*** *In directed graphs, in-degree of a vertex, is the number of edges getting into the vertex, and out-degree of a vertex, is the number of edges getting out of the vertex. Min-degree is the minimum of in-degree and out-degree of the vertex.*

To retrieve communities over directed attributed graphs, we should construct a new model, $k - ADcore$.

Definition 3 ***(k-ADcore).*** *Given a directed attributes graph G and an integer k (k≥1), a* ***k-ADcore*** *is compact subgraphs, which should hold the following properties:*

1. ***Strong Connectivity.*** *G_q is strongly connected and contains q;*
2. ***Structure cohesiveness.*** *∀v∈ G_q, min-degree of $(v) \geq$ the threshold k;*
3. ***Structure cohesiveness.*** *∀v∈ G_q share common attributes.*

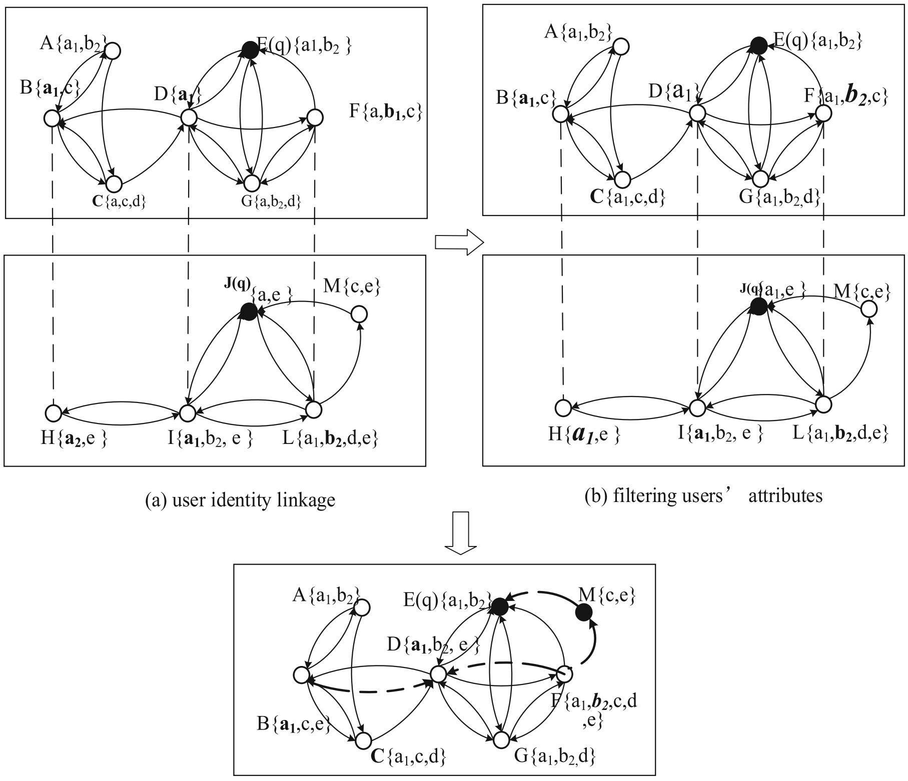
Proposition 1 **(IC-ADH).** Given two heterogeneous social networks $G_1(V, E, AN)$, $G_2(V, E, A, N)$ and the vertex query q, **IC-ADH** (Identifying communities online over attributed directed heterogeneous graphs) is the framework to get a set of communities G_q.

4 Processing Information over Heterogeneous Graphs

In this section, we demonstrate how we integrate vertices' attributes and edges between vertices over heterogeneous graphs, and how to filter matched users' wrong attributes based on information over heterogeneous graphs. Finally, we build a new social network, which contains all the information in the two heterogeneous social networks.

4.1 Overview of Our Solution

In order to utilize users' relationships and attributes in other social networks, we should first link users' accounts over heterogeneous social networks. Secondly, as mentioned above, users could selectively fill up the attribute with different keywords, which could lead to confusion of the attribute. Thus, we find a method to filter matched users' attributes, which is suitable for the next step. Finally,



(a) user identity linkage

(b) filtering users' attributes

(c) a new social network

Fig. 1. Overview of processing information over heterogeneous graphs

for convenience of taking full advantage of users' information over heterogeneous social networks, we combine the two heterogeneous social networks into one which contain all information of users, and identify communities from the new social network.

As shown in Fig. 1, the vertices $A, B, C, E, F, G, H, I, J, K, L$ are users in the social networks, and a, a, b, b, c, d, e are the attributes of users. a_1, a_2 are the different keywords of attributes a, b_1, b_2 are the different keywords of attributes b. For an example, a_1, b_2 is the set of A's attributes. The edges represent the relationships between users. The directed edge from D to E means that user D follows user E.

Firstly, we propose a method to compare users online in the two heterogeneous social networks, and finally match the users' account as the same natural person, by taking advantage of user identity linkage. In Fig. 1(a), we match users B, D and F in the first social network with users I, F and L, respectively. Then, we filter users' attributes based on the result of matched users. As shown in Fig. 1(b), user account B, in the first social network owns the keyword a_1 of the attributes a, which is different from the matched user account in the second social network H's keyword a_2 of the attributes a. We choose a keyword to represent' attribute a and find a method to replace user H's a_2 in the second social network with user B's a_1 in the first social network. We also take place of user F's b_1 with L's b_2. Finally, based on the result of first two steps, we supplement vertices' attributes(D's b, e), edges(edge from B to D) and vertices(M), and combine the two graphs into a new one. In this way, we can make full use of users' information over heterogeneous social networks.

4.2 User Identity Linkage

We could utilize one individual's profile to link his accounts in different social networks. We all know that several profile attributes such as, location, job, age, etc. could reveal users. For one thing, a real natural person can select different attributes in different online social network. For another, the information is not always publicly available in the two different social networks. Thus, it is difficult for us to choose profile to compare.

Labitzke11 [21] uses the name as the profile to solve the problems. It is a extraordinary character for user identity linkage. For one thing, friends' name strings of users are always publicly available in most social networks. For another, users could not change their friends' name by purpose. In addition, the method simply comparing names is suitable for online search. Thus, we choose name strings of user's friends as their profile and use overlap of friends' name strings as metrics to measure the similarity of two users.

Firstly, we investigate a pruning algorithm to get candidate users. We search communities based on query user, so it is no need to compare all the users in social networks. We choose users in 1-ADcore to compare with others. 1-ADcore is a compact subgrah based on query user, which could be described in the next chapter. We compare a user in 1-ADcore of the first social network with every ones in 1-ADcore of the second social network, which is called a comparison set. In this way, we will get n comparison sets. n is the number of users in 1-ADcore of the first social network.

Then, we choose the two users of the comparison' friend lists as the character of the profile to compare. We take one comparison set as an example. We will compute all the overlap of the comparison set' friend lists. We choose largest overlaps and distinction distance as a correlation metric to judge. As shown in Fig. 2, the gap between the target comparison' overlap value and the next lower one is called the distinction distance. Only if the two compared users' overlap in the comparison set is maximum and distinctive distance is greater than the threshold θ, we get the match finally. We deal with n comparison sets in the same way.

4.3 Filtering Attributes

As mentioned above, a real natural person can choose different attributes in different online social network sites, due to different usage scenarios. For matched

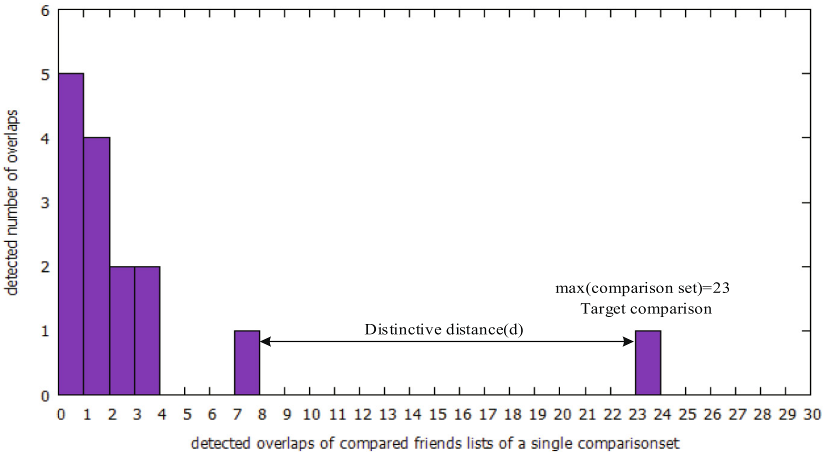


Fig. 2. Histogram of detected comparison overlaps for a comparison set(1:15)

users, we may get different keywords, K_1 and K_2, of the same attribute. It is very difficult for us to integrate users' information in different heterogeneous social networks. Thus, we need find some way to filter wrong attributes.

We could take advantage of candidate users' information to filter matched users' wrong attributes. As shown in algorithm 1, M_1 and N_1 is a pair of matched users in two social networks. If the keyword of users' one attribute is different from the other, we call the keyword of the same kind of attributes as $M.k(j)$ and $N.k(j)$, and filter keyword of the attribute. Firstly, we get al.l candidate users' keyword set of this attribute, as $attribute(j)$. If the M_1's keyword of the attribute $M.k(j)$ in $attribute(j)$, is more than $M.k(j)$'s number. We choose this first keyword to represent the person's attribute. If not, we choose the second keyword. In this way, we enhance the accuracy of users' attributes.

4.4 Combination of Social Networks

Based on the pairs of matched users and the filtered users, we can create a new social network which contains all the users' information. We supplement users, users' relationships and users' attributes from the second social network to the first one.

As shown in Fig. 1(c), we first supplement vertices M with its attributes from the second social network to the first one because user M is the user of 1-ADcore in the second social network, which means M is of great importance and should not be ignored. Secondly, we supplement vertices' edges, which means relationships between users. We add the edge from B to D, the edge form F to D from the second social network to the first one. At last, we supplement vertices' attributes(D's b, e, F's e) from the second social network to the first one. In this way, we can get a new social network containing more information and search communities in the new one.

Algorithm 1. Filtering attributes

```
filterAttribute(m) is the function that filter matched users' attribute.
Input: K-ADcores of two social networks, $G_1 = (V, E, A, N)$ and $G_2 = (V, E, A, N)$,
    two sets of matched users $M$ and $N(M_1$ and $N_1$ is a pair of match users in two
    social networks),
    two sets of keyword of the same kind of attributes $M.k(j)$ and $N.k(j)$(if there are
    same kind of attributes);
Output: matched users $M_{SN1}$ and $M_{SN2}$;
for each $M_i$ and $N_i$ do
    for each key word $M_i.kw(j)$ and $N_i.kw(j)$ do
        for each user $U_k \in G_1$ and $G_2$ do
            if $U_k.kw(j)$==$M_i.kw(j)$ then
                numMKeyWord++;
            end if
            if $U_k.kw(j)$==$N_i.kw(j)$ then
                numNKeyWord++;
            end if
        end for
        if numMKeyWord $\geq$ numNKeyWord then
            $N_i.kw(j)$=$M_i.kw(j)$;
        else
            $M_i.kw(j)$=$N_i.kw(j)$;
        end if
    end for
end for
return $M$, $N$;
```

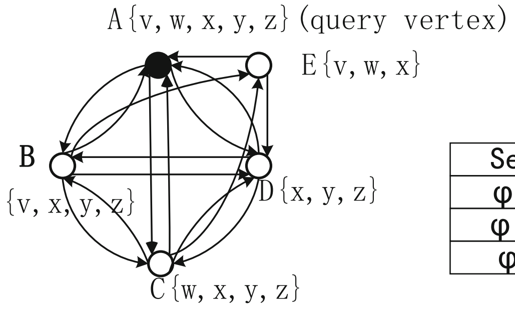
5 Retrieval of Communities

In this section, we demonstrate how to get k-ADcores in directed attributed graphs, as the communities of the query request, and introduce pruning algorithms to reduce processing time.

5.1 Generating Candidate Attribute Combinations

A straightforward method to retrieve compact subgraphs with a attribute set is that we enumerate all the subset of query user's attributes. And find the subgraphs of all the attribute combinations. It could waste a lot of time. Thus, we can use frequent pattern mining algorithms to get candidate attribute combinations.

For an example, S is the attribute set of query user q. As defined above, k is in-degree and out-degree's minimum of every users in communities. Thus, if $S'(S' \subseteq S)$ is a qualified attribute set, then there are at least k of qs neighbors containing set S'. Based on this thought, we apply frequent subset mining algorithm FP-growth to find the frequent attribute combinations, where attributes correspond to item sets. We could retrieve different possible attribute combinations and related attributed subgraphs, instead of enumerating all the attribute combinations. As shown in Fig. 3, we get attributes combinations. The min-degree k is 3, which means there should be 3 neighbors of query user q own the attribute combinations. According to FP-growth algorithm, we first get the set of attribute combinations φ_1, in which the size of attribute combination is 1, Then, we get φ_2 based on φ_1, and retrieve φ_3 based on φ_2. We could not get φ_4, which means there are not attribute combination of size 4 in query user q and his neighbors.



(a) query vertex and its neighbors

k=3

Set	attribute sets
φ_1	{v}, {w}, {x}, {y}, {z}
φ_2	{x, y}, {x, z}, {y, z}
φ_3	{x, y, z}

(b) attribute combinations

Fig. 3. Attribute combinations

5.2 Cores Decomposition

For the question of retrieving communities in the directed graph, it is important to guarantee the subgraphs are strongly connected. We apply Tarjans strongly connected components algorithm [22], which runs in linear time, into our core decomposition algorithm. We use the algorithm to deal with the subgraphs with attribute combination.

In Fig. 4, given a graph G, we order the vertices increasingly according to their min-degree. We first get strongly directed component G_1 of G. We record vertices in G_1 and move away the vertices with min-degree 1. The left subgraph is called G_1'. Then, we get strongly directed component G_2 of G_1'. We record vertices in G_2 and move away the vertices with min-degree 2. We deal with the graph iteratively, and finally retrieve one 1-Dcore, one 2-Dcores, and two 3-Dcore. If we want one 4-Dcore, we can retrieve it from the 3-Dcore, which is strongly connected with query vertex. The strongly connected algorithm prune the other 3-core. As shown in Fig. 4(b), we could use the result of cores decomposition to construct an index, for enhancing efficiency.

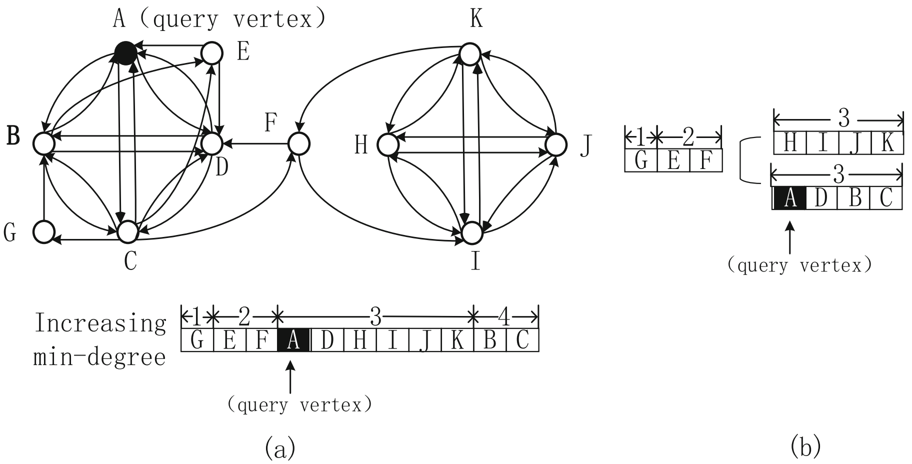


Fig. 4. Cores decomposition

6 Cost Analysis

In the first process of user identity linkage, we get matched users by simply comparing name strings. We can regard that we can finish matching users in constant time a. After that, we filter vertices' attributes by comparing keyword string of the attribute. we can also regard it as constant time a. Based on the result above, we combine two social networks into one in $O(m+n)$. In the process of communities' retrieval, attribute combinations could be done in constant time a. Core decomposition [22] can be done in $O(m)$ and Tarjans algorithm used to verify strongly connected component costs $O(m+n)$, which is applied in different core numbers. k_{max} maximum of structure cohesiveness, the total time cost is O$((k_{max}+1){\cdot}m{+}2{\cdot}n)$.

7 Experiment Evaluation

We now present the experimental results of our method, IC-ADH. In Subsect. 7.1, we introduce our experimental setup. In Subsect. 7.2, we demonstrate our measurements of the effectiveness and efficiency and two other method, Local search and Global search to compare with us. Then, we show our method effectiveness result in Subsect. 7.3 and efficiency result in Subsect. 7.4. All the result are compared with others.

7.1 Experimental Setup

In this section, we introduce the data sets which we user to conduct our experiments. The data set are crawled from the social networks, Twitter and

Foursquare. Furthermore, the data sets were gathered in Singapore, from 2014.11 to 2016.1.

The data set contain a great deal of information for the users. The data sets contain users, relationships between the users, attributes of users and name string of users' friend list. The scale of two data sets of Twitter and Foursquare in Table 1.

Table 1. Datasets of the networks

Dataset	Vertices	Edges
Twitter	160,338	2,405,628
Foursquare	76,503	1,531,357

We randomly selected 200 query vertices and record the average value for every experimental results to guarantee that our experiments could not be influenced by special cases.

Our methods were implemented on a machine with CPU Inter(R) Core(TM)i7-2600, 8.00 GB memory, 3.40 GHz frequency, 500 GB hard disk. All programs are coded in Java.

7.2 Evaluation Methodology

We design two measurement CMF, CPJ to value the effectiveness of the communities search methods. And then we conduct experiments of classical algorithms and our method IC-ADH. We compare IC-ADH's experimental result with others', measured by CMF, CPJ, respectively.

7.2.1 Compared Methods

Global search [2], Local search [3] are the classic community search methods. We choose these two methods and introduce the methods below.

- Global search [2]: Global search is the first method for community search which proposes a measure of density based on minimum degree and distance constraints. It modifies the greedy algorithm and present two heuristic algorithms that find communities of size no greater than a specified upper bound. In this way, Global search yields a compact subgraph based on the query vertex.
- Local search [3]: Local search upgrades Global search to retrieve communities based on query vertex, by using local expansion techniques. It investigates sufficient conditions for deciding whether a neighboring vertex should be added to expand the community. Thus, it save a great deal of time to get communities. In the worst case, its evaluation may become as costly as performing a global search.

7.2.2 Evaluation Measurement

To measure effectiveness of communities, we apply CMF (community member frequency) and CPJ (Community pair-wise Jaccard).

CMF uses the occurrence frequencies of query vertex qs attributes in C_i to measure degree of cohesiveness. CMJ computes how similar the attribute of any pair of vertices of communities. In this way, we can get effectiveness of the communities. The higher their values are, the more cohesive is a community, which ranges from 0 to 1.

- CMF: $CMF(C(q)) = \frac{1}{\psi\cdot|Attr(q)|}\sum_{i=1}^{\psi}\sum_{h=1}^{|Attr(q)|}\frac{f_(i,h)}{|C_i|}$

In the formula, C_i denotes one of the communities retrieved by the method. $Attr(q)$ is the set of query vertex q's attributes, and $f_{(i,h)}$ is the number of vertices of C_i whose attribute contain the h-th attribute of Attr(q). $\frac{f_(i,h)}{|C_i|}$ is the relative occurrence frequency.

- CPJ: $cpj(C(q)) = \frac{1}{\psi}\sum_{i=1}^{\psi}\left[\frac{1}{|C_i|^2}\sum_{j=1}^{|C_i|}\sum_{k=1}^{|C_i|}(\frac{|Attr(C_(i,j))\cap Attr(C_(i,k))|}{|Attr(C_(i,j))\sqcup Attr(C_(i,k))|})\right]$

In the formula, C_i denotes one of the communities we retrieve by using the method. $C_(i,j)$ is the j-th vertex of C_i. $Attr(C_(i,j))$ is the set of vertex $C_(i,j)$'s attributes.

7.3 Effectiveness Evaluation

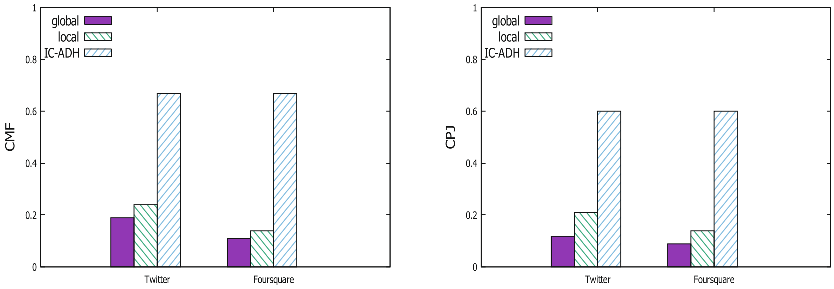
We did a lot of experiments to measure effectiveness of three method, Global search, Local search and IC-ADH, based on measurement CPJ, CMF. We first demonstrate how we choose the value of min-degree k, in Subsubsect. 7.3.1. Then, we use column graphs to show the result of experiments and compare three methods, in Subsubsect. 7.3.2.

7.3.1 Effect of Min-degree *k*

All of the three methods propose the definition of minimum degree measure, which is the threshold of the communities we retrieve from graphs and represents the cohesiveness of the communities. In our method, we define Min-degree k. If the threshold k is higher, the vertices in the communities is more related to the query vertex. A very significant point is that min-degree k we select have a great influence on our experiment result. If k is small, the communities we get could be easily influenced by noise, and it makes no sense in practical application for query vertex. Thus, min-degree k should be higher than 5 in our experiments, which could represent ordinary users and avoid noise.

7.3.2 Effectiveness Comparison

Global search and Local search identify a connected subgraph undirected non-attributed graphs, based on the query vertex. Our method IC-ADH is also based on query vertex, but it works in directed attributed graphs, and also takes full use of information over heterogeneous graphs. The three methods all adopt minimum degree measurement to guarantee the structure cohesiveness of communities. Based on different query vertices with the min-degree higher than 5, we did the experiment for three methods in 200 times. And then, we retrieve communities and compute the average of their CPJ and CMF values.



(a) CMF

(b) CPJ

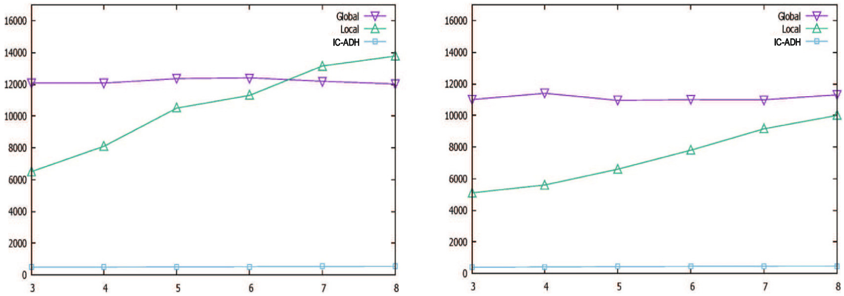
Fig. 5. Effectiveness of the algorithms

Figure 5 shows that the CMF and CPJ values of the algorithms. We can learn that cohesiveness of communities generated by IC-ADH is better than others. For one thing, we make full use of direction of relationships and users' attributes. For another, we deal with users' information in heterogeneous graphs. Considering that IC-ADH utilize more information for the vertices, IC-ADH outperform other online community identification methods.

7.4 Efficiency Comparison

In this part, we do experiments to measure efficiency of three methods, and compare the three methods' efficiency. As mentioned above, we choose min-degree higher than 5. Under different threshold of community cohesiveness k, we did the experiment for every method's 200 query vertices. We compute the average of the time to retrieve communities.

In Fig. 6, we can see Local search outperforms Global search in general, because it uses local expansion techniques to enhance the performance. and it is apparent that IC-ADH executes more efficiently than others because of index construction. Local search and Global search can only get one compact subgraph based on query vertex, which is not efficient.



(a) Twitter (b) Foursquare

Fig. 6. Efficiency of the algorithms about different k

8 Conclusion

In this paper, we investigate the problem of identifying communities online in attributed directed heterogeneous graphs. We propose a novel framework IC-ADH to retrieve effective community. To the best of our knowledge, this is the first work on identifying communities online in attributed directed heterogeneous graphs. As shown in experiments, IC-ADH method performs better than others in efficiency and more effective.

Acknowledgments. This work was supported by National Key R&D Program of China (No. 2018YFC0308205).

References

1. Wang, Z., Ye, Y., Wang, G., Qin, H., Ma, Y.: An effective method for community search in large directed attributed graphs. In: International Conference on Mobile Ad-hoc and Sensor Networks (2017)
2. Sozio, M., Gionis, A.: The community-search problem and how to plan a successful cocktail party. In: SIGKDD (2010)
3. Cui, W., Xiao, Y., Wang, H., Wei, W.: Local search of communities in large graphs. In: SIGMOD (2014)
4. Cui, W., Xiao, Y., Wang, H., Lu, Y., Wei, W.: Online search of overlapping communities. In: SIGMOD (2013)
5. Huang, X., Cheng, H., Qin, L., Tian, W., Yu, J.X.: Querying k-truss community in large and dynamic graphs. In: SIGMOD (2014)
6. Li, Y., Jing, C., Liu, R., Wu, J.: A spectral clustering-based adaptive hybrid multi-objective harmony search algorithm for community detection. In: Evolutionary Computation (2012)
7. Fang, Y., Wang, Z., Cheng, R., Wang, H., Jiafeng, H.: Effective and efficient community search over large directed graphs. IEEE Trans. Knowl. Data Eng. **31**(11), 2093–2107 (2019)
8. Fang, Y., Cheng, R., Luo, S., Jiafeng, H.: Effective community search for large attributed graphs. Proc. VLDB Endowment **9**(12), 1233–1244 (2016)

9. Huang, X., Lakshmanan, L.V.S.: Attribute-driven community search. Proc. VLDB Endowment **10**(9), 949–960 (2017)
10. Nie, Y., Yan, J., Li, S., Xiang, Z., Li, A., Zhou, B.: Identifying users across social networks based on dynamic core interests. Neurocomputing **210**, S0925231216306178 (2016)
11. Zhang, Z., Liu, L., Shen, F., Shen, H.T., Shao, L.: Binary multi-view clustering. IEEE Trans. Pattern Anal. Mach. Intell. **41**(7), 1774–1782 (2018)
12. Goga, O.: Exploiting innocuous activity for correlating users across sites. In: International Conference on World Wide Web (2013)
13. Zhou, X., Liang, X., Zhang, H., Ma, Y.: Cross-platform identification of anonymous identical users in multiple social media networks. IEEE Trans. Knowl. Data Eng. **28**(2), 1 (2016)
14. Zafarani, R., Tang, L., Liu, H.: User identification across social media. ACM Trans. Knowl. Discov. Data **10**(2), 1–30 (2015)
15. Shmatikov, V., Narayanan, A.: De-anonymizing social networks. In: 30th IEEE Symposium on Security and Privacy (2009)
16. Korula, N., Lattanzi, S.: An efficient reconciliation algorithm for social networks (2014)
17. Perito, D., Castelluccia, C., Kaafar, M.A., Manils, P.: How unique and traceable are usernames? (2011)
18. Zafarani, R., Liu, H.: Connecting users across social media sites: a behavioral-modeling approach. In: KDD (2013)
19. Yong, X., Zhang, Z., Guangming, L., Yang, J.: Approximately symmetrical face images for image preprocessing in face recognition and sparse representation based classification. Pattern Recogn. **54**, 68–82 (2016)
20. Meira, W., Jr., Malhotra, A., Totti, L.: Studying user footprints in different online social networks. In: ASONAM (2012)
21. Vosecky, J., Dan, H., Shen, V.Y.: User identification across multiple social networks. In: International Conference on Networked Digital Technologies (2009)
22. Zaversnik, M., Batagelj, V.: An o(m) algorithm for cores decomposition of networks. arXiv Preprint, p. 0310049 (2003)

Predictive Analytics

MPB: Multi-Peak Binarization for Pupil Detection

Chengkun He[1](✉), Xiangmin Zhou[1](✉), and Chen Wang[2]

[1] School of Science, RMIT University, Melbourne, Australia
{s3730729,xiangmin.zhou}@rmit.edu.au
[2] CSIRO Data61, Canberra, Australia
chen.wang@data61.csiro.au

Abstract. Automatic pupil detection is a fundamental part of eye-related tasks like eye tracking, gaze estimation and eye movement identification. Especially, in ophthalmology, to provide assistance and fulfil the demand of diagnosis and treatment, an accurate and real-time algorithm is required. In this paper, we propose a fast and robust Multi-Peak Binarization (MPB) based method for pupil detection in ophthalmology scenarios. A novel strategy for region of interest and candidate connected area detection is presented. Constraints for pruning the irregular shapes and accelerating the MPB algorithm are defined. The proposed method is evaluated on an open-dataset and the experimental results demonstrate the high performance of our approach.

Keywords: Pupil detection · Pupil center · High speed

1 Introduction

Robust and accurate detection of the pupil position plays an important role in the eye tracking and its applications like gaze-based interaction [19] or driving assistance [11,14]. For instance, if a driver gets drowsy, it can be discovered easily by detecting the pupils and tracking eyes. If there is a distraction, something catches the eye of driver, the alarm will be released according to the location of pupil. Detection pupil is also the first step in gaze interaction. Only if the pupil is located, it could be possible to make further exploration.

We study the problem of pupil detection in ophthalmology. Locating the pupil is the fundamental step for ophthalmologic equipment like OCT(optical coherence tomography), fundus camera, tonometer and so on. For the data captured in ophthalmology scenarios, it is challenging to detect the pupil with variety caused by illumination, occlusions, human anatomical eye variability, as well as other sources of noise. Besides, speed of operation and hardware cost should be also considered. To address the pupil detection, there are three mainstream methods: shape-based [8,12,17], appearance-based [1,9,15] and neural network-based [3,20] approaches. Shape-based methods are constructed from either the local point features of the eye and face region or from their contours. The suitable

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 283–298, 2020.
https://doi.org/10.1007/978-3-030-65390-3_22

features can be edges, eye corners, or points selected based on specific schemes. Shape-based approaches have advantages. With elaborate handcrafted design, these approaches get great performance under the given conditions, laboratory conditions. However, the suitable feature is not always obvious. For example, the image could possess low average pixel value, and this makes edge detector preform poorly. The appearance-based methods, which are also known as image template or holistic methods, rely on models built directly on the appearance of the eye region. These methods detect and track eyes directly, based on the photometric appearance as characterized by the color distribution or filter responses of the eye and its surroundings [7]. Binarization is an efficient way to segment the image and locate the pupil [13]. To increase the robustness, schemes are design to select threshold adaptively. However, it is strongly influenced by illumination in ophthalmology. When the adaptivity cannot cover the fluctuation of the illumination, selected threshold will product unwished binarization. In recent years, convolutional neural network (CNN) is adopted in the pupil detection. Considering the runtime and hardware dependency, CNN-based approaches will not fit the situation. Therefore, pupil detection in ophthalmology is still a challenging task and further investigation is required.

To overcome the problem of existing pupil detection approaches, we propose a multi-peak binarization strategy. Multiple binarized images are generated to enlarge the possibility of containing the pupil. First, histogram is built on the image, then multiple extremum bins are selected as thresholds. Adaptivity of this process increases the robustness against illumination.

In this paper, the proposed method contains two phases: regions of interest (RoI) detection and ellipse detection. The input image is scaled down first, and multi-peak binarization (MPB) is adopted to generate multiple binarized images and a set of RoIs is detected. After that, the MPB strategy is utilized again for each RoI to find out connected areas. Ellipses are fit to the filtered ares and one ellipse is selected as pupil based on the defined similarity. Experiments on datasets have been conducted to validate our proposed method and the promising results demonstrate its performance. The main technical contributions of our work are:

- We propose a fast coarse detection model to locate pupil, namely RoI detection. The image is scaled down to a mini size. Although much information loses, edges get sharpened. And it leads to better performance of binarization.
- We propose a novel binarization strategy, which is a key to both coarse and fine pupil detection. With the adaptivity, it improves the robustness of algorithm against noise.
- We define the shape-based conditions and constraints to accelerate the algorithm. Experimental results demonstrate the effectiveness.

The rest of the paper is organized as follows. In Sect. 2, we review some related work about pupil detection. Section 3 describes our method in detail. We present the experimental results and give analyses in Sect. 4, and finally we draw a conclusion and discuss about future work in Sect. 5.

2 Related Work

There is a huge number of work for pupil detection, we recommend works [4, 18] for a further appraisal. As mentioned above, much noise might occur in ophthalmology scenarios. So, methods with robustness will be introduced.

SET [10] first segments the image by a threshold and contours of segments are extracted. After fitting ellipses, the one which is most like a circle is selected as pupil. Only one fixed threshold cannot deal with variety of image generated in ophthalmology scenarios. When the image is relatively dark, shadows, eyelid, even skin could have pixel value similar to pupil. Binarizing by a large threshold, the resulting segment might contain noise. On the other hand, too small threshold could miss the area containing pupil. So, in this work, we design an MPB strategy to select multiple thresholds, generate multiple binarized images and enlarge the possibility of containing the pupil. ExCuSe [6] detects the reflection first. If no reflection found, a coarse pupil position will be selected and then it gets refined by intensity values; If a reflection is detected, an edge detector is adopted followed by morphological operations. Ellipses are fitted from the filtered edges. The one whose enclosed intensity is darkest will be selected. ElSe [5] is extend from ExCuSe. It detects edges by Canny detector. The resulting edges are filtered through morphological operations. Fitted ellipses are selected by roundness and enclosed intensity value. If no pupil detected, a center surround filter is utilized to find a coarse pupil. With an adaptive threshold, an area in the coarse pupil will be revealed and its center will be regarded as the pupil center. As mentioned above, edge detector is strongly influenced by illumination. In ophthalmology, it is hard to set a general threshold for the edge detector. Vera-Olmos [20] has 2 deep CNNs. One is for a coarse estimation and another one is for a fine estimation. It outperforms the state of the art in pupil detection. However, the model only runs at 25∼28 fps with GPU.

In this work, we aim to detect pupil in ophthalmology. The captured image possess relatively low average pixel value and is influenced by noise like light blob and occlusion. The runtime of algorithm is demanded for the real-time usage.

3 Proposed Method

We build the model as shown in Fig. 1. Similar to previous work, two phases, coarse and fine detection, are adopted: (i) find the regions that contain the eye; (ii) detect and select an ellipse as the approximation of pupil. An elaborate strategy is designed: we make multiple binarizations for the same image with adaptive thresholds, efficient conditions and constraints prune the redundant and unwished areas and speed up the algorithm. Specifically, in the first phase, the input image is scaled down to a mini image. Then MPB is adopted: the histogram of scaled image is accumulated and multiple extremum bins (peaks) are selected, which leads to binarizing the mini image with corresponding pixel

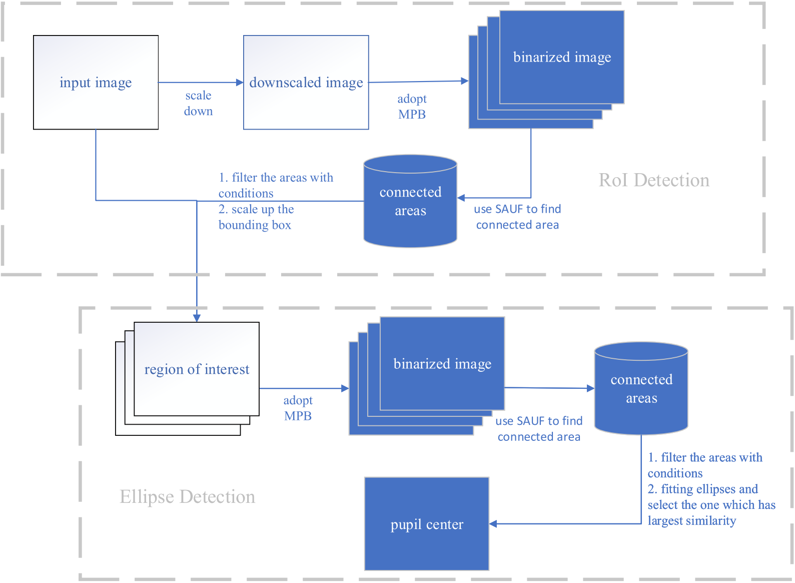

input image
scale down
downscaled image
adopt MPB
binarized image
1. filter the areas with conditions
2. scale up the bounding box
connected areas
use SAUF to find connected area
RoI Detection
region of interest
adopt MPB
binarized image
connected areas
use SAUF to find connected area
1. filter the areas with conditions
2. fitting ellipses and select the one which has largest similarity
Ellipse Detection
pupil center

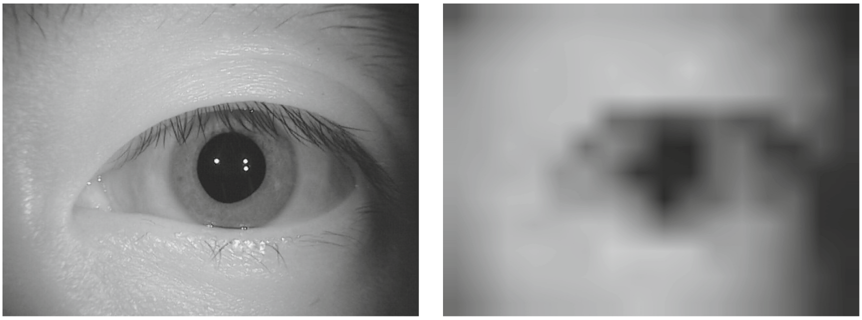
Fig. 1. The detection rate of 60 samples for 120,000 images.

values. Subsequently, RoIs for the original image are selected by conditions. For the ellipse detection and selection, we utilize MPB again on the RoIs. It generates multiple binarized images and connected areas within them. After being filtered by constraints, only connected areas which are similar to ellipse are reminded as candidates. For each candidate area, fitting ellipse is done and then we compute the similarity between the contours of the area and the fitted ellipse. According to the similarity, the ellipse is selected for approximating the pupil.

3.1 MPB for Region of Interest Detection

To detect RoIs fast, the input image is scaled down to a $W_{min} \times H_{min}$ mini image. As shown in Fig. 2(b), scaled image loses most information but it still infers the location of eye. Compared with the original image, scaling-down sharpens the edges in the image. What's more, it costs far less to compute on the mini image for MPB-based RoI detection. We calculate the histogram for the mini image as shown in Fig. 3. Because the area of eye gets relatively low grey scale in ophthalmology scenarios, then $MultiPeak_1$ extremum bins from left to right are selected as thresholds. Binarization is adopted according to these histogram values.

Connected areas are found on the each binarized mini image (Fig. 4(a)). SAUF algorithm [21] is adopted for 4-way connectivity to find the connected areas. We filter these areas by conditions. Bounding rectangles of the filtered



(a) Original image

(b) Scaled image

Fig. 2. Example of scaling down. The original image is 640 × 480 and is scaled to 16 × 12. For presentation, we re-scale the image to 640 × 480.

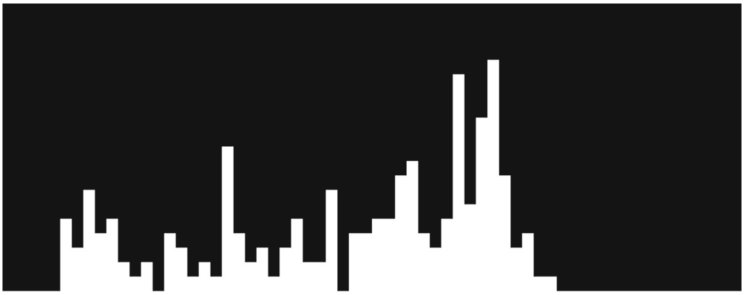
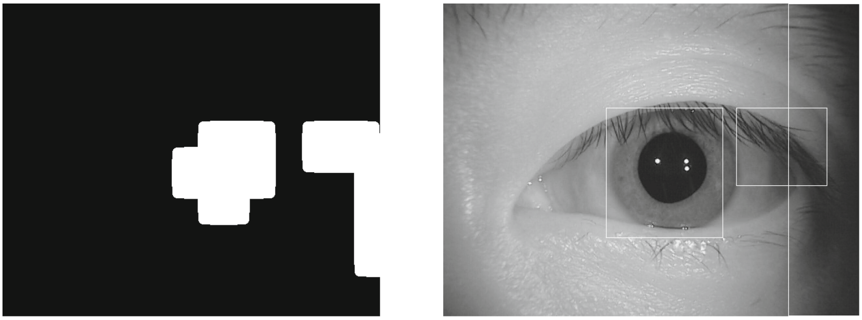


Fig. 3. The histogram of scaled image.

connected areas can be reflected on the original image (Fig. 4(b)), which are regarded as candidate regions, RoIs.

3.2 MPB for Ellipse Detection

After we get candidates regions which might include the eye, ellipse detection will run to finely locate the pupil. For each RoI, MPB is adopted again. Histogram is computed on the candidate region. Multiple binarizations get done for the $MultiPeak_2$ selected peaks and M connected areas can be found, $M = \sum_{i=1}^{MultiPeak_2} m_i$ and m_i is the number of connected area for the i^{th} binarized image. Constraints are utilized to prune the unstable area. The rest areas are overlapped together as the candidate areas. With this process, only areas which are like ellipse are reminded. And it also reduces the computational requirements for ellipse fitting. An example of this process is presented in Fig. 5.



(a) Binarized image

(b) RoIs

Fig. 4. Detecting region of interest. (a) is one of the re-scaled binarized image and there are 3 connected area (ignoring the background). The bounding boxes of these areas are reflected on the original image and generate region of interest like the (b) shows.

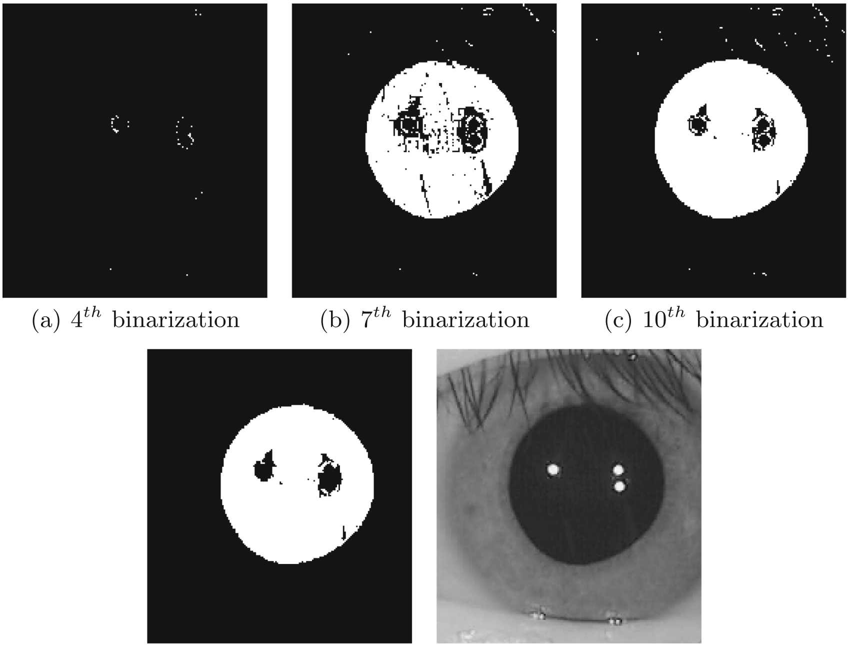
Algorithm 1: Similarity measure

```
Input: Contours $S_1$ and $S_2$.
Output: Similarity $Sim$.
Initialize the average distance $dist_{aver} = 0$;
for $\forall q \in S_1$ do
    Find the $p \in S_2$ which has the shortest distance to $q$;
    Accumulate the distance to $dist_{aver}$;
end
Compute the average distance $dist_{aver} = dist_{aver}/length(S_1)$;
Compute the similarity $Sim = 1/dist_{aver}$.
```

To fit ellipses, we first pad each candidate area and turn it into a convex polygon, which increases the capability of handling the noise like occlusion and light blob. Contour of the area can be extracted easily as a set of 2D points, S_1. Then the ellipse that fits the contour is calculated by the algorithm [2]. According to the ellipse information, a set of 2D points S_2 on the ellipse boundary is generated. For each point p in S_1, we find a point q in S_2 which gets the shortest distance from p. Then an average distance $dist_{aver}$ is computed to measure the similarity between S_1 and S_2. The area having the minimal $dist_{aver}$ is selected and the corresponding ellipse is regarded as the approximation of pupil as shown in Fig. 6. This process is presented in Algorithm 1.

3.3 Conditions and Constraints

In this section, we discuss the conditions and constraints in RoI detection and ellipse detection to prune redundant and unwished areas.



(a) 4^{th} binarization (b) 7^{th} binarization (c) 10^{th} binarization

(d) candidate areas (e) RoI

Fig. 5. Detecting candidate areas.(a)–(c) are the binarized images. (d) is the detected candidate area. (e) is the RoI. Comparing (d) with (c), after filtering, noise (separated points) is excluded.

- In RoI detection, if the connectivity of the area in binarized image is less than min_area_1, then we drop it out; In ellipse detection, if the connectivity is less than min_area_2, the area is dropped out. By this condition, noise like light blob can be filtered, which reduces the number of connected area and accelerates the following operation.
- In ellipse detection, if the connected area intersects the boundary of RoI, it might infer that the current area is a sub set of a larger connected area which will be handled later. And it also prunes some area caused by shadow.
- In the ophthalmology scenarios, the width cannot be too larger than the height of area bounding box, and vice versa. This constraint guarantees the aspect ratio of fitted ellipse will not get too large or too small, which means the ellipse is more like a circle.
- In ellipse detection, if the area of connected pixel is less than the area of its bounding box, say $ConnectedArea < areaRatio \cdot BoundingBox$, then it will be dropped out. This process filters some irregular areas.

MPB strategy enlarges the possibility of find a suitable binarization for pupil detection, while increasing the computational requirements. However, conditions and constraints mentioned above prune candidate areas and accelerates the operation. Combination of two parts guarantees the performance with high speed.

4 Experiments

In this section, we present the dataset information for the experiments and discuss the selection of parameters. Then the performance of the proposed method and analyses are given.

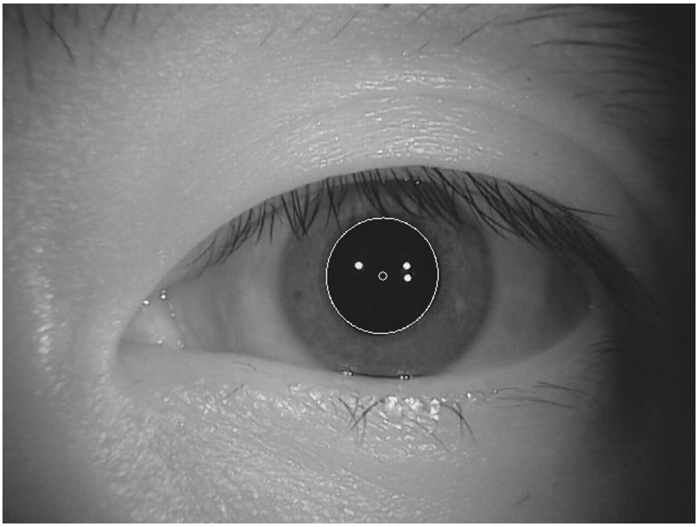


Fig. 6. The approximation of pupil.

4.1 Experimental Setup

We perform the experiments on datasets: labelled pupils in the wild (LPW) [18]. It contains data captured by Pupil Pro High Speed Eye Camera (120 Hz) from 22 persons. For each object, three videos are contained with different recording environment tags. Especially, the individual wear no glasses in ophthalmology diagnosis and treatment, so the samples tagged Glasses will be excluded in our experiment[1]. More details are presented in Tables 1 and 2.

The experiments was performed on a CPU i5-8300H @2.30 GHz with 8 GB RAM under Windows 10.

Table 1. Datasets used in the experiments.

Dataset	Format	Instance	Resolution
LWP	Video	130,856 frames	640 × 480

[1] Object 3 and 5, including 6 samples, are dropped out.

Table 2. Tags for recording environment of LPW.

Tags for recording	Description
Outdoor	True if the corresponding video was recorded outdoors
Natural light	True if the video was recorded in natural light
Artificial light	True if the video was recorded in artificial light
Glasses	True if the participant was wearing glasses during the recording
Contact lensesn	True if the participant was wearing contact lenses during the recording
Prescription	Prescription values of the participant
Nationality	Nationality of the participant
Eye color	Eye color of the participant
Gender	Gender of the participant

4.2 Evaluation Methodology

We report our results in terms of the average pupil detection rate as a function of pixel distance between the algorithmically established and the hand-labeled pupil center. Because the ground truth was labeled by hand, imprecision cannot be avoided. Hence we focus on results where pixel error is greater than 6.

In this experiments, we evaluate the effectiveness by comparing the proposed method with a baseline method which contains only a fixed threshold and another MPB-based method without scaling-down part. When the efficiency is presented, the state-of-the-art pupil detection methods are considered.

4.3 Parameter Selection

Parameter selection can greatly affect the quality of pupil detection in real applications. In this work, we decide the values of parameters involved in the pupil detection including scaled image resolution, numbers of extremum peaks and numbers of histogram bins, by experimental analysis. The minimum connected area min_area_1 and min_area_2 filter little areas which are caused by noise like light blob. Following the setting in [10], we set min_area_1 to 4 and min_area_2 to 600 empirically.

Effect of Image Resolution. We report the results generated on part of the dataset with different image resolution. As shown in the Table 3, when the image is scaled to 16 × 12, we get the best performance. It is intriguing that larger scaled image leads to less time cost. It is because too little scale leads to huge loss of information, and it could return larger RoI after being rescaled on the original image, which takes more time to detect ellipse. Thus, we set the default image resolution for pupil detection to 16 × 12.

Table 3. Effect of image resolution.

Resolution	At 12 pixel	At 14 pixel	At 16 pixel	Runtime
32×24	97.99	98.11	98.25	**4.69 ms**
21×16	97.96	98.10	98.23	5.54 ms
16×12	**98.18**	**98.31**	**98.46**	5.70 ms
13×10	92.65	93.01	93.36	6.05 ms
11×8	93.84	94.43	94.68	6.56 ms

Table 4. Effect of extremum peak number 1.

$MultiPeak_1$	At 12 pixel	At 14 pixel	At 16 pixel	Runtime
5	98.05	98.19	98.3428	**4.54 ms**
7	**98.18**	**98.31**	**98.46**	5.70 ms
9	98.11	98.23	98.37	5.54 ms
11	97.58	97.77	97.93	8.65 ms

Effect of Extremum Peak Numbers. Number of extremum peaks $MultiPeak_1$ in RoI detection and $MultiPeak_2$ in connected area detection control how many binarized images will be generated. We evaluate the impact over our dataset to find the optimal values. According to the Tables 4 and 5, with number of extremum peaks increasing, the time cost gets larger. It is because the more peaks are selected, the more binarized images need to post-process. Taking both effectiveness and effectiveness into consideration, $MultiPeak_1$ and $MultiPeak_2$ are set to 7 and 10 separately.

Effect of Histogram Bin Numbers. For the RoI detection phase, we set $HistNum_1 = 64$ according to Table 6. For the reason that scaled image loses much information, too many bins might lead to the fewer peaks. For instance, the scaled image is 16×12, after computing histogram with 256 bins, most bins might possess only 0 or 1 and it weakens the robustness. On the other hand, for the original image, a precise binarization plays an important role for detecting the edge of pupil. So, we set $HistNum_2 = 256$.

4.4 Effectiveness Evaluation

Effectiveness Comparison. We use the optimal settings and evaluate our final detection effectiveness and compare MPB with: (i) an MPB-based model only contains ellipse detection part (without scaling-down for RoI detection); (ii) a baseline method which binarizes the image with fixed threshold [10], then fits ellipse on the segments. The average detection rate with different pixel error from 2 to 16 is reported in Fig. 7. According to the figure, the MPB-based approaches outperform the baseline. Comparing to using only one fixed threshold, multiple binarization strategy improves the performance. The method without scaling-down gets slightly better detection rate than MPB (85.74/85.53 at 14 pixel error and 87.49/86.94 at 16 pixel error). That is caused by the RoI detection when the pupil is not included.

Table 5. Effect of extremum peak number 2.

$MultiPeak_2$	At 12 pixel	At 14 pixel	At 16 pixel	Runtime
5	97.22	97.56	97.82	**3.97 ms**
10	98.18	**98.31**	**98.46**	5.70 ms
15	**98.19**	98.30	98.33	7.60 ms
20	97.38	97.52	97.57	9.19 ms

Table 6. Effect of histogram bin number 1.

Number	At 12 pixel	At 14 pixel	At 16 pixel	Runtime
80	97.67	97.88	98.04	9.02 ms
64	**98.18**	**98.31**	**98.4**	**5.70 ms**
48	97.24	97.56	97.73	6.05 ms
32	97.20	97.50	97.68	6.56 ms

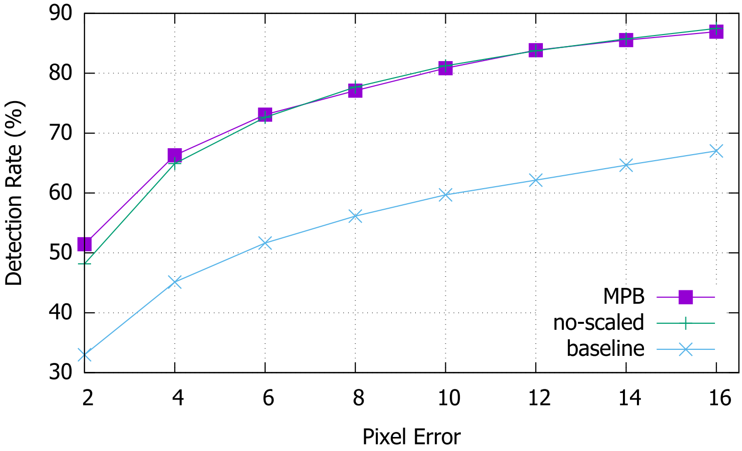


Fig. 7. The average detection rate of 60 samples for 120,000 images at different pixel error.

We select 6 samples for further discussion. As shown in Fig. 8, detection rate gets notable advance from 2 to 8 pixel error and become flat after getting 10 pixel error. This demonstrates the imprecision caused by the hand labelling.

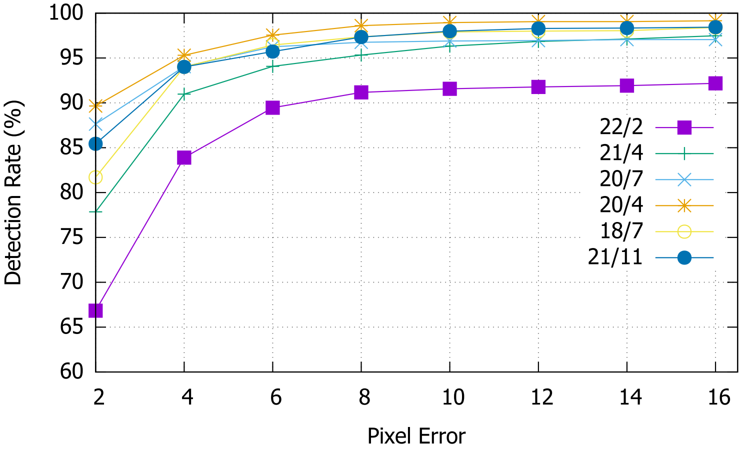


Fig. 8. The average detection rate of 6 samples for 12,000 images at different pixel error.

Table 7. Samples with low detection rate. The average detection rate is report at 10 pixel error.

Videos	Outdoor	Natural light	Artificial light	Detection rate	Average pixel value
10/1	FALSE	TRUE	FALSE	97.22%	112.74
10/8	FALSE	FALSE	TRUE	96.42%	100.934
10/11	TRUE	TRUE	FALSE	**33.58%**	143.734
22/1	FALSE	TRUE	FALSE	**65.01%**	121.275
22/2	FALSE	FALSE	TRUE	92.18%	92.8329
22/17	FALSE	TRUE	FALSE	**20.73%**	119.525

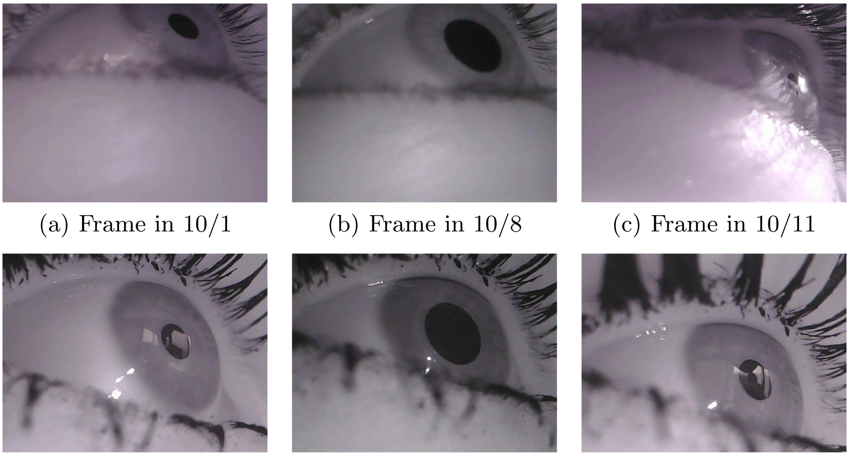
Effect of Illumination. We explore samples with low detection rate. 2 groups of sample are presented first. As shown in Table 7, for the same object, the detection rate can differ markedly. According to the average pixel value, 10/11 gets relatively high value and the worst detection rate, 22/1 and 22/17 are in the same situation. So, the illumination does have effect on the performance.

Specifically, frames extracted from object 10 and 22 are presented in Fig. 9. The strong outdoor natural light causes the reflection in eye (Fig. 9(c) (d) (f)). We compute the mean detection rate of samples tagged *Artificial light* against results generated on the whole dataset (Fig. 10), and it is 92.50% which improves 6.4% (by 86.94%) at 16 pixel error. Hence, the proposed method preform well for the recording with artificial light and that is a common case in ophthalmology diagnosis and treatment.

4.5 Efficiency Evaluation

Runtime is another criterion for the real-time pupil detection. We evaluate the performance on 130,856 images with 640×480, and the average runtime is

7.11 ms with standard deviation 2.16, which meets the real-time requirement (120 fps). The runtime of MPB-based model without scaling-down and the baseline method is also reported. Comparison with several state-of-the-art pupil detection methods is present too in Table 8. Baseline gets the best performance for binarizing with the fixed threshold. Swirski [16] gets the second best performance for the parallelized implement. Considering the image size, MPB gets a comparable performance with ELSe [5] and ExCuse [6]. With out scaling-down, the runtime of MPB-based model is two times larger than the proposed one. It demonstrates that scaling down the image and detecting the RoI reduce the computational requirements. It is worthnothing that Vera-Olmos [20], a GPU-based detection method, gets runtime which is far larger than algorithms running on CPU.



(a) Frame in 10/1 (b) Frame in 10/8 (c) Frame in 10/11

(d) Frame in 22/1 (e) Frame in 22/2 (f) Frame in 22/17

Fig. 9. The influence of illumination.

Table 8. Comparison of runtime. For MPB, ElSe, ExCuSe, Swirski, they were evaluated on a CPU and only Swirski was parallelized. Vera-Olmos is implemented in parallel with GPUs

Methods	Image size	Parallelization	Device	Runtime
MPB	640 × 480	No	CPU	7.11 ms
No_Scaled	640 × 480	No	CPU	14.57 ms
Baseline	640 × 480	No	CPU	2.47 ms
ElSe	346 × 260	No	CPU	6.59 ms
ExCuSe	320 × 240	No	CPU	5.51 ms
Swirski	620 × 460	Yes	CPU	3.77 ms
Vera-Olmos	384 × 288	Yes	GPU	36 ms

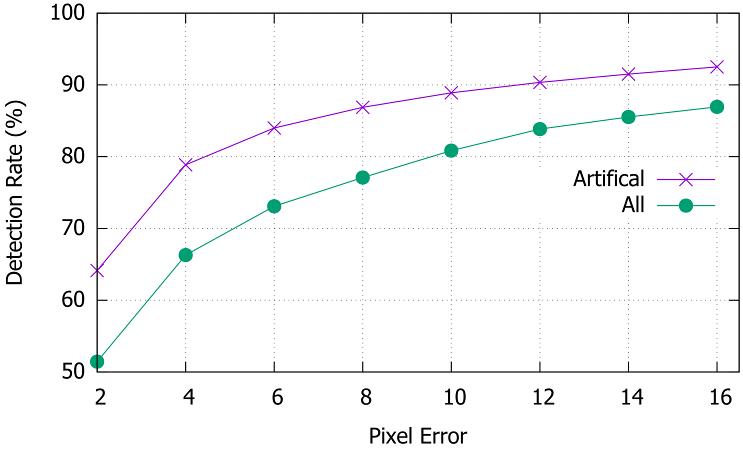


Fig. 10. The detection rate of 21 samples tagged with *Artificial light*.

5 Conclusion and Future Work

In this work, we propose a novel pupil detection method. To fast detect the RoIs, the input image is scaled down. And the proposed MPB strategy is adopted. In the second phase, MPB is utilized again for detecting candidate connected areas. By the similarity measure, we give the fitted ellipse as the approximation of pupil. The experimental results show our proposed method fits well for the ophthalmology cases. Comparing with the baseline method, MPB-based methods outperform in the performance. It demonstrates the effectiveness of MPB strategy. Furthermore, the experimental runtime shows that the proposed method operates in real-time for modern eye trackers (120 fps), which is essential in ophthalmology scenarios.

In the future, we plan to optimize the algorithm to achieve a faster operation. Extending the method to handing pupil detection in pervasive situation is also under consideration.

Acknowledgments. This work is supported by ARC Discovery Project DP200101175 and CSIRO Data61 Grant Australia.

References

1. Fasel, I.R., Fortenberry, B., Movellan, J.R.: A generative framework for real time object detection and classification. Comput. Vis. Image Underst. **98**(1), 182–210 (2005)
2. Fitzgibbon, A.W., Fisher, R.B.: A buyers guide to conic fitting. In: Pycock, D. (ed.), Proceedings of the British Machine Vision Conference, BMVC 1995, Birmingham, UK, September 1995, pp. 1–10. BMVA Press (1995)

3. Fuhl, W., Santini, T., Kasneci, G., Kasneci, E.: PupilNet: convolutional neural networks for robust pupil detection. CoRR abs/1601.04902 (2016)
4. Fuhl, W., Tonsen, M., Bulling, A., Kasneci, E.: Pupil detection for head-mounted eye tracking in the wild: an evaluation of the state of the art. Mach. Vis. Appl. **27**(8), 1275–1288 (2016)
5. Fuhl, W., Santini, T.C., Kübler, T.C., Kasneci, E.: ElSe: ellipse selection for robust pupil detection in real-world environments. In: Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research and Applications, pp. 123–130. ACM (2016)
6. Fuhl, W., Kübler, T., Sippel, K., Rosenstiel, W., Kasneci, E.: ExCuSe: robust pupil detection in real-world scenarios. In: Azzopardi, G., Petkov, N. (eds.) CAIP 2015. LNCS, vol. 9256, pp. 39–51. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23192-1_4
7. Hansen, D.W., Ji, Q.: In the eye of the beholder: a survey of models for eyes and gaze. IEEE Trans. Pattern Anal. Mach. Intell. **32**(3), 478–500 (2010)
8. Hennessey, C., Lawrence, P.D.: 3D point-of-gaze estimation on a volumetric display. In: Proceedings of the 2008 Symposium on Eye Tracking Research and Applications, ETRA 2008, p. 59. Association for Computing Machinery (2008)
9. Huang, W., Mariani, R.: Face detection and precise eyes location. In: Proceedings of the 15th International Conference on Pattern Recognition, ICPR00, Barcelona, Spain, 3–8 September 2000, pp. 4722–4727. IEEE Computer Society (2000)
10. Javadi, A.H., Hakimi, Z., Barati, M., Walsh, V., Tcheang, L.: SET: a pupil detection method using sinusoidal approximation. Front. Neuroengineering **8**, 4 (2015)
11. Kasneci, E.: Towards the Automated Recognition of Assistance Need for Drivers with Impaired Visual Field. Universitt Tübingen, Tübingen (2013)
12. Xu, Y., Zhang, Z., Lu, G., Yang, J.: Approximately symmetrical face images for image preprocessing in face recognition and sparse representation based classification. Pattern Recogn. **54**, 68–82 (2016)
13. Kim, K.N., Ramakrishna, R.S.: Vision-based eye-gaze tracking for human computer interface. In: IEEE SMC 1999 Conference Proceedings, 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028) (2002)
14. Liu, X., Xu, F., Fujimura, K.: Real-time eye detection and tracking for driver observation under various light conditions. In: Intelligent Vehicle Symposium, vol. 2, pp. 344–351. IEEE (2002)
15. Pentland, A., Moghaddam, B., Starner, T.: View-based and modular eigenspaces for face recognition. In: Conference on Computer Vision and Pattern Recognition, CVPR 1994, 21–23 June 1994, pp. 84–91. IEEE, Seattle (1994)
16. Swirski, L., Bulling, A., Dodgson, N.A.: Robust real-time pupil tracking in highly off-axis images. In: Proceedings of the Symposium on Eye Tracking Research and Applications, pp. 173–176. ACM (2012)
17. Tian, Y., Kanade, T., Cohn, J.F.: Dual-state parametric eye tracking. In: Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), pp. 110–115 (2000)
18. Tonsen, M., Zhang, X., Sugano, Y., Bulling, A.: Labelled pupils in the wild: a dataset for studying pupil detection in unconstrained environments. In: Proceedings of the ACM International Symposium on Eye Tracking Research and Applications (ETRA), pp. 139–142 (2016)

19. Turner, J., Alexander, J., Bulling, A., Schmidt, D., Gellersen, H.: Eye pull, eye push: moving objects between large screens and personal devices with gaze and touch. In: Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J., Winckler, M. (eds.) INTERACT 2013. LNCS, vol. 8118, pp. 170–186. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40480-1_11
20. Vera-Olmos, F.J., Malpica, N.: Deconvolutional neural network for pupil detection in real-world environments (2017)
21. Wu, K., Otoo, E.J., Suzuki, K.: Optimizing two-pass connected-component labeling algorithms. Pattern Anal. Appl. **12**(2), 117–135 (2009)

Rice Leaf Diseases Recognition Using Convolutional Neural Networks

Syed Md. Minhaz Hossain[1], Md. Monjur Morhsed Tanjil[1],
Mohammed Abser Bin Ali[1], Mohammad Zihadul Islam[1], Md. Saiful Islam[2](✉),
Sabrina Mobassirin[1], Iqbal H. Sarker[3], and S. M. Riazul Islam[4]

[1] Premier University, Chittagong, Bangladesh
minhazpuccse@gmail.com, taaanjil@gmail.com, abserbinali@gmail.com,
zihadulislam44@gmail.com, sabrinaa.samiaa@gmail.com
[2] Griffith University, Gold Coast, Australia
saiful.islam@griffith.edu.au
[3] Chittagong University of Engineering and Technology, Chittagong, Bangladesh
iqbal@cuet.ac.bd
[4] Sejong University, Seoul 05006, Korea
riaz@sejong.ac.kr

Abstract. The rice leaf suffers from several bacterial, viral, or fungal diseases and these diseases reduce rice production significantly. To sustain rice demand for a vast population globally, the recognition of rice leaf diseases is crucially important. However, recognition of rice leaf disease is limited to the image backgrounds and image capture conditions. The convolutional neural network (CNN) based model is a hot research topic in the field of rice leaf disease recognition. But the existing CNN-based models drop in recognition rates severely on independent dataset and are limited to the learning of large scale network parameters. In this paper, we propose a novel CNN-based model to recognize rice leaf diseases by reducing the network parameters. Using a novel dataset of 4199 rice leaf disease images, a number of CNN-based models are trained to identify five common rice leaf diseases. The proposed model achieves the highest training accuracy of 99.78% and validation accuracy of 97.35%. The effectiveness of the proposed model is evaluated on a set of independent rice leaf disease images with the best accuracy of 97.82% with an area under curve (AUC) of 0.99. Besides that, binary classification experiments have been carried out and our proposed model achieves recognition rates of 97%, 96%, 96%, 93%, and 95% for Blast, Brownspot, Bacterial Leaf Blight, Sheath Blight and Tungro, respectively. These results demonstrate the effectiveness and superiority of our approach in comparison to the state-of-the-art CNN-based rice leaf disease recognition models.

Keywords: Rice leaf diseases · Image recognition · Convolutional neural networks

1 Introduction

Rice, the most important food crop in the world, has always been crucial for global food security and socioeconomic stability. Food security is not just about

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 299–314, 2020.
https://doi.org/10.1007/978-3-030-65390-3_23

fulfilling hunger; it is essential for fulfilling the demand for the nutrition of expanding populations in rice developing countries. About 70% of the population in Asia rely upon a prevalent rice-based diet [1]. Rice gives 15% of per capita protein and 21% of per capita vitality [1]. However, it is unfortunate that all varieties of rice suffer from several diseases and pests. Diseases can affect productivity as well as crop quality. A review conducted in Bangladesh [18] uncovered 20 rice diseases, including two viral, two bacterial, 13 fungal, two nematodes and one micro-nutrient deficient issue. Among the diseases, Bacterial leaf blight, Brownspot, Blast, Tungro and Sheath blight cause substantial loss to rice both in quality and quantity.

The detection of rice leaf diseases is quite a hard task for farmers and experts with naked eyes. Identifying the abnormality in plants with similar symptoms in different disorders is highly challenging. Moreover, these challenges are aggravated by the different backgrounds of images and image capturing conditions. Recent progress in computer vision and deep learning has made it possible to identify the deep features of numerous diseases irrespective of variety in image backgrounds and image capture conditions. There exists many works on the detection and recognition of rice leaf diseases and pests using convolutional neural network (CNN) [2,16,17,21,22]. Moreover, plant leaf diseases have been identified in various works using state-of-the-art CNN architectures such as VGG16 in [3,4,9,13], GoogleNet in [11,19], CaffeNet in [26], ResNet50, ResNet101, ResNet152, Inception V4 in [4], ResNet34 in [3], Student-teacher CNN in [30], AlexNet in [3,9,19] and DenseNet in [3,4]. Recently, a method for recognizing rice leaf diseases using two-stage custom CNN has been proposed in [22] on a dataset from Bangladesh. The major drawback of this work is the manual division process of symptom classes, which is laborious and may cause misclassifications.

Most of the existing CNN-based plant leaf disease recognition models is limited to image capturing conditions and backgrounds. It restricts the disease recognition into the known dataset. For example, the works proposed in [4,9,19,26] and [30] are limited to plain backgrounds. Moreover, tuning better network parameters of the model for recognizing plant leaf diseases still depends on the existing state-of-the-art CNN architectures. All of the architectures achieve better recognition rates, but researchers do not consider the effectiveness of large scale parameters in memory restricted devices such as mobile phones. This is particularly important for farmers living in the rural areas. In addition, CNN-based models are sometimes restricted to generalization, e.g.., whenever new data are included into the dataset its accuracy falls down drastically [9].

In this paper, a novel CNN model is proposed and extensive experiments have been conducted to tune the parameters of the model in recognizing rice leaf diseases. Our custom CNN model is designed by using a number of convolution and pooling layers followed by a dense layer and a softmax layer. Our model significantly reduces the number of parameters, which has the advantage of deploying it in memory restricted devices. A novel dataset containing diverse image backgrounds is used to validate the proposed model. We adopted data augmentation to improve the generalization of our model. An independent set of rice leaf disease images have been tested to evaluate the performance of our model in terms accuracy, precision, recall, F1-score and area under curve (AUC).

The rest of the paper is organised as follows. Section 2 discusses the related works; proposed model for recognizing rice leaf diseases is presented in Sect. 3; experiments, results and observations are illustrated in Sect. 4; and finally, the paper is concluded in Sect. 5.

2 Related Work

There exists many works for identifying and classifying the plant diseases specifically for recognizing the rice leaf diseases [23]. In [2], a CNN-based framework is developed to identify three different rice diseases and healthy images, while in [10] a CNN-based model is used to detect various plant leaf diseases. In [6], principal component analysis (PCA) is applied to remove the redundant information and provide a vector with a reduced dimension for each rice leaf disease image. Moreover, various classifiers were used to evaluate the performances and SVM is found to achieve the best recognition rate of rice leaf diseases. In [29], diagnosis of plant disease based on CNN is achieved by extracting learned features via perturbation, gradient and reference based visualization using InceptionV3. Mixed layer is used to generate deep features like shape, diversity of colors etc. It can remove 75% of parameters without affecting the accuracy and loss. In [16], it is shown that CNN-based model is more effective than traditional feature extractors including LBPH and HaarWT for recognizing rice blast disease.

The authors of [28] used various state-of-the-art CNN architectures such as AlexNet [14], GoogleNet [27] and LeNet [15] for the detection of 10 different diseases of Tomato leaf. They found that LeNet gives the best result in terms of accuracy. In [11], better accuracy is obtained using GoogleNet architecture to recognize plant leaf disease. For detecting plant disease, a transfer learning model is developed in [19] using the state-of-the-art CNN models such as AlexNet and GoogleNet. A deep learning framework CaffeNet [7] manifested by Berkley Vision and Learning Centre, was used to perform the CNN training for the recognition of plant disease [26]. The work in [17] conducts a comparative study between different pooling strategies such as *mean-pooling*, *max-pooling*, *stochastic pooling*, and *gradient descent algorithm* is applied to train CNNs for recognising rice leaf diseases. In [12], the authors calculated the texture features by using color co-occurrence matrix and Naïve Bayes was applied for the classification of plant diseases from leaves. In [13], two deep pre-trained models such as VGG16 [25] and CaffeAlexNet are used for feature extraction of some selected fruit crops diseases and the most discriminated features are used to classify diseases using multi-class SVM. In [22], two state-of-the-art CNN architectures such as VGG16 and InceptionV3, and a two-stage CNN model have been fine-tuned and implemented for rice diseases and pests identification. The authors found the two-stage CNN model to be effective for memory restricted devices. In [21],

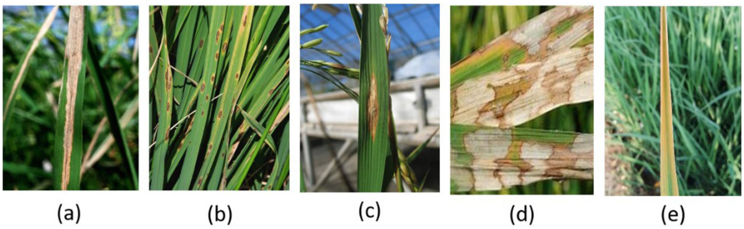


Fig. 1. Some examples of rice leaf disease images with different backgrounds and different types of symptoms: (a) Bacterial leaf blight; (b) Brownspot (c) Blast; (d) Sheath blight; and (e) Tungro

Resnet-34 architecture is used as a transfer learning and cyclical learning rates (CLR) is utilized for the classification of rice plant diseases.

The major challenge of recognizing rice/plant leaf diseases is overcoming the issue of image capturing conditions and backgrounds. These condition restrict model performance, e.g.., some of the existing works are limited to plain backgrounds [4,9,19,26,30] and intolerant to image capturing conditions [16]. Moreover, tuning the large-scale network parameters of the model for recognizing plant leaf diseases depends on state-of-the-art CNN architectures. Though, all of the architectures achieve better recognition rates, they are not effective to be deployed in memory restricted devices due to their large scale network parameters. Finally, CNN-based models for recognizing rice leaf diseases are sometimes restricted to generalization [9].

To overcome the aforementioned issues, we propose a novel CNN-based model by using a number of convolution and pooling layers followed by a dense layer and a softmax layer for recognizing rice leaf diseases. Our custom CNN-based model is designed to reduce the number of network parameters. We have prepared a novel dataset containing diverse image backgrounds and image capturing conditions, and augmented it to improve the generalization of our model. To verify the effectiveness and superiority of our model, it is tested on an independent set of rice leaf disease images.

3 Our Approach

This section describes our proposed method of recognizing rice leaf diseases. The entire process is partitioned into different stages: beginning with the preparation of a novel training dataset, development of a novel CNN model, deep feature extraction for training the model and finally, classification of the rice leaf diseases.

3.1 Dataset

A total of 323 original RGB colored images of five common rice leaf diseases, including Blast, Bacterial leaf blight, Brownspot, Sheath blight, and Tungro are

Table 1. Dataset descriptions of rice leaf disease recognition

Disease Class	#Org. Images	Augmentation Techniques					# Aug. Images
		Rotations	Flipping	Shifting	Scaling	Zooming	
Blast	63	252	126	126	189	63	756
Bacterial Leaf Blight	70	280	140	140	210	70	840
Brownspot	70	280	140	140	210	70	840
Sheath blight	57	228	114	114	171	57	684
Tungro	63	252	126	126	189	63	756
Total	**323**	**1292**	**646**	**646**	**969**	**323**	**3876**

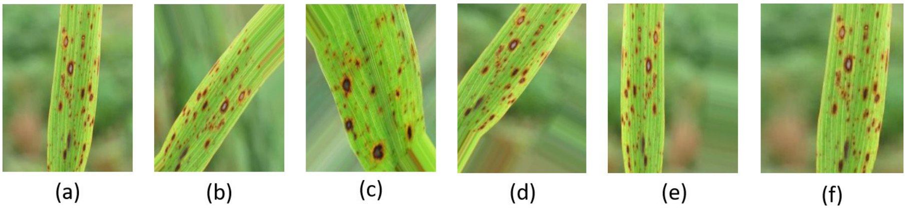


Fig. 2. Augmented images of rice leaf diseases: (a) original Brownspot image; (b) rotated Brownspot image; (c) zoomed Brownspot image; (d) shifted Brownspot image; (e) flipped Brownspot image; and (f) scaled Brownspot image

collected from the International Rice Research Institute (IRRI)[1] and Bangladesh Rice Research Institute (BRRI)[2]. A sample of each class of rice leaf disease is shown in Fig. 1. In all our experiments conducted in this paper, different sizes of images are used to evaluate the performances of recognizing rice leaf diseases. The sizes of the rice leaf disease images are 128×128, 256×256 and 512×512. To tackle the challenge of identifying the best features in different backgrounds, we include natural, plain and complex image backgrounds. Moreover, our experiment includes different types of symptoms: small, large, isolated, and spread. In Fig. 1, five samples are shown in different image backgrounds with various types of symptoms. For example, samples in Fig. 1(a), Fig. 1(b) and Fig. 1(e) are the images of Bacterial leaf blight, Brownspot and Tungro, respectively, which are in natural background. On the contrary, sample in Fig. 1(c) is the Blast image, which is in the complex background, whereas sample in Fig. 1(d) is the Sheath blight image, which is in the plain background. A summary statistics of the original 323 rice leaf dataset is given in Table 1.

3.2 Dataset Augmentation

Fewer amount of training data is a major bottleneck for developing an effective deep learning models including CNN-based models for rice leaf disease recognition [16]. To ensure the robustness of the neural network based model, we need bigger amount of data to expand the functional diversity of the model. For this,

[1] https://www.irri.org/.
[2] https://www.brri.gov.bd/.

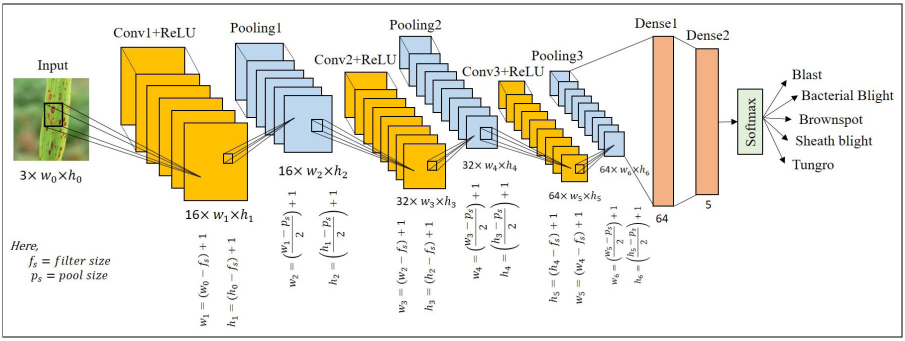

Input
$3\times w_0 \times h_0$
Conv1+ReLU
$16\times w_1 \times h_1$
Pooling1
$16\times w_2 \times h_2$
Conv2+ReLU
$32\times w_3 \times h_3$
Pooling2
$32\times w_4 \times h_4$
Conv3+ReLU
$64\times w_5 \times h_5$
Pooling3
$64\times w_6 \times h_6$
Dense1
64
Dense2
5
Softmax
Blast
Bacterial Blight
Brownspot
Sheath blight
Tungro
Here,
f_s = filter size
p_s = pool size
$w_1 = (w_0 - f_s) + 1$
$h_1 = (h_0 - f_s) + 1$
$w_2 = \left(\frac{w_1 - p_s}{2}\right) + 1$
$h_2 = \left(\frac{h_1 - p_s}{2}\right) + 1$
$w_3 = (w_2 - f_s) + 1$
$h_3 = (h_2 - f_s) + 1$
$w_4 = \left(\frac{w_3 - p_s}{2}\right) + 1$
$h_4 = \left(\frac{h_3 - p_s}{2}\right) + 1$
$h_5 = (h_4 - f_s) + 1$
$w_5 = (w_4 - f_s) + 1$
$w_6 = \left(\frac{w_5 - p_s}{2}\right) + 1$
$h_6 = \left(\frac{h_5 - p_s}{2}\right) + 1$

Fig. 3. Our CNN-based model for rice leaf disease recognition

we adopt the image data augmentation [24] to enhance the dataset with slight distortion. This data enhancement enables the model to improve its generalization. To prepare the augmented data, we use 63, 70, 70, 57 and 63 original RGB images of Blast, Bacterial leaf blight, Brownspot, Sheath blight, and Tungro, respectively, and apply twelve types of augmentation techniques [20,24]. The statistics of the augmented dataset is shown in Table 1. To augment data with image rotations, we rotate the images by $-90°$, $90°$, $180°$ and $270°$, respectively. To augment data, we also apply flipping (up-down, left-right), shifting (horizontal and vertical) and scaling (by 0.6, 0.75 and 0.90). Finally, zooming (only in) is applied in augmenting rice leaf disease images. To implement the above data augmentation techniques we adopt OpenCV library [5]. Some samples of augmented images of different data augmentations are shown in Fig. 2.

3.3 CNN-Based Rice Leaf Disease Recognition Model

We propose a custom CNN-based model for recognizing rice leaf diseases. The model is designed with a depth of 10 layers. These are input layer, convolution layer 1 (Conv1), max pooling layer (Pooling1), convolution layer 2 (Conv2), max pooling layer 2 (Polling2), convolution layer 3 (Conv3), max pooling layer 3 (Pooling3), two dense layers (Dense1 and Dense2) and an output (softmax) layer as shown in Fig. 3 for an input image of size w × h.

Input Layer. The input layer of our model is fed by an RGB image of size $w_0 \times h_0$, where w_0 is the width and h_0 is the height of the image, respectively.

Convolution Layer(s). A convolution layer's primary task is to identify local conjunctions of features from the previous layer and map their presence to a feature map. In our model, we use three convolution layers, including several filters to get the output feature maps. Thus, these maps save the information where the feature takes place in the image and how well it assembles to the filter. Therefore, each filter is trained spatial regarding the position in the volume it is applied to, and each filter detects certain features from the rice leaf disease image. In this layer, the following equation computes the output feature maps.

Table 2. Related parameters of our CNN-based model for recognizing rice leaf diseases

Layers	Function	Filter/Pool	#Filters	Output	#parameters
Input	–	–	–	256×256	0
Conv1	Convolution	3×3	16	$16 \times 254 \times 254$	448
Pooling1	Max pooling	2×2	–	$16 \times 127 \times 127$	0
Conv2	Convolution	3×3	32	$32 \times 125 \times 125$	4640
Pooling2	Max pooling	2×2	–	$32 \times 62 \times 62$	0
Conv 3	Convolution	3×3	64	$64 \times 60 \times 60$	18496
Pooling3	Max pooling	2×2	–	$64 \times 30 \times 30$	0
Dense1	–	–	–	$1{\times}1 \times 64$	3686464
Dense2	–	–	–	$1 \times 1 \times 5$	325
Output	Softmax	–	–	$1 \times 1 \times 5$	0

$$Z_i^l = b_i^l + \sum_{j=1}^{n^{l-1}} K_{i,j}^l * Z_j^{l-1} \tag{1}$$

where b_i^l is a bias, l represents the layer, $K_{i,j}^l$ represents the filter, n^{l-1} represents the number of filters, Z_j^{l-1} and Z_i^l denotes the input and output feature maps, respectively. We use nonlinear activation function ReLU to allow complex relationships in the data to be learned easily. In our model, we use three convolution layers, namely Conv1, Conv2 and Conv3. We use 16, 32 and 64 filters for Conv1, Conv2 and Conv3, respectively. Table 2 illustrates the model parameters of these layers for an RGB image of size 256 × 256 and 3 × 3 filter.

Pooling Layer(s). In our model, pooling plays a vital role by reducing variance and computation complexity, resulting in fewer parameters to learn. It performs a down-sampling operation along with the spatial dimensions and reduces the dimensions of the feature map. Furthermore, it summarizes the feature that appears in a portion of the feature map generated by the convolution layer. Therefore, the rest of the operations are performed on summarized features that make the model more robust to variations in the location of the rice leaf disease images' features. In our model, we use 3 pooling layers, namely Pooling1, Pooling2 and Pooling3. Table 2 illustrates the model parameters of these pooling layers for an RGB image of size 256 × 256 and 2 × 2 pool.

Dense Layer(s). The output of the final max Pooling layer is flattened into a one-dimensional vector to fed into a fully connected dense layer. This layer produces a one-dimensional vector M of size 64 which is fed into second fully connected dense layer to produce a one-dimensional vector M' of size 5.

Output (Softmax) Layer. The output layer applies the softmax activation function which exponentially normalizes the dense layer(s) output M' and produces a distribution of probabilities across the five different rice leaf disease classes. The softmax function in our model relies upon the following formula.

$$\sigma(M')_i = \frac{e^{M'_i}}{\sum_{j=1}^{5} e^{M'_j}} \text{ for } i = 1, 2, ..., 5 \text{ and } M' = (M'_1, ..., M'_5)\varepsilon\mathbb{R}^5 \quad (2)$$

3.4 Training the Model and Classification of Rice Leaf Diseases

In our CNN-based rice leaf disease recognition model, deep features of rice leaf diseases are extracted by our custom CNN-based model. Activations in each layer of our CNN-based model transform the detailed information in the input rice leaf disease image into a more abstract representation as the image passes through the deeper layers of the model and summarize the important features of it. The visual representations of a sample rice leaf image in each of the convolution and pooling layers of our model are illustrated Fig. 4. Deeper and more accurate (summarised) information is then used as feature and classified using softmax layer of our model.

To train our model, we pass the images in batches to learn and optimize the network parameters in the convolution, pooling and dense layers to summarize the features into a 1×64 vector. These features are then passed into another dense layer to produce a 1×5 vector. This vector is finally then passed into the softmax layer to classify a rice leaf disease image into its corresponding class. We pass the training images a number of times called epochs and validate the model and the corresponding parameters through the set of validation images. We use "Categorical Cross-Entropy" as the loss function for our model.

Our model can also be extended to adapt into the binary classification task by having two class of training images and restricting the output of the softmax layer into two labels.

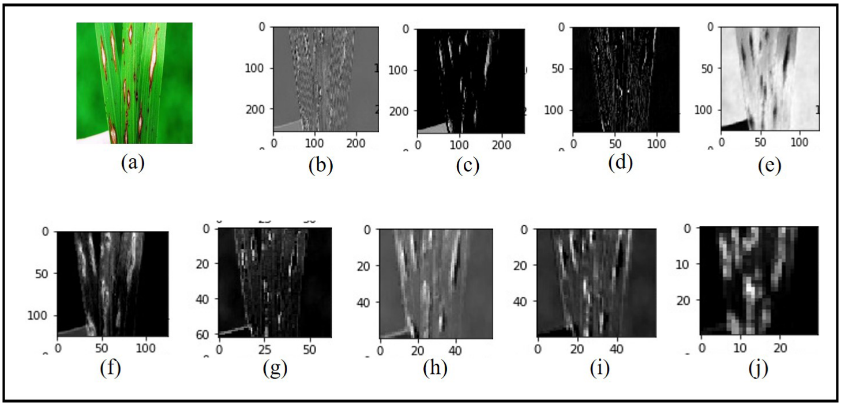


Fig. 4. Visual representations of a rice leaf disease image in each layer of our model: (a) input RGB rice blast leaf disease image; (b) a feature map of Conv1; (c) a feature map of ReLU activation1; (d) a feature map of Pooling1; (e) a feature map of Conv2; (f) a feature map of ReLU activation2; (g) a feature map of Pooling2; (h) a feature map of Conv3; (i) a feature map of ReLU activation3; and (j) a feature map of Pooling3

4 Experiments and Results

This section presents our experiments and result analysis of our model.

Environment. The experiment is conducted on an AMD Ryzen 7 2700X Eight-core 3.7GHz Processor with 32 GB RAM and NVIDIA GEFORCE RTX 2060 SUPER 8GB GPU memory. The proposed model is implemented in Python with packages Keras and TensorFlow under OS Ubuntu 18.04.

Training, Validation and Test Datasets. We analyze 4199 RGB colored 256×256 sized images of five common rice leaf diseases, including Blast, Bacterial leaf blight, Brownspot, Sheath blight, and Tungro. We use 50% data for training, 30% data for validation and the rest 20% for the test similar to [19] as shown in Table 3. Validation data is used to get optimal hyper-parameters for preventing the overfitting of the model.

Table 3. Training, validation and test datasets

Class	#Training Images	#Validation Images	#Test Images
Blast	410	246	163
Bacterial Leaf Blight	455	272	183
Brownspot	456	272	182
Sheath Blight	371	221	149
Tungro	410	246	163
Total	**2102**	**1257**	**840**

Hyper-Parameters of Our CNN-Based Model. We use categorical cross-entropy as the loss function while training our model. We run our model for a maximum of 50 epochs as there are no further improvements in training and validation accuracies observed. Optimizer Adam is applied to optimize the loss function. Table 4 shows the best tuned hyper-parameters used in our model. In our experiments, by default we use 16, 32 and 64 3×3 filters in Conv1, Conv2 and Conv3 layers, respectively, and 2×2 max pooling in the pooling layers of our model. The default number of epochs and batch size in our experiment are 50 and 32, respectively.

Table 4. Hyper-parameters used in our CNN-based model for rice leaf disease recognition

Hyper-parameters	Value(s)
Loss function	Categorical cross-entropy
Epochs	30, 50
Batch size	32, 64
Optimizer	Adam
Learning rate	0.001
β_1	0.9
β_2	0.999

Effect of Epochs. Various number of epochs up to 50 are used to train the proposed CNN-based model under default settings. The best tuned epoch is 50, as there are no further improvements in training and validation accuracies found for our model. In Fig. 5, it is shown that the validation loss is almost the same as the training loss in $7th$ to $8th$ epochs. But in Fig. 6, at that time, the validation accuracy is greater than the training accuracy. In addition training and validation accuracies are decreasing. At $22nd$ to $23rd$ epochs and $25th$ to $26th$, validation loss is close to the training loss, but in the first case both metrics are still increasing and the later case, both of the metrics are decreasing. During $47th$ to $48th$ epochs, validation loss is closer to the training loss and the validation accuracy is at its peak. At this point, training accuracy is still increasing. From these observations, we stop training our model at the $50th$ epoch.

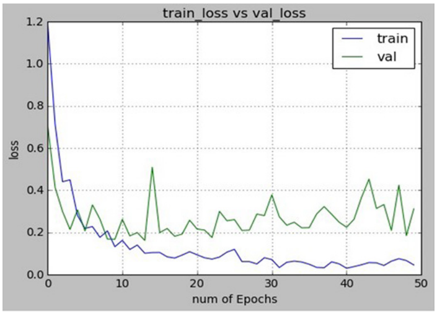


Fig. 5. Training loss vs validation loss in recognizing rice leaf diseases

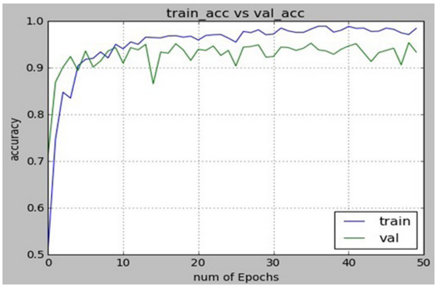


Fig. 6. Training accuracy vs validation accuracy in recognizing rice leaf diseases

Table 5. Effect of filters, poolings, epochs and batches on CNN-based rice leaf disease recognition model (dense layers dropout rate = 5%)

Convolution Layers				Pooling Layers		Epochs	#Batches	Accuracy (%)		
Filters	#Filters			Pooling	Size					
	Conv1	Conv2	Conv3					Training	Validation	Test
3 × 3	**16**	**32**	**64**	Avg	**2 × 2**	30	64.00	95.00	94.00	93
				Max		**50**	**32**	**99.78**	**97.35**	**97.82**
	32	64	128	Avg	2 × 2	50	32	96.00	94.00	93.45
5 × 5	32	64	128	Max	4 × 4	30	32	97.65	96.22	96.00
	64	128	256				64	98.45	97.56	97.12

Effect of Batches. Batch size has a strong effect on the learning of a model. We use the popular batch sizes such as 32 and 64 in our CNN-based model. The effect of these batch sizes on our model under various settings are shown in Table 5. From Table 5, we observe that our model performs the best under the default settings when the batch size is set to 32.

Effect of Filters. In our CNN-based rice leaf disease recognition model, there are three convolution layers: Conv1, Conv2 and Conv3. To test the effectiveness of different filters and number of filters in the convolution layers, we use 16, 32 and 64 in Conv1; 32, 64 and 128 filters in Conv2; and 64, 128 and 256 filters in Conv3 layers with 3 × 3 and 5 × 5 filters. Our model achieves the best training, validation and test accuracies for 16, 32 and 64 filters in Conv1, Conv2 and Conv3 layers, respectively, with 3 × 3 filter as shown in bold in Table 5.

Table 6. Effect of activation functions on our CNN-based rice leaf disease recognition model under default settings

Activation Function	tanh	ReLU	Sigmoid
Training Accuracy	78.75%	**99.78%**	68.78%
Validation Accuracy	73.45%	**97.35%**	66.32%
Test Accuracy	72.00%	**97.82%**	66.00%

Effect of Pooling. We use max and average (avg) poolings to evaluate the performances of our CNN-based model. In all cases, there are three pooling layers followed by each convolution layer: Pooling1, Pooling2 and Pooling3 with the pooling sizes of 2 × 2 and 4 × 4. From the experiments, the best result has been found using max pooling with size of 2 × 2 as shown in bold in Table 5.

Effect of Activation Functions. Various activation functions such as ReLU, tanh and sigmoid are experimented in each of the convolution layer of our CNN-based model for rice leaf disease recognition. From our experiments, the best training, validation and test accuracies of our model under default settings are found using the ReLU activation function as it is evident from Table 6.

Table 7. Effect of dropout rates on our CNN-based rice leaf disease recognition model under default settings

Dropout	0.01	0.02	0.03	0.04	0.05	0.06	0.07
Training Accuracy	98.35%	99.25%	99.21%	99.25%	**99.78%**	**99.82%**	99.50%
Validation Accuracy	96%	97.43%	97.43%	97.21%	**97.35%**	**97.35%**	97.21%
Test Accuracy	95.78%	97.172%	97.172%	97.65%	**97.82%**	**97.82%**	97.65%

Effect of Dropout Rates. Different dropout rates 1%∼7% are experimented in the dense layer(s) of our CNN-based model for rice leaf disease recognition. From our experiments, we find that our model showcases higher training accuracies in comparison to validation accuracies for dropout rates less than 5% and dropout rates greater than 6% (data overfitting). We find stable performance of our model in terms of training and validation accuracies for dropout rates 5%∼6% as it is evident from Table 7.

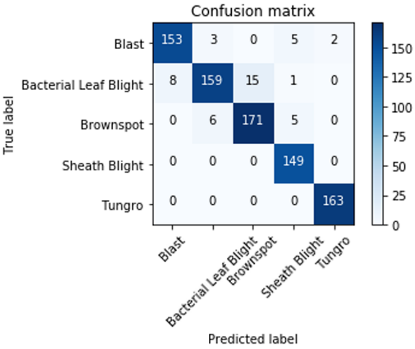


Fig. 7. Confusion matrix for recognizing five different rice leaf diseases

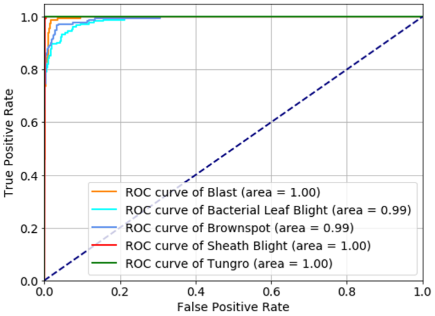


Fig. 8. ROC curve of each rice leaf disease recognition of our model

Performance Analysis. An independent dataset of 840 rice leaf disease images of five classes is used to evaluate our CNN-based model. The classification and misclassifications of our CNN-based model for each rice leaf disease is represented in a confusion matrix as shown in Fig. 7. For assessing and comparing classifier performance over its entire operating range, a receiver operating characteristic (ROC) curve for rice leaf disease recognition is illustrated in Fig. 8. The area under the ROC curve (AUC) is 0.99 for our proposed CNN-based rice leaf disease recognition model. In our model, the AUC for the classes Bacterial leaf blight and Brownspot is 0.99, while the highest value of AUC 1.00 is found for Blast, Sheath Blight and Tungro.

To further evaluate the performance of our model, we also consider the metrics such as accuracy, precision, recall and F1 score of each rice leaf disease class and compare the performance of our model with the benchmark model proposed by Liang et al. in [16], Lu et al. [17] and Rahman et al. [22]. These metrics are summarised in Table 8. From Table 8, it is evident that our model achieves better results in most of the cases than the model in [16]. Our CNN-based model achieves accuracy, precision, recall and F1 score 97.82%, 94.8%, 95% and 94.6%, respectively, on average, which is superior to the average performance achieved by the state-of-the-art models.

Binary Classification. To compare the effectiveness of recognizing rice leaf diseases among our CNN-based model, Liang et al. in [16], Lu et al. [17] and Rahman et al. [22], we implement the binary classification for each class of rice leaf disease using the proposed best tuned CNN-based model as shown in the bold line of Table 5. The experiments are Blast vs Non-Blast, Brownspot vs Non-Brownspot, Bacterial leaf blight vs Non-Bacterial leaf blight, Sheath blight vs Non-Sheath blight and Tungro vs Non-Tungro. Experiments are conducted using 628 training images, 268 validation images and 210 test images for each class. In each experiment, test images are equally distributed as positive and negative classes. Figure 9 represents the confusion matrices of binary classifications using our CNN-based model. The improvements in recognition rates (#true

Table 8. Performance evaluation of our CNN-based model for each rice leaf disease recognition on independent dataset and the state-of-the-art models

Class	Our Model				Model in [16]			
	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
Blast	**97.85%**	95%	**94%**	**94%**	97.60%	90%	93%	92%
Bacterial Leaf Blight	96.00%	**95%**	87%	91%	95.83%	92%	83%	87%
Brownspot	**96.90%**	92%	94%	**93%**	95.83%	91%	91%	91%
Sheath Blight	98.60%	93%	**100%**	96%	97.83%	97%	100%	**98%**
Tungro	**99.76%**	**99%**	**100%**	**99%**	99.64%	95%	100%	97%
Average	**97.82%**	**94.8%**	**95.0%**	**94.6%**	**97.35%**	**93.0%**	**93.4%**	**93.0%**
Class	Model in [17]				Model in [22]			
	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
Blast	91.90%	98%	92%	92%	94.40%	**99%**	72%	83%
Bacterial Leaf Blight	96.43%	77%	84%	80%	**96.79%**	88%	**99%**	**93%**
Brownspot	92.86%	**94%**	72%	81%	96.31%	87%	**98%**	92%
Sheath Blight	**99.40%**	**99%**	97%	**98%**	98.69%	93%	100%	96%
Tungro	99.17%	96%	100%	98%	98.33%	99%	92%	96%
Average	**95.95%**	**92.8%**	**89.0%**	**89.8%**	**96.90%**	**93.2%**	**92.2%**	**92.0%**

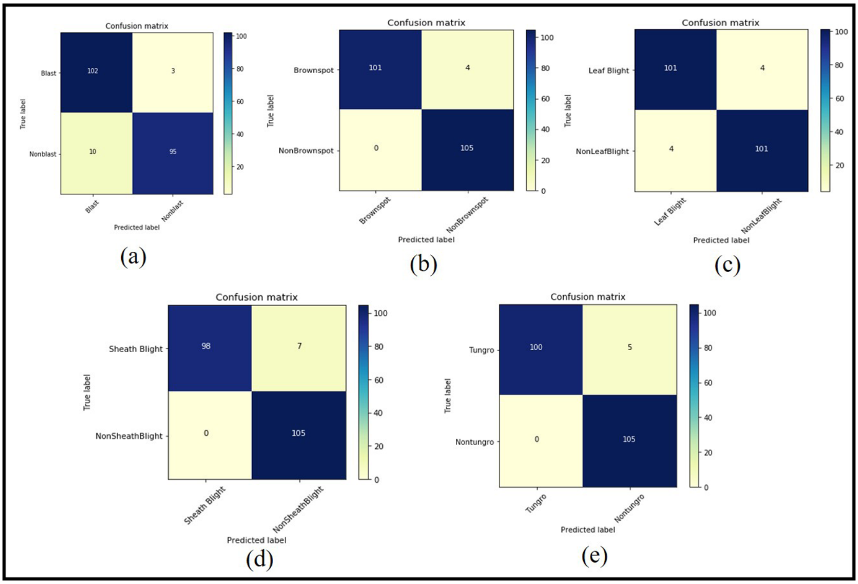


Fig. 9. Confusion matrices of binary classifications using our CNN model: (a) Blast vs Non-Blast; (b) Brownspot vs Non-Brownspot; (c) Blight vs Non-Blight; (d) Sheath blight vs Non-Sheath blight; and (e) Tungro vs Non-Tungro

positives/total # samples of the class) achieved by our model in comparison to the state-of-the-art models are shown in Table 9. From Table 9, it is evident that our model achieves better results in most of the cases than the state-of-the-art models.

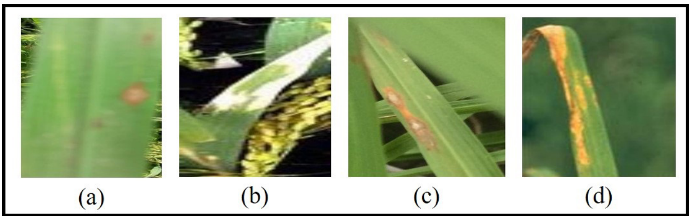


Fig. 10. Example images of misclassification: (a) Blast misclassified as Brownspot; (b) Bacterial Leaf Blight misclassified as Blast; (c) Blast misclassified as Bacterial Leaf Blight; and (d) Bacterial Leaf Blight misclassified as Sheath Blight

Table 9. Comparison of binary classification performance between our model and the state-of-the-art models

Binary Classification	Recognition Rates				Improvement		
	Our Model	Model [16]	Model [17]	Model [22]	Model [16]	Model [17]	Model [22]
Blast	97%	95%	91%	97%	2.11%	6.59%	0.00%
Non-Blast	90%	88%	88%	95%	2.27%	2.27%	−5.26%
Brownspot	96%	85%	88%	92%	12.94%	9.09%	4.35%
Non-Brownspot	100%	96%	92%	93%	4.16%	8.70%	7.5%
Blight	96%	90%	91%	95%	6.67%	5.49%	1.05%
Non-Blight	96%	95%	82%	93%	1.05%	17.07%	3.23%
Seathblight	93%	95%	92%	99%	−2.10%	1.09%	−6.06%
Non-Seathblight	100%	100%	92%	100%	0.00%	8.70%	0.00%
Tungro	95%	90%	96%	98%	5.56%	−1.04%	−3.06%
Non-Tungro	100%	100%	100%	99%	0.00%	0.00%	1.01%
Average	**96.3%**	**93.4%**	**91.2%**	**96.1%**	**3.27%**	**5.80%**	**2.76%**

Critical Evaluation. After analyzing the reasons of misclassifications, we find that the image of Fig. 10(a) is misclassified as Brownspot due to the blur and some incomplete lesions. The reason behind the misclassification of the image in Fig. 10(b) is the surrounding complex background and the image in Fig. 10(c) is the similar symptoms among the diseases. Finally, the image in Fig. 10(d) is misclassified as Sheath Blight due to the changes of appearance caused by the illumination. These are the challenges that need to be addressed in the future. Table 10 shows the number of network parameters of different rice leaf disease recognition models. As farmers work in the field, it is understandable that they would need a handheld device. So, it is important to make a trade-off between memory restriction and accuracy for choosing the rice leaf disease recognition models for the farmers. Though the number of parameters of our model is larger than the parameters found in Lu et al. [17] and Rahman et al. [22], still our model is effective for memory restricted devices in comparison of the state-of-the-art CNN models such as VGG16, InceptionV3 and NasNet based models explored in [22].

Table 10. Number of network parameters in different models

Model	#Network Parameters
Our model	3,710,373
Model proposed by Liang et al. [16]	5,668,857
Model proposed by Lu et al. [17]	4,69,317
Model proposed by Rahman et al. [22]	7,33,138

5 Conclusion and Future Work

In this paper, we have proposed a custom CNN-based model that can classify five common rice leaf diseases commonly found in Bangladesh. Our model is trained to recognize the rice leaf diseases in different image backgrounds and capture conditions. Our model achieves 97.82% accuracy on independent test images. Moreover, our model is effective with respect to memory storage due to its reduced number of network parameters. Despite having better accuracy, we aim to improve the reliability and robustness of our model on different datasets from other regions. We will work on classifying rice leaf disease images when complex backgrounds are present and have varied illumination condition. Also, as classification accuracy is an incomplete description of most real-world tasks [4,8], we will concentrate on interpretable CNN-based models to present features in understandable terms for which diseases will be classified.

References

1. Importance of Rice (2020). http://www.knowledgebank.irri.org/ericeproduction/Importance_of_Rice.htm. Accessed 20 June 2020
2. Bhattacharya, S., Mukherjee, A., Phadikar, S.: A deep learning approach for the classification of rice leaf diseases. In: Bhattacharyya, S., Mitra, S., Dutta, P. (eds.) Intelligence Enabled Research. AISC, vol. 1109, pp. 61–69. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-2021-1_8
3. Boulent, J., Foucher, S., Théau, J., St-Charles, P.L.: Convolutional neural networks for the automatic identification of plant diseases. Front. Plant Sci. **10** (2019)
4. Brahimi, M., Mahmoudi, S., Boukhalfa, K., Moussaoui, A.: Deep interpretable architecture for plant diseases classification. In: Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), pp. 111–116 (2019)
5. Brahmbhatt, S.: Practical OpenCV. Apress, New York (2013)
6. Das, A., Mallick, C., Dutta, S.: Deep learning-based automated feature engineering for rice leaf disease prediction. In: Das, A.K., Nayak, J., Naik, B., Dutta, S., Pelusi, D. (eds.) Computational Intelligence in Pattern Recognition. AISC, vol. 1120, pp. 133–141. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-2449-3_11
7. Ding, W., Wang, R., Mao, F., Taylor, G.: Theano-based large-scale visual recognition with multiple GPUs. arXiv preprint arXiv:1412.2302 (2014)
8. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608 (2017)

9. Ferentinos, K.P.: Deep learning models for plant disease detection and diagnosis. Comput. Electron. Agric. **145**, 311–318 (2018)
10. Hanson, A.M.J., Joy, A., Francis, J.: Plant leaf disease detection using deep learning and convolutional neural network, vol. 7 (2017)
11. Jeon, W.S., Rhee, S.Y.: Plant leaf recognition using a convolution neural network. Int. J. Fuzzy Logic Intell. Syst. **17**(1), 26–34 (2017)
12. Kaur, R., Kaur, V.: A deterministic approach for disease prediction in plants using deep learning, vol. 7, February 2018
13. Khan, M.A., et al.: CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep CNN features. Comput. Electron. Agric. **155**, 220–236 (2018)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 (2012)
15. LeCun, Y., et al.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (1989)
16. Liang, W.J., Zhang, H., Zhang, G.F., Cao, H.X.: Rice blast disease recognition using a deep convolutional neural network. Sci. Rep. **9**(1), 1–10 (2019)
17. Lu, Y., Yi, S., Zeng, N., Liu, Y., Zhang, Y.: Identification of rice diseases using deep convolutional neural networks. Neurocomputing **267**, 378–384 (2017)
18. Miah, S., Shahjahan, A., Hossain, M., Sharma, N.: A survey of rice diseases in Bangladesh. Int. J. Pest Manag. **31**(3), 208–213 (1985)
19. Mohanty, S.P., Hughes, D.P., Salathé, M.: Using deep learning for image-based plant disease detection. Front. Plant Sci. **7**, 1419 (2016)
20. Pai, P.: Data Augmentation Techniques in CNN using Tensorflow (2017)
21. Patidar, S., Pandey, A., Shirish, B.A., Sriram, A.: Rice plant disease detection and classification using deep residual learning. In: Bhattacharjee, A., Borgohain, S.K., Soni, B., Verma, G., Gao, X.-Z. (eds.) MIND 2020. CCIS, vol. 1240, pp. 278–293. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-6315-7_23
22. Rahman, C.R., et al.: Identification and recognition of rice diseases and pests using convolutional neural networks. Biosyst. Eng. **194**, 112–120 (2020)
23. Saleem, M.H., Potgieter, J., Arif, K.M.: Plant disease detection and classification by deep learning. Plants **8**(11), 468 (2019)
24. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. J. Big Data **6**(1), 60 (2019)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
26. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D.: Deep neural networks based recognition of plant diseases by leaf image classification. Comput. Intell. Neurosci. **2016** (2016)
27. Szegedy, C., et al.: Going deeper with convolutions. In: IEEE CVPR, pp. 1–9 (2015)
28. Tm, P., Pranathi, A., SaiAshritha, K., Chittaragi, N.B., Koolagudi, S.G.: Tomato leaf disease detection using convolutional neural networks. In: International Conference on Contemporary Computing (IC3), pp. 1–5. IEEE (2018)
29. Toda, Y., et al.: How convolutional neural networks diagnose plant disease. Plant Phenomics **2019**, 9237136 (2019)
30. Too, E.C., Yujian, L., Njuki, S., Yingchun, L.: A comparative study of fine-tuning deep learning models for plant disease identification. Comput. Electron. Agric. **161**, 272–279 (2019)

STCNet: Spatial-Temporal Convolution Network for Traffic Speed Prediction

Mingjun Ma[1], Bo Peng[2], Ding Xiao[1](✉), Yugang Ji[1], and Chuan Shi[1]

[1] Beijing University of Posts and Telecommunications, Beijing, China
{mamingjun,dxiao,jiyugang,shichuan}@bupt.edu.cn
[2] State Grid Jibei Information and Telecommunication Company, Beijing, China
25811247@qq.com

Abstract. With cars being a necessity of life, traffic congestion has been more and more serious in recent years. To solve such problems, it is promising to predict the speed of sensors in traffic networks, namely traffic speed prediction. While there are abundant spatial and temporal dependencies within in traffic networks, existing researches are unable to capture the structural information and dynamic evolution of sensors at the same time. In this paper, we propose a **S**patial-**T**emporal **C**onvolution **Net**work (**STCNet**), which mainly consists of a temporal block and a spatial block. We first design the temporal traffic network to model the temporal information in the topological graph. And then, we design the temporal block to model the short-term and long-term dependencies via different receptive fields. We further present the spatial block employs the convolution operation on graph to capture the spatial dependencies among nodes. Finally, we integrate both temporal and spatial representations of sensors into a unified framework for optimization. Extensive experiments on traffic speed dataset demonstrate that our proposed STCNet model outperforms the state-of-the-art baselines.

Keywords: Spatial dependencies · Temporal dependencies · Convolution network

1 Introduction

Traffic speed prediction, which is to predict the average speed of observed nodes at several future time steps on the traffic network, has become a long-standing and critically important topic in the area of ITS [14]. The abundant time series data with its corresponding geographic information becomes the foundation of traffic speed prediction.

To capture both spatial and temporal information for traffic speed prediction, it is naïve to extract both spatial and temporal features. In spatial dimension, many researches [8] use Convolutional Neural Network (CNN) to extract the spatial dependencies, which regards the traffic road as a kind of grid-based

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 315–323, 2020.
https://doi.org/10.1007/978-3-030-65390-3_24

data. However, these CNN-based methods have limitations for the traffic network in the real daily life is graph-based data. In temporal dimension, current methods [10] mainly employ RNN and its variants to handle the historical traffic data. However, these methods usually suffer limitations when applied to extract temporal dependencies. While there are both the short-term and long-term dependencies in the real world traffic network, these RNN-based methods cannot change the receptive fields to both extract the short-term and long-term correlations. Moreover, such models often cause gradient explosion or vanishing as well. In this paper, we focus on exploring the non-linear and complicated spatial-temporal correlations for traffic speed prediction.

As mentioned above, there are two main challenges in this paper: ***1) How to model both short-term and long-term temporal dependencies of sensors?*** On the one hand, the speed of sensors often changes at different temporal periods, leading to morning/evening peaks. On the other hand, the changes of traffic speed are often periodic such as weeks, months, quarters and so on. ***2) How to effectively integrate both temporal dependencies and spatial dependencies at the same time to describe the situations of sensors?*** Traditional CNN-based models fail to model the topological information, while such graph-data often contains abundant spatial dependencies of sensors.

To tackle the above challenges, we propose a novel deep learning model: **S**patial-**T**emporal **C**onvolution **Net**work (STCNet) to predict traffic speed based on the historical data and the traffic network. To model both short-term and long-term dependencies, we propose a multi-receptive fields temporal block to convolute historical and current information of sensors. To make full use of spatial information, we design the global aware spatial block via multi-layer neighborhood aggregation so as to extract spatial dependencies. Then we evaluate the performance on the real-world public dataset PEMS-BAY.

2 Related Work

Traffic prediction has been widely studied in recent years. With the rapid development of deep learning, some researches propose to detect the non-linear latent information for traffic prediction by inputting features into neural networks. The Stacked Auto Encoder (SAE) [7] is the first to predict the traffic condition on different nodes. In the perspective of spatial dimension, [12] and [13] both treat the city as an image and the traffic volume is the pixel value of each grid during a certain time. By using a set of historical traffic images (i.e., snapshots), these models are able to predict the future traffic images. Convolutional Neural Networks (CNNs) [7] have been adopted to capture the spatial correlations. [5] proposes to apply both CNNs and the residual connections to predict the traffic patterns. In order to modeling the evolution of sensors, some works [4,11] propose to input the encoded spatial features of sensors into recurrent neural networks, such as Long Short Term Memory network (LSTM) and Gated Recurrent Unit (GRU) [3]. In [7,15] the authors utilize LSTM to deal with the temporal evolution for traffic speed prediction. As mentioned above, for spatial modeling, directly transforming the traffic network into grid data will break the nature of the original graph data, while for temporal modeling, the existing methods can't both measure the short-term and long-term dependencies.

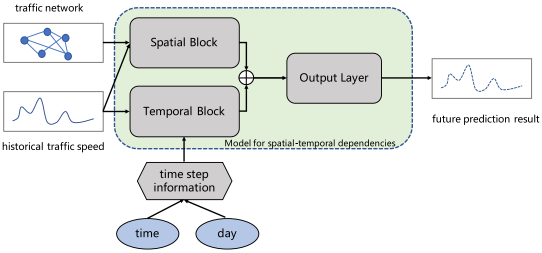


Fig. 1. The overview of the STCNet.

3 Proposed Model

In this section, we propose our **S**patial-**T**emporal **C**onvolution **Net**work (**STCNet** in short) framework. We outline the overall framework of STCNet in Fig. 1. In this framework, we respectively design the multi-receptive fields temporal block and the global aware spatial block to fully use the temporal dependencies and spatial dependencies. We will introduce each component in the following.

3.1 Multi-receptive Fields Temporal Block

Because the traffic speed can not only be impacted by the neighboring time steps but can also be influenced by the time steps a few moments ago. To overcome the challenge of dynamics, we propose the multi-receptive fields temporal block to model the evolution of traffic speed of sensors.

As illustrated in Fig. 2, the input of the temporal block can be divided into two parts, one is the traffic speed value and the other is the time position information of the current time step. The current time step can be recorded by the time of a day and the day of a week. The multi-receptive fields temporal block is composed of several convolution operations along the time axis with different receptive fields. Because the traffic speed data is time series data, the outputs at current time are only related on the historical data. The result of the convolution [1] at node v_i is represented as follow:

$$y_{i,t} = \sum_{k=1}^{K} w_k \cdot x_{i,t-d(k-1)}, \tag{1}$$

where $y_{i,t}$ stands for the convolutional result of node v_i at time t, d is the dilation rate to control the length of the receptive field and w_k is the convolution kernel.

In order to expand the receptive field, the dilation rate increase at an exponential speed and calculated as $d^{(l)} = 2^{(l-1)}$, where $d^{(l)}$ is the dilation rate at the l^{th} layer of the network. When applying the convolution operation to all the nodes in the graph, the output of the l^{th} layer is:

$$y^l = \begin{cases} \mathcal{X}, & \text{if } l = 0 \\ \sigma(\mathbf{W}^l *_{d^{(l)}} y^{l-1}), & \text{if } l = 1, 2, \cdots L \end{cases} \tag{2}$$

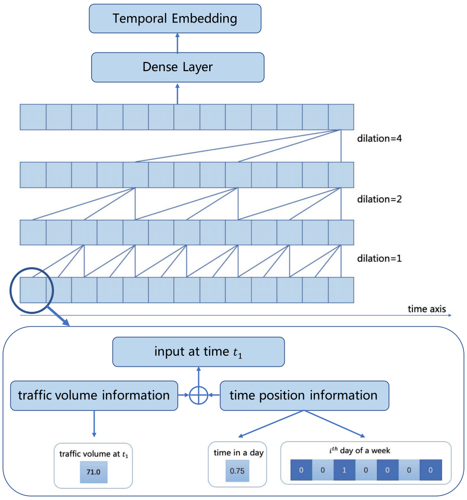


Fig. 2. The details of the temporal block

where $y^l \in \mathbb{R}^{N \times Q}$ means the outputs of N nodes of the l^{th} layer, $\mathbf{W}^l$ is the convolutional kernel, $\sigma(\cdot)$ is the non-linear activation function and $*d^{(l)}$ means the convolution with the dilation rate $d^{(l)}$. Finally, through a dense layer, we can adjust the output dimension and get the temporal embedding T_{emb}.

3.2 Global Aware Spatial Block

Aiming at fully modeling the spatial information among sensors in the temporal traffic network, the global aware spatial block is developed to extract the latent structural dependencies. Usually, the spatial dependencies are different at different time steps. In order to consider this rule and make full use of the topological properties, we adopt graph convolution operations to process the traffic speed at each time slice via neighborhood aggregation.

From the perspective of spectral graph analysis, a graph is represented by its Laplacian matrix. The Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacent matrix and $\mathbf{D} \in \mathbb{R}^{N \times N}$ stands for the diagonal degree matrix, consisting the node degrees with $D_{ii} = \sum_j \mathbf{A}_{ij}$. The normalized form of Laplacian matrix is $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{I}_N$ is a unit matrix with N dimension. $\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ is the eigenvalue decomposition of Laplacian matrix, where $\boldsymbol{\Lambda} \in \mathbb{R}^{N \times N}$ is the diagonal matrix with corresponding eigenvalues and $\mathbf{U} \in \mathbb{R}^{N \times N}$ is Fourier basis. Back to our traffic speed forecasting scene, the traffic speed at time t is x and the graph Fourier transform is $\hat{x} = \mathbf{U}^T x$. Taking the Laplacian matrix properties into account, the inverse Fourier transform is $x = \mathbf{U}\hat{x}$. Based on all the inferences mentioned before, we can calculate the traffic speed x on the graph G filter by a kernel k_θ as $k_\theta *_G x = k_\theta(\mathbf{L})x = k_\theta(\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T)x$, where $*_G$ is the convolution operation on the graph G. However, this kind of graph convolution needs to directly perform the eigenvalue decomposition, which is expensive when the scale of the traffic network is large. Therefore, the convolution kernel is replaced with Chebyshev polynomial [2] as $k_\theta *_G x = k_\theta(\mathbf{L})x = \sum_{k=0}^{K-1} \theta_k T_k(\widetilde{\mathbf{L}})x$, where θ_k is a learnable polynomial parameter and $\widetilde{\mathbf{L}} = 2\mathbf{L}/\lambda_{max} - \mathbf{I}_N$. λ_{max} is the maximum among all the eigenvalues. $T_k(\widetilde{\mathbf{L}})$ is the Chebyshev polynomial with the recursive definition $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, where $T_0(x) = 1$ for $k = 0$ and $T_1(x) = x$ for $k = 1$. The activation function we used is *ReLU*. After the convolution operation on the whole graph, we can get the spatial embedding S_{emb}.

3.3 The Unified Framework

In order to extract both the temporal dependencies and spatial dependencies, we concatenate temporal embedding T_{emb} and spatial embedding S_{emb} together of node v_i at time t as a new vector ST_{emb}, followed by an output layer to predict the future traffic speed value. MSE loss function is used to train the model as: $\ell(\theta) = \|\widetilde{\mathcal{X}} - \hat{\mathcal{X}}\|^2$, where $\widetilde{\mathcal{X}}$ is the real traffic speed value and $\ell(\theta)$ is the loss function we need to optimize.

4 Experiments

In order to evaluate the performance of our proposed model, we present our experiments on the dataset—PEMS-BAY. The pre-processing steps follow the methods used in [6]. STCNet uses 12 time steps as the train time sequence to predict the future traffic speed value. The model can predict various length of the time sequence to verify the saclability of STCNet. As for the temporal block, we use *Adam* optimizer to optimize the model and the batch size in our experiments is 512. The number of the multi-receptive fields convolution layers is 3 with the dilation rate 1, 2, 4 respectively. In the spatial block, we set the number of the terms of Chebyshev polynomial as 2. Also, the optimizer is *Adam* and *ReLU* is the activation function. In the experiments, we use three common regression metrics to evaluate the performance of our proposed model, which are MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error) and RMSE (Rooted Mean Square Error).

4.1 Methods for Comparison

In this subsection, we compare our STCNet with other methods. Then, we report the performance on the given dataset. The parameter settings of the baselines follow [6].

- **HA**: Historical Average is a traditional time series method, which uses the average value of last several time steps to predict the next value.
- **ARIMA**: Auto-Regressive Integrated Moving Average combines the moving average and the autoregressive components together with Kalman filter to model the time series. The seeting is the same as [6].
- **SVR**: Support Vector Regression method, which uses linear support vector machine to predict the traffic speed value as a regression problem.
- **FNN**: Feed forward neural network, which is a deep learning method. We use this method with 2 hidden layers.
- **FC-LSTM** [9]: Recurrent Neural Network with fully connected LSTM hidden units (FC-LSTM) and it is also a deep learning model.

Table 1. Performance comparison of different approaches for traffic speed forecasting.

		HA	ARIMA	SVR	FNN	FC-LSTM	STCNet
1 h	MAE	2.88	3.38	3.28	2.46	2.37	**2.33**
	RMSE	5.59	6.50	7.08	4.98	4.96	**4.89**
	MAPE	6.80%	8.30%	8.00%	5.89%	**5.70%**	5.72%
30 min	MAE	2.88	2.33	2.48	2.30	2.20	**2.03**
	RMSE	5.59	4.76	5.18	4.63	4.55	**4.36**
	MAPE	6.80%	5.40%	5.50%	5.43%	5.20%	**5.04%**
15 min	MAE	2.88	1.62	1.85	2.20	2.05	**1.58**
	RMSE	5.59	3.30	3.59	4.42	4.19	**3.27**
	MAPE	6.80%	3.50%	3.80%	5.19%	4.80%	**3.41%**

We compare our model with five baseline methods on the real-world dataset. Table 1 illustrates the performance of STCNet on different evaluation metrics while comparing with five baseline methods. The results of the baselines follow the performance reported in [6]. In our experiments, we use the speed value of last 12 historical time steps to predict the traffic speed of next 1 h, 30 min and 15 min respectively.

From the whole results, we can see our proposed model performs best on three cases. The reason that our proposed model performs best is twofold. The first one is STCNet combines the spatial information and temporal information together into a unified model. Second one is that we find there exist short-term and long-term dependencies. Using multi-receptive fields makes it possible. When we fix the methods and compare along the prediction length, we can find that all the models perform better when the prediction length is shorter. Since long prediction length will make the difference between real value and prediction value accumulate, which leads to lower value on the evaluation metrics.

Scalability Analysis. Scalability is a significant property of models, which shows the ability of the model to apply in several situations. In order to verify the scalability of the STCNet, we compare the performance on different prediction length. The results are illustrated in Fig. 3. In the figure, the MAPE values are shown by multiplying 100 to make the figure clearer. If the prediction length is shorter, then the performance is better. The differences among different length is not large, which proves that the model has a strong scalability. This find can help the model to adjust the prediction length when the application need is different.

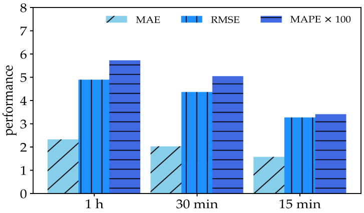


Fig. 3. Scalability analysis.

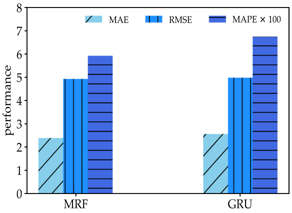


Fig. 4. Evolution modeling comparison.

Evolution Modeling Comparison. To verify the effectiveness of the multi-receptive fields temporal block, we conduct a comparative experiment. The first one is the original temporal block mentioned in our paper, and the other one is the famous block named GRU. We only use the temporal information to predict the result, aiming at find out the effectiveness of the temporal block mentioned in our paper. In Fig. 4, MRF means the multi-receptive fields temporal block and for the same reason, the MAPE values are shown by multiplying 100. We can see from the figure that the original temporal block outperforms GRU on three evaluation metrics, which verifies the importance of modeling both the short-term dependencies and long-term dependencies.

5 Conclusion

In this paper, we propose a novel method to predict the traffic speed by modeling the spatial and temporal dependencies. Spatial block is used to capture the spatial relationships among nodes and temporal block is used to model the long-term and short-term temporal relationships. Experiments on the real-world dataset show the performance of our method while forecasting the traffic speed.

In the future, we will consider more explicit information, like POI information, weather information and so on. Because these kinds of external information can influence the traffic speed at different level. Also, the traffic speed pattern is another significant feature to be considered.

Acknowledgement. This paper is funded by the State Grid Jibei Power Co. Ltd. Science and Technology Project "Research on Data Fusion and Visualization Key Technologies Based on Full-service Unified Data Center" (52018E18006N).

References

1. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. CoRR abs/1803.01271 (2018)
2. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, 5–10 December 2016, Barcelona, Spain, pp. 3837–3845 (2016)
3. Fu, R., Zhang, Z., Li, L.: Using LSTM and GRU neural network methods for traffic flow prediction. In: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), pp. 324–328. IEEE (2016)
4. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017. Proceedings of Machine Learning Research, vol. 70, pp. 1243–1252. PMLR (2017)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016, pp. 770–778. IEEE Computer Society (2016)
6. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: data-driven traffic forecasting. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, 30 April–3 May 2018, Conference Track Proceedings. OpenReview.net (2018)
7. Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.: Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. **16**(2), 865–873 (2015)
8. Ma, Y., Peng, B., Ma, M., Wang, Y., Xiao, D.: Traffic prediction on communication network based on spatial-temporal information. In: 2020 22nd International Conference on Advanced Communication Technology (ICACT), pp. 304–309. IEEE (2020)
9. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, 8–13 December 2014, Montreal, Quebec, Canada, pp. 3104–3112 (2014)
10. Williams, B.M., Hoel, L.A.: Modeling and forecasting vehicular traffic flow as a seasonal arima process: theoretical basis and empirical results. J. Transp. Eng. **129**(6), 664–672 (2003)
11. Yu, R., Li, Y., Shahabi, C., Demiryurek, U., Liu, Y.: Deep learning: a generic approach for extreme condition traffic forecasting. In: Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, Texas, USA, 27–29 April 2017, pp. 777–785. SIAM (2017)
12. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 4–9 February 2017, San Francisco, California, USA, pp. 1655–1661. AAAI Press (2017)

13. Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X.: DNN-based prediction model for spatio-temporal data. In: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2016, Burlingame, California, USA, October 31 - November 3, 2016, pp. 92:1–92:4. ACM (2016)
14. Zhang, J., Wang, F., Wang, K., Lin, W., Xu, X., Chen, C.: Data-driven intelligent transportation systems: a survey. IEEE Trans. Intell. Transp. Syst. **12**(4), 1624–1639 (2011)
15. Zhou, X., Shen, Y., Zhu, Y., Huang, L.: Predicting multi-step citywide passenger demands using attention-based neural networks. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, 5–9 February 2018, pp. 736–744. ACM (2018)

Discriminative Features Generation for Mortality Prediction in ICU

Suresh Pokharel[1](✉), Zhenkun Shi[2](✉), Guido Zuccon[1](✉), and Yu Li[1](✉)

[1] The University of Queensland, Brisbane, Australia
{s.pokharel,g.zuccon}@uq.edu.au, yuli@itee.uq.edu.au
[2] Jilin University, Changchun, China
shizk14@mails.jlu.edu.cn

Abstract. Effective methods for mortality prediction for Intensive Care Unit (ICU) patients assist health professionals by producing alerts ahead of time regarding the critical changing degeneration of a patient's health. This can guide health professionals to take immediate interventions to rescue the lives of patients. However, existing methods only use raw clinical features and ignore the compound information exhibited by Electronic Health Records (EHRs) data, i.e., the co-occurrence of different clinical events at the same point of time (or within a short time interval). In this paper we use a recently proposed method, called Temporal Tree, to capture the compound information and use them as discriminative features. In addition, to test the impact of preserving temporal information, we capture compound information in terms of patient situations (i.e., the patient's medical condition at particular point of time), and represent a patient as a sequence of patient situations. This is contrasted with the baseline approach of representing a patient by the associated sequence of clinical events (bag-of-words like). These representations are further processed to obtain low dimensional embeddings used to represent patients and their situations.

The effectiveness of the proposed methods is evaluated using a real ICU dataset with nine different baselines and state-of-the-art classifiers. The empirical results show the Temporal Tree method is able to generate discriminative patient representations. Classifiers that exploited the compounded information significantly improved the quality of ICU mortality predictions, in all cases and across both bag-of-words (up to 7.814% improvements) and patient situations representations (up to 2.720% improvements).

Keywords: Compound information · Mortality prediction · Temporal tree

1 Introduction

Intensive care units (ICUs) services are one of the largest ticket items in an hospital operational expenditure budget: according to Coopersmith et al. [7]

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 324–338, 2020.
https://doi.org/10.1007/978-3-030-65390-3_25

in fact, these costs amount to between 17.4% and 39.0% of the total hospital costs. ICUs provide highly specialised and costly treatments, such as mechanical ventilation, to acutely ill or injured patients. Among the ICU settings, the prediction of patient likelihood of mortality (i.e. whether the patient dies in ICU or is discharged alive at the end of the ICU stay), is a crucial task for assessing the critically of the patient's illness and of the possible interventions and the treatments parameter settings. However, only between 10% to 15% of ICUs in the US use mortality prediction scoring systems [4], with the major obstacle to adoption being accuracy, cost and applicability to the patient population.

Popular mortality prediction scoring methods such as Sequential Organ Failure Assessment (SOFA) [25], Simplified Acute Physiology Score 3 (SAPS 3) [19], and Acute Physiology and Chronic Health Disease Classification System III (APACHE III) [13] use clinical and laboratory variables for informing the prediction. They do so however based on simple rules or heuristics, and on a limited number of variables. For example, the SOFA score is calculated using data from only six organ systems (respiratory, cardiovascular, hepatic, coagulation, renal and neurological). But the data routinely captured in ICUs is much richer: for example the MIMIC III dataset [12] contains more than 4,000 indicators that influence a patient's conditions. It is not surprising then that research has recently been directed towards using such rich data to further improve the accuracy of ICU mortality prediction: this has taken the form of using machine learning to address this task.

Machine learning research for ICU mortality prediction has seen the application of both traditional classification methods such as Logistic Regression [15,16,26] and Support Vector Machines [9], and of recent deep learning methods such as Long Short-Term Memory (LSTM) [1,24] and Recurrent Neural Networks [5] (although this last line of work attempted to predict the illness severity at any point of time, rather then predict mortality). A common problem all these methods face is producing or learning an effective representation of the data: in either case, previous work has mainly exploited simple temporal characteristics and features of the ICU Electronic Health Records datasets such as MIMIC III (see more detail in Sect. 3.3).

However, EHRs are characterised by

1. *Complex Characteristics*: EHRs are sparse, irregular, temporal, heterogeneous and multivariate
2. *Compound Information*: along with temporal information (events occurring at subsequent points in time), EHRs contain compound information defined as co-occurrence of different clinical events at the same point of time (or within a short time interval).

Existing work ignores these types of inherent relationships between clinical events – but modelling these relationships may provide deeper representational power and insights. These relationships are due to the temporal and multivariate characteristics of EHRs. Figure 1a shows an example of the occurrence of clinical events at different time points for a *chronic renal failure* ICU patient. In the example, laboratory tests performed on 2153-07-6 at 20:11 report *high magnesium* and *high glucose*; we shall treat this as a single compound information. In EHRs, there are a large amount of these kinds of compound information which, if adequately modelled, may add insights to the EHR representation, and thus likely improve the downstream machine learning tasks, e.g., mortality prediction. Thus, it is desirable to have a way to handle the complex data characteristics as well as to capture the compound information. To achieve this, in this paper we adapt a novel temporal structural network representation called *Temporal Tree* [21].

Temporal Tree is a structural network representation which preserves sequence information related to clinical temporal events and captures compound information. Compound information is generated through a re-labelling approach and is represented in a hierarchical form. A Temporal Tree is constructed for each patient, and the tree preserves the sequence of medical situations for that patient. A medical patient situation, simply called a *patient situation*, is a medical condition observed for a patient, e.g., body temperature, blood pressure, consciousness level, at a particular point of time (or within a short time interval).

Once EHR data is represented by means of a Temporal Tree, patients and their medical situations are represented into a lower dimensional vector space through the use of embeddings. A well known document-level embedding technique [14] (doc2vec) is applied so as to learn lower vector representations. This is done by using an optimisation function tailored to patients as well as their situations. The motivations for adapting the embedding method are that (i) all patients and their situations are represented within the same dense, lower dimensional embedding space, so that processing is easier and faster; and, (ii) similar patients, and similar patients situations, have similar embeddings: this can potentially translate into improvement in the downstream classification tasks.

Patients embeddings are generated in two ways: (1) by treating all clinical events of a patient as a Bag of Word (BOW), so that the corresponding patient vector is generated. In this method, the patient's temporal information is lost. And, (2) by modelling a patient with a sequence of the patient's medical situations, so that then a patient vector is calculated from the patient situation vectors. This preserves the patient's temporal information. In summary, in this paper we provide the following contributions:

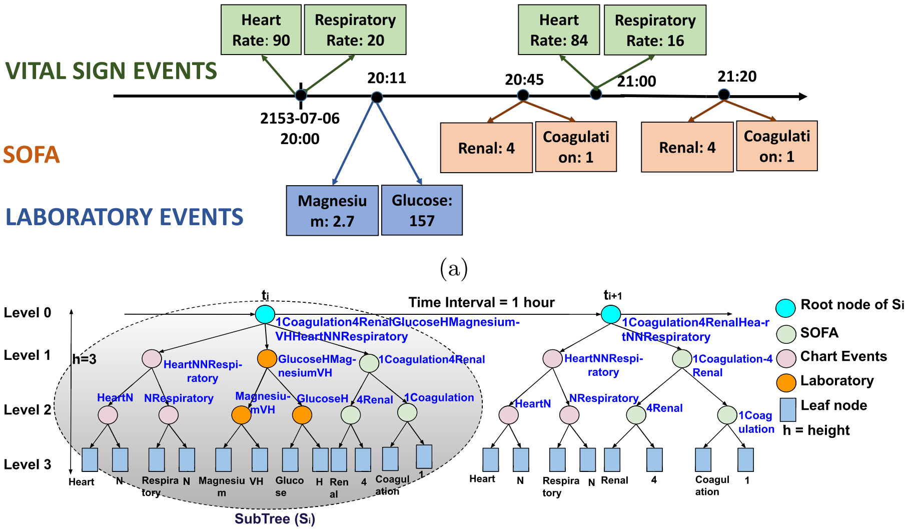
1. We adapt the Temporal Tree representation to incorporate compound EHR information, which captures the complex relationships between clinical events, thus providing a richer representation of the EHR data.
2. We adapt an embedding technique to be able to produce embeddings from Temporal Tree to represent a patient. In addition, we show that considering temporal information improves the performance of the downstream task of ICU mortality prediction.
3. We demonstrate the effectiveness of the proposed representation technique using real ICU data, across an extensive set of baselines and state-of-art methods.

2 Related Work

ICU Mortality Prediction. The mortality prediction task for ICU patients has received considerable attention. Existing methods use either structured data, e.g., data from observations such as blood pressure, temperature, laboratory test values [1,11,20,23,24], or a combination of structured and unstructured data, e.g., include clinical notes [9,15]. Our work only focuses on representing and using structured data, leaving the interpretation and representation of clinical notes to future work. Thus, next we consider some of the most representative works that have used structured data.

Suresh et al. [24] have proposed a two-step pipeline to (i) learn patient subgroups using an LSTM autoencoder, and (ii) predict patients mortality for separate subgroup of patients within a multi-task framework. Harutyunyan et al. [11] have proposed linear regression models and neural baselines (a standard LSTM, a channel-wise LSTM, and LSTMs with deep supervision and multitask training) for four prediction tasks: mortality, forecasting the length of stay, detecting physiologic decline, and phenotype classification. Nori et al. [20] have proposed a method which integrates domain knowledge, showing this is more effective than a logistic regression baseline, both with and without multitask learning. Shi et al. [23] have proposed the Deep Interpretable Mortality Model (DIMM), which employs Multi-Source Embedding, Gated Recurrent Units (GRU), Attention mechanism and Focal Loss techniques for mortality prediction. Aczon et al. [1] have proposed a deep learning architecture composed of three LSTMs. Luo et al. [16] have proposed a Subgraph Augmented Non-negative Matrix Factorization (SANMF) where time series data is represented within a graph which is then used to automatically extract the temporal trends by applying frequent subgraph mining. Trends are then grouped using matrix factorization and logistic regression is applied using features from trend groups. Lehman et al. [15] have proposed combining the learned "topic" structure from nursing notes using Hierarchical Dirichlet Processes (HDP) with physiologic data (from the Simplified Acute Physiology Score I (SAPS I)). They then use multivariate logistic regression for hospital mortality prediction. Similarly, Ghassemi et al. [9] used Support Vector Machine for mortality prediction, where they generate aggregated features which are the combination of structured information (age, sex, admitting SAPS-II score as well as derived features such as maximum/minimum SAPS-II score) and features obtained from free-text clinical notes using Latent Dirichlet Allocation. However, all these methods do not consider the compound information.

Patient Embeddings. Patient embedding techniques generate a fixed, low dimensional vectors that are used to represent patients, such that patients that are "similar" are represented by similar embeddings. Intuitively, effective patient embedding techniques may be useful because they may generate representations able to discriminate e.g. between survival and non-survival patients. A number of patient embedding techniques have been recently proposed. Patient2Vec [27] learns an interpretable deep representation for a patient



(a)

(b)

Fig. 1. (a) An example of EHR data for a patient treated in ICU. (b) Temporal Tree representation, where vital signs, sofa and laboratory are shown as examples of events.

by relying on *word2vec* [18] to embed clinical events into vectors – in a similar manner to how we rely on *doc2vec* (a *word2vec* variant) for the same goal. Bajor et al. [2] used the document-level embeddings [14] to represent patients and data elements from clinical codes and laboratory tests, thus generating a sequence of data elements from temporal data. Unlike our work, however, they do not consider the values associated to clinical events and the compound information. Choi et al. [6] proposed an embedding method for learning lower dimensional medical concept representations. Their method considers co-occurrence information along with visit sequence information present in EHRs data, while, Glicksberg et al. [10] used *word2vec* to create medical concept embeddings of the phenotype space and ranked patients based on the distance from the corresponding query embedding. Unlike this prior work, we generate temporal medical data sequences by considering clinical events along with their compound events. In addition, we also consider the patient's temporal situations while generating the patient embeddings so as to retain the temporal information.

3 Methodology

The framework we propose to represent ICU patients data and perform mortality prediction is shown in Fig. 2. The Temporal Tree is constructed for each patient and captures temporal discriminative data. Then, the embedding technique is applied to obtain a low dimensional vector representation of the patient as well as their medical situations. These embeddings are used as input to train

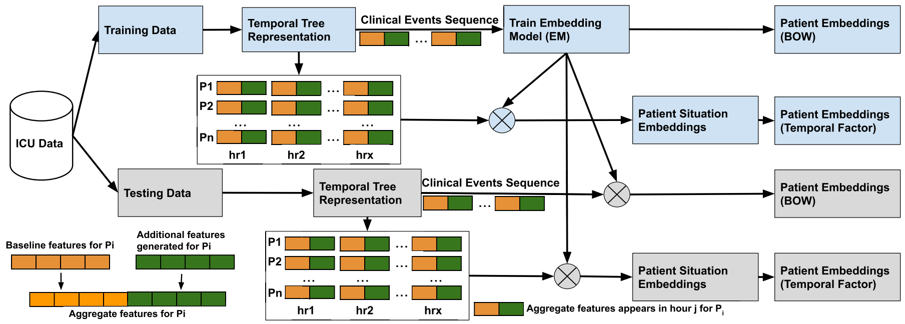

Training Data
Temporal Tree Representation
Clinical Events Sequence
Train Embedding Model (EM)
Patient Embeddings (BOW)
ICU Data
P1
P2
Pn
hr1
hr2
hrx
Patient Situation Embeddings
Patient Embeddings (Temporal Factor)
Testing Data
Temporal Tree Representation
Clinical Events Sequence
Patient Embeddings (BOW)
Baseline features for Pi
Additional features generated for Pi
Aggregate features for Pi
P1
P2
Pn
hr1
hr2
hrx
Patient Situation Embeddings
Patient Embeddings (Temporal Factor)
Aggregate features appears in hour j for Pi

Fig. 2. Framework to represent ICU patients and perform mortality prediction.

a classification model (several considered and evaluated in this work). Details about the components of this framework are provided in the next sub-sections.

3.1 Temporal Tree

The Temporal Tree technique [21] is used to capture and model the compound information present in EHR data. A Temporal Tree is a temporal hierarchical structural network which is constructed based on the temporal co-occurrence of clinical events. An example of Temporal Tree is shown in Fig. 1b. In Temporal Tree, compound information is generated based on the local neighbourhood relationships between clinical events and is represented in a hierarchical form. Here, a leaf node represents the actual clinical events that occur at particular time, while the upper levels of the tree represent compound information. An approach to model event information across multiple levels of the Temporal Tree is to generate compound terms by re-labelling the Temporal Tree using the Weisfeiler-Lehman graph kernels re-labelling method [22]. The Temporal Tree technique offers the following advantages:

1. It models the temporal aspects of the data, along with other complex properties typical of EHR data. Each subtree, S_i, is constructed based on temporal information. A temporal event can encompass any number and any type of attributes (thus modelling multivariate data). Each subtree only considers the available data and there is no need of ad-hoc policies to deal with missing data (thus tackling issues related to irregular and sparse data). The representation also models sparsity in the data due to the frequency and time interval in which events occur, e.g., laboratory test data is often reported at a larger time interval than chart event data. Finally, the heterogeneous nature of EHR data is tackled by the use of abstract values (the more detail is described in Sect. 3.2).
2. No domain knowledge is required to construct the temporal tree.

3.2 Abstract Values

EHR data often contains both numerical and categorical values. To deal with this heterogeneous data, for all the clinical events, we convert each numerical value to one of five categories: Very Low (VL), Low (L), Normal(N), High(H), Very High(VH), using value abstraction [3]. We refer to these as *abstract values* or simply *values*. But in case of SOFA events, we put sofa score as categorical value (refer to Sect. 4.2 for more details).

3.3 Baseline and Aggregate Features

Most existing methods for mortality prediction only use basic temporal features, e.g., sequences of clinical events with their values – we shall refer to those as *baseline features*. However, EHRs contain many compound information which express very important temporal relationships between clinical events; these can be used as additional discriminative features for prediction tasks. While existing methods do not consider these kinds of temporal relationships, they can be captured by constructing a Temporal Tree. Then, the baseline and the additional features from the Temporal Tree can be combined to form a set of *Aggregate Features*. Note that baseline features are also directly extractable from the Temporal Tree: these are the features captured at Level 2 of the Temporal Tree hierarchy (see Fig. 1b). Each Temporal Tree (a tree is constructed for each patient) is used to assemble all baseline features and the Term Frequency - Inverse Document Frequency (TF-IDF) weighting scheme is applied to extract the top additional features in such a way that the amount of additional features is equivalent to the amount of baseline features. The *Aggregate Features* are constructed by appending the additional features to the baseline features.

3.4 Embeddings

Once the Temporal Tree for a patients is constructed, the patient and their patient situations are embedded into fix length lower dimensional vector. For this, we use Paragraph Vector [14], a state-of-art embedding technique. To prepare data ready for training the embedding model, we consider each node label as a word and each Temporal Tree as a document. We then traverse the Temporal Tree using Breath First Search (BFS) to generate the sequence of clinical events which is then the input used for training the embedding model (EM). The EM generates both patient and clinical event embeddings. To generate the patient situation embeddings, we consider a patient situation as being compose of many clinical events (including compound information) at a specific time (or within a short time interval) and thus feed these events into the EM to generate an embedding for each patient situation. Each patient is considered as a sequence of patient situations and a patient vector can thus be generated by taking the average of its patient situation vectors. Note that patient vectors are generated thus in two ways: (1) by representing a patient with a bag-of-words (BOW) of the clinical events relevant to the patient, or, alternatively, (2) by representing a patient with the sequence of relevant patient situations. The advantage of this second approach is that the embeddings constructed in that way preserve the temporal information about the patient and the sequence of clinical events, while the BOW approach does not preserve such important information.

4 Evaluation Methodology

In this work we are aim to answer the following research questions:

1. Do the aggregate features generated using the Temporal Tree model improve ICU mortality prediction?
2. Does the modelling of patients temporal information ICU mortality prediction, compared to modelling clinical events as bag-of-words?

To this aim, we set up an empirical evaluation that considers classifiers used for the task ICU mortality prediction and explore the impact different representation settings have on the effectiveness of the classifiers. The reminder of this section describes the experiment settings, while empirical results are reported in Sect. 5.

4.1 Dataset and Patient Cohort Selection

To evaluate the proposed approach we use MIMIC III[1] [12], a dataset of publicly available de-identified ICU encounters. It contains structured (e.g., real time sensor data, laboratory tests, prescriptions) as well as unstructured data (e.g., free-text clinical notes) for more than 60,000 ICU admissions between 2001 and 2012 from a US hospital.

The dataset is processed so as to select a cohort of patients that provide a meaningful mean of evaluation mortality prediction methods. Patients were selected according to the following criteria: (i) adults (patients aged 16 years or more), (ii) length of stay in ICU greater than 24 h, and 48 h respectively[2], (iii) have at least one vital signs entry recorded in the dataset (see Table 2 for more details), (iv) have been admitted to ICU for the first time. The reason for excluding re-admitted patients is that it is likely a patient is re-admitted for the same condition, and thus the data would show a high correlation; in addition there are only a small number of re-admissions in the dataset that satisfy the other criteria.

Following the criteria above, we generate a dataset where each ICU admission is regarded as a unique patient. This data is then randomly split into training (this is further divided for folds for cross-validation) and testing subsets using a 80:20 ratio. Details of the patient cohort used in our evaluation are provided in Table 1. Note the dataset presents a bias towards survival (not died) patients.

[1] https://mimic.physionet.org/.

[2] We consider two settings, one based on a minimum length of stay of 24 h, and one of 48 h.

Table 1. Details regarding the patient cohort used for evaluation – patient data is extracted from the MIMIC III dataset.

Time interval	Subset	Number of patients	Not dead (N)	Dead (n)
24 h	Training	25,889	24,130	1,759
	Testing	6,473	6041	432
	Total	32,362	30,171	2,191
48 h	Training	15,947	14,759	1,188
	Testing	3,988	3,687	301
	Total	19,935	18,446	1,489

4.2 Feature Selection

For the experiment, we use the structured data present in MIMIC III for the patient cohort detailed in Sect. 4.1; this includes vital signs, laboratory tests, and static information (demographic information and type of admission). Table 2 reports the information types extracted for each patient. The vital signs are highly sampled while laboratory tests are irregular. Along with this information, we also add the SOFA score [25]: this measures the severity of organ disfunction (score between 0 and 4) for six organ systems: respiration, coagulation, liver, cardiovascular, neurological, and renal. The higher the value, the higher the severity of the dysfunction for that patient. The overall severity is calculated by adding all the individual scores of each organ system. At each time point, or within a short time interval[3], we capture the following information: static information, vital signs, laboratory tests and individual and overall SOFA score. As there are many missing observations which may impact the calculation of the SOFA severity score, the SOFA score at any point in time is calculated based on the latest six hours observations.

4.3 Experiment Setting

In our experiments, patient mortality is predicted for two time intervals: (a) after 36 h from admission at ICU, based on the first 24 h of data, and (b) after 72 h from admission at ICU, based on the first 48 h of data. We intentionally consider a time gap between the observation data and the outcome being predicted. This is so that the medical practitioners would have enough time to consider the output of the system and administer a treatment plan. Both the considered time interval, and the gap between observed data and predicted outcome are commonly used in the relevant prior works is counted which is common in literature because it allows enough time for the medical practitioners to take proper decision such as intervene or other treatment parameter settings: specifically, Legman et al. [15]

[3] We consider one hour as per interval because the clinical events that occur in close temporal proximity often have a stronger relationship than those far away, at least in the ICU context.

Table 2. Information about a patient captured in the Temporal Tree.

Static information	Admission type, gender, age
Vital signs	SpO2, Arterial PaCO2, Arterial pH, Heart Rate, Arterial Blood Pressure Systolic, Arterial Blood Pressure Diastolic, Respiratory Rate, GCS - Eye Opening, Temperature Celsius, Inspired O2 Fraction, GCS - Verbal Response, GCS - Motor Response, Anion Gap, Prothrombin Time
Laboratory tests	Bicarbonate, Bilirubin - Total, Calcium - Total, Chloride, Creatinine, Glucose, Potassium, Sodium, Urea Nitrogen, Hematocrit, White Blood Cells, Hemoglobin, Magnesium, INR(PT), Phosphate, pH, Lactate, Platelet Count
SOFA score	respiration, coagulation, liver, cardiovascular, cns, renal

have considered the 24 h setting, Harutyunyan et al. [11] have considered the 48 h setting, while Suresh et al. [24] have considered both settings.

4.4 Evaluation Measure

We use Area Under the Curve (AUC) for measuring the effectiveness of the studied methods for ICU mortality prediction. AUC has been consistently used as the target evaluation measure in prior work that addressed this prediction task [1,11,16,17,20,23,24]. Classifiers are trained and validated using 5-fold cross validation on the training data, and then evaluated on the withhold testing data for the purpose of model comparison and effectiveness. This process is repeated 10 times and effectiveness is averaged to weed out bias due to the random partition of the training data for validation purposes into 5-folds.

The MIMIC-III dataset contains mortality information for each patient including date and time of death, if death occurred. This information is used as ground truth to evaluate the classifiers.

4.5 Baselines and State-of-art Methods

The problem of ICU mortality prediction for a given patient is cast into the problem of assigning a binary label (died, not died) to a patient, given the patient's EHR data as input. To investigate the effectiveness of the representation method proposed in this paper, the following baseline as well as state-of-art classification methods are implemented.

- *Logistic Regression (LR)* is a popular method used in numerous previous work that has considered the problem of ICU mortality prediction [11,15,16].
- *Support Vector Machines (SVM)* have also been often applied to the ICU mortality prediction problem, e.g., [9].
- *Random Forest (RF)* has been shown to provide the highest mortality prediction accuracy among a number of other alternative methods when evaluated

for predicting the six-month mortality in a population of elderly Medicare beneficiaries [17].

- *Gradient-Boosted Trees (GBDT)* produce an ensemble of weak prediction models, typically decision trees. Darabi et al. [8] have shown that GBDT provide high effectiveness for the task of predicting the 30-day mortality risk after admission to ICU. This however was demonstrated on a small dataset only. We further note that their method, like ours, relied on the use of medical embeddings, but in their case this was so as to reduce the dimensionality of the data.
- *Gaussian Naive Base (GaussianNB)* is a simple statistical classifier technique based on the Bayes Theorem, and thus forms a naive baseline.
- *Extreme Gradient Boosting (xgbGBT)* is a specialisation of the Gradient Boosting method which uses a more regularised model formalization for controlling over-fitting, often delivering better classification performance.
- *K-Nearest Neighbourhood (KNN)* exploits proximity among the representation of data items for classification. In the situation at hand, it assumes that similar patients are likely represented by embeddings that are close to each other. It is clear then that the effectiveness of the KNN method depends on the fidelity of the representation in terms of maintaining the similarity between patients. In our implementation, cosine similarity is used to determine the similarity between patients embeddings. The value of k is determined from training data. Specifically, 10-fold cross validation on the training data is used to tune the value of k with respect to AUC; k is varied in the range $[0, 100]$ with step 2.
- *Aczon2017:* Aczon et al. [1] consider the task of predicting the mortality risk for paediatric critical care patient. To that aim, they construct a model architecture comprised of three LSTMs. We replicate this approach by considering the patient situation vectors as the input for the LSTM architecture.
- *Suresh2018:* Suresh et al. [24] propose a method that comprises two steps: (1) Learn relevant patient subgroups in an unsupervised manner. They use a LSTM autoencoder to produce dense representations and then use a Gaussian Mixture Model (GMM) to identify the patient group. (2) Use multitask learning for each separate subgroup. In our experiments, we adapt the same GMM method to cluster the patient into three groups. To this aim, patients embeddings generated by the methods described in Sect. 3.4 are used as input for clustering. Then the same single task and multi-task model is used for each group. Effectiveness is measured across subgroups and averaged into a single value.

In the case of *LR*, *SVM*, *RF*, *GBDT*, *GaussianNB*, *xgbGBT* and *KNN*, patient embeddings are considered as the input for training and testing the model. Conversely, for *Aczon2017* and *Suresh2018*, patient situation embeddings are considered as the input for training and testing the model.

5 Empirical Results

5.1 Effectiveness of Aggregate Features

We consider whether the use of aggregate features as extracted from Temporal Tree improve the effectiveness of classifiers on the task of ICU mortality prediction. To this aim, we consider the results obtained (1) When considering the patient as a sequence of clinical events, i.e. a BOW approach. The results obtained for this setup are reported in Table 3). (2) When preserving patients temporal information in terms of patients situations, and thus considering patients as sequences of patients situations. The results obtained for this setup are reported in Table 4).

Aggregate features improve ICU mortality prediction effectiveness (AUC) when using a BOW-based approach of on average 7.814% (for 24 h) and 4.889% (for 48 h), compared to when aggregate features are not considered. This occurs across all considered classifiers, and the highest AUC is obtained when using the *KNN* approach, for both time intervals.

When analysing the effect preserving temporal information has on the results, we find that aggregate features provide between 3.130% (for 24 h) and 2.562% (for 48 h) average improvements in terms of AUC. Moreover, all classifiers obtain higher prediction effectiveness when aggregate features are used, except when considering KNN for predicting mortality in the 48 h setting. In both time settings, the best effectiveness is provided by the SVM classifier when considering aggregate features.

Among all classifiers, KNN and SVM, when used on aggregate features, provide the highest ICU patient mortality predictions, independently of the time interval considered. We stress the fact that the effectiveness of the KNN method is highly dependent on the representation chosen for the patients. Thus, the fact that KNN is among the best classifiers, if not the best, when using the proposed representation techniques, speaks in favour of the Temporal Tree approach coupled with the embedding method, and specifically of their ability to effectively model patient similarity.

Table 3. ICU mortality prediction effectiveness (AUC) across methods and for the bag-of-word approach. Results are reported distinguished between the 24hr and 48hr setup. Standard deviation is provided in brackets and represents the variation obtained across different rounds of tuning of the learnt classifier.

Classifers	24hr_baseline	24hr_aggregate	% Improvement	48hr_baseline	48hr_aggregate	% Improvement
LR	0.836 ± 0.002	0.862 ± 0.001	3.110	0.818 ± 0.002	0.838 ± 0.002	2.445
SVM	0.820 ± 0.004	0.851 ± 0.003	3.780	0.814 ± 0.004	0.816 ± 0.004	0.246
RF	0.724 ± 0.015	0.773 ± 0.009	6.768	0.716 ± 0.01	0.727 ± 0.009	1.536
GBDT	0.788 ± 0.008	0.849 ± 0.003	7.741	0.775 ± 0.006	0.818 ± 0.005	5.548
GaussianNB	0.474 ± 0.006	0.613 ± 0.003	29.325	0.513 ± 0.002	0.622 ± 0.004	21.248
XgbGBT	0.849 ± 0.003	0.867 ± 0.002	2.120	0.810 ± 0.004	0.833 ± 0.002	2.840
KNN*	0.864 ± 0.000	**0.880 ± 0.000**	1.852	0.832 ± 0.000	**0.835 ± 0.000**	0.361
Average			**7.814**	Average		**4.889**

Table 4. ICU mortality prediction effectiveness (AUC) across methods and for the approach that preserves temporal information. Results are reported distinguished between the 24hr and 48hr setup. Standard deviation is provided in brackets and represents the variation obtained across different rounds of tuning of the learnt classifier.

Classifers	24hr_baseline	24hr_aggregate	% Improvement	48hr_baseline	48hr_aggregate	% Improvement
LR	0.832 ± 0.000	0.870 ± 0.001	4.567	0.795 ± 0.001	0.831 ± 0.001	4.528
SVM	0.847 ± 0.001	**0.875 ± 0.001**	3.306	0.821 ± 0.001	**0.844 ± 0.001**	2.801
RF	0.760 ± 0.005	0.781 ± 0.009	2.763	0.743 ± 0.008	0.745 ± 0.012	0.269
GBDT	0.842 ± 0.002	0.867 ± 0.002	2.969	0.823 ± 0.004	0.829 ± 0.003	0.729
GaussianNB	0.558 ± 0.002	0.582 ± 0.002	4.301	0.563 ± 0.003	0.592 ± 0.002	5.151
XgbGBT	0.853 ± 0.002	0.874 ± 0.002	2.462	0.831 ± 0.002	0.838 ± 0.002	0.842
KNN*	0.852 ± 0.000	0.862 ± 0.000	1.174	0.820 ± 0.000	0.810 ± 0.000	−1.220
Aczon2017	0.829 ± 0.024	0.858 ± 0.014	3.498	0.757 ± 0.06	0.813 ± 0.019	7.398
Suresh2018-Single	0.831 ± 0.000	0.839 ± 0.000	1.003	0.769 ± 0.00	0.784 ± 0.000	1.950
Suresh2018-Multitask	0.832 ± 0.000	0.842 ± 0.000	1.161	0.781 ± 0.000	0.771 ± 0.000	−1.323
Average			**2.720**	Average		**2.113**

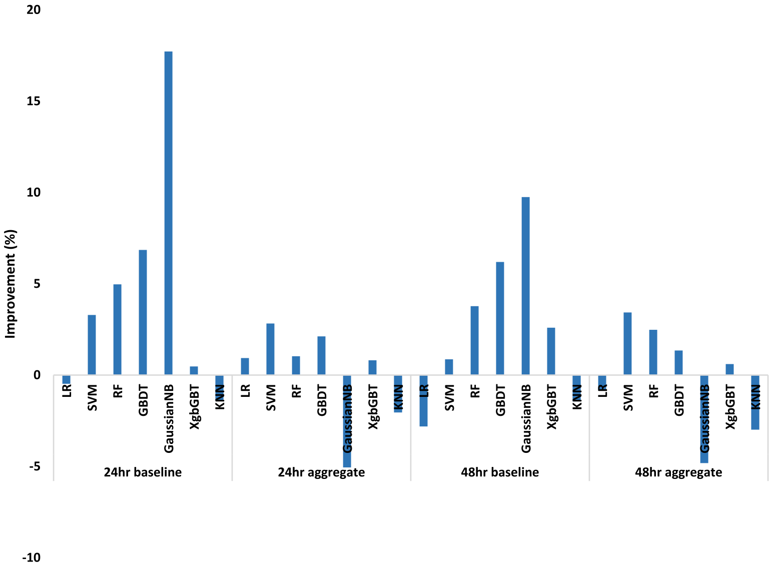


Fig. 3. Performance Improvement of Temporal Information over Bag-Of-Word Approach.

Finally, the empirical results also clearly suggest that ICU mortality prediction is easier when performed on the first 24 h of data obtained after admission, compared to the 48 h setting. We note a that similar result was found in previous studies, e.g., [24].

5.2 Impact of Modelling Temporal Information

We now consider whether modelling temporal information provide an edge over the more simplistic bag-of-word approach. Figure 3 shows the percentage improvement while considering patient's temporal information (i.e. patient situations) over bag-of-word indicated by y-axis value. SVM, RF, GBDT, and

XgbGBT always provide better performance when preserving temporal information; the results however exhibits the opposite trend when KNN is considered, while LR and GaussianNB provide mixed results. Overall, we found that, in most cases, preserving temporal information leads to better mortality prediction accuracy.

6 Conclusion and Future Work

In this paper we have investigated the problem of ICU mortality prediction from EHR data: this is a challenging but important prediction task as improvements in prediction accuracy translate into better clinical decision support and thus likely better healthcare delivery. To provide an effective mean for accurately predicting patient mortality, in this paper we propose to represent patients EHR data using the Temporal Tree technique, a recently introduced method for representing compounded EHR data. This method is used to generate patients and patients situation embeddings, which are then used as the input to a suite of common and state-of-the-art classifiers for the ICU mortality prediction task. Our extensive empirical results on a dataset of real ICU EHR data demonstrate that compound information generated by Temporal Tree is useful for producing discriminative representations, which in turn improve the mortality prediction accuracy of the considered classification methods. In addition, the results also demonstrate that preserving temporal information leads to further gains in effectiveness in most cases and settings.

References

1. Aczon, M., et al.: Dynamic mortality risk predictions in pediatric critical care using recurrent neural networks. Stat 1050, 23 (2017)
2. Bajor, J.M., Mesa, D.A., Osterman, T.J., Lasko, T.A.: Embedding complexity in the data representation instead of in the model: a case study using heterogeneous medical data. arXiv preprint arXiv:1802.04233 (2018)
3. Batal, I., Valizadegan, H., Cooper, G.F., Hauskrecht, M.: A temporal pattern mining approach for classifying electronic health record data. ACM Trans. Intell. Syst. Technol. (TIST) **4**(4), 63 (2013)
4. Breslow, M.J., Badawi, O.: Severity scoring in the critically ill: part 1–interpretation and accuracy of outcome prediction scoring systems. Chest **141**(1), 245–252 (2012)
5. Chen, W., Long, G., Yao, L., Sheng, Q.Z.: AMRNN: attended multi-task recurrent neural networks for dynamic illness severity prediction. World Wide Web **23**(5), 2753–2770 (2019). https://doi.org/10.1007/s11280-019-00720-x
6. Choi, E., et al.: Multi-layer representation learning for medical concepts. In: Proceedings of the 22nd ACM SIGKDD, pp. 1495–1504. ACM (2016)
7. Coopersmith, C.M., et al.: A comparison of critical care research funding and the financial burden of critical illness in the united states. Crit. Care Med. **40**(4), 1072–1079 (2012)
8. Darabi, H.R., Tsinis, D., Zecchini, K., Whitcomb, W.F., Liss, A.: Forecasting mortality risk for patients admitted to intensive care units using machine learning. Procedia Comput. Sci. **140**, 306–313 (2018)

9. Ghassemi, M., et al.: Unfolding physiological state: mortality modelling in intensive care units. In: Proceedings of the 20th ACM SIGKDD, pp. 75–84. ACM (2014)
10. Glicksberg, B.S., et al.: Automated disease cohort selection using word embeddings from electronic health records. In: PSB, pp. 145–156. World Scientific (2018)
11. Harutyunyan, H., Khachatrian, H., Kale, D.C., Ver Steeg, G., Galstyan, A.: Multitask learning and benchmarking with clinical time series data. Sci. Data **6**(1), 96 (2019)
12. Johnson, A.E., et al.: MIMIC-III, a freely accessible critical care database. Sci. Data **3**, 160035 (2016). https://doi.org/10.1038/sdata.2016.35
13. Knaus, W.A., et al.: The apache iii prognostic system: risk prediction of hospital mortality for critically iii hospitalized adults. Chest **100**(6), 1619–1636 (1991)
14. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196 (2014)
15. Lehman, L.W., Saeed, M., Long, W., Lee, J., Mark, R.: Risk stratification of ICU patients using topic models inferred from unstructured progress notes. In: AMIA Annual Symposium Proceedings, vol. 2012, p. 505. American Medical Informatics Association (2012)
16. Luo, Y., Xin, Y., Joshi, R., Celi, L., Szolovits, P.: Predicting ICU mortality risk by grouping temporal trends from a multivariate panel of physiologic measurements. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
17. Makar, M., Ghassemi, M., Cutler, D.M., Obermeyer, Z.: Short-term mortality prediction for elderly patients using medicare claims data. Int. J. Mach. Learn. Comput. **5**(3), 192 (2015)
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
19. Moreno, R.P., et al.: Saps 3-from evaluation of the patient to evaluation of the intensive care unit. part 2: development of a prognostic model for hospital mortality at ICU admission. Intensive Care Med. **31**(10), 1345–1355 (2005)
20. Nori, N., Kashima, H., Yamashita, K., Ikai, H., Imanaka, Y.: Simultaneous modeling of multiple diseases for mortality prediction in acute hospital care. In: Proceedings of the 21th ACM SIGKDD, pp. 855–864. ACM (2015)
21. Pokharel, S., Zuccon, G., Li, X., Utomo, C.P., Li, Y.: Temporal tree representation for similarity computation between medical patients. Artif. Intell. Med. **108**, 101900 (2020)
22. Shervashidze, N., Schweitzer, P., Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-lehman graph kernels. J. Mach. Learn. Res. **12**(Sep), 2539–2561 (2011)
23. Shi, Z., Chen, W., Liang, S., Zuo, W., Yue, L., Wang, S.: Deep interpretable mortality model for intensive care unit risk prediction. In: Li, J., Wang, S., Qin, S., Li, X., Wang, S. (eds.) ADMA 2019. LNCS (LNAI), vol. 11888, pp. 617–631. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-35231-8_45
24. Suresh, H., Gong, J.J., Guttag, J.V.: Learning tasks for multitask learning: heterogenous patient populations in the ICU. In: Proceedings of the 24th ACM SIGKDD, pp. 802–810. ACM (2018)
25. Vincent, J.L., et al.: The sofa (sepsis-related organ failure assessment) score to describe organ dysfunction/failure. Intensive Care Med. **22**(7), 707–710 (1996)
26. Xu, Y., Zhang, Z., Lu, G., Yang, J.: Approximately symmetrical face images for image preprocessing in face recognition and sparse representation based classification. Pattern Recogn. **54**, 68–82 (2016)
27. Zhang, J., Kowsari, K., Harrison, J.H., Lobo, J.M., Barnes, L.E.: Patient2vec: a personalized interpretable deep representation of the longitudinal electronic health record. IEEE Access **6**, 65333–65346 (2018)

Pre-trained StyleGAN Based Data Augmentation for Small Sample Brain CT Motion Artifacts Detection

Kang Su[1](✉), Erning Zhou[1], Xiaoyu Sun[2], Che Wang[2], Dan Yu[2], and Xianlu Luo[1](✉)

[1] Neusoft Institute Guangdong, Foshan 528225, China
{sukang,luoxianlu}@nuit.edu.cn
[2] Neusoft Education Technology Group, Dalian 116023, China

Abstract. A serious issue with deep learning in medical applications is the limited availability of labeled medical image data. One of the data argumentation approach, Generative Adversarial Networks (GANs), can generate new samples that are drawn from the distribution of original dataset. Unfortunately, small sample medical imaging data is always not sufficient to train GANs with millions of parameters. Therefore, this paper proposes a pre-trained Style-based Generative Adversarial Networks (StyleGAN) to transfer knowledge from the Magnetic Resonance Imaging (MRI) domain to Computed tomography (CT) domain with limited sample images. This first pre-trained StyleGAN based Data augmentations (DA) method can generate high quality 256 × 256 CT artifacts and artifact-free images for CT motion artifacts detection. Furthermore, we demonstrate this technique on a CT motion artifacts classification task and have achieved an improvement of 5.59% in sensitivity using synthetic images. Our results indicate that pre-trained models can provide priori knowledge to overcome the small sample problem in medical image processing.

Keywords: Data augmentation · Generative Adversarial Network · Transfer learning · Brain CT · Motion artifacts detection

1 Introduction

Artificial intelligence techniques, in particular deep convolutional neural networks (CNN) have revolutionized medical image analysis across large number of applications ranging from image classification and region localization to organ segmentation [1]. In all cases, large-scale annotated medical imaging data have been utilized to alleviate the risk of overfitting. However, many medical applications do not have access to big data, since obtaining such a large medical imaging dataset is both expensive and time-consuming. As a promising data argumentation method inspired by game theory, Generative Adversarial Networks (GANs) can create novel unseen samples, which are difficult to collect. As a result, GAN have made significant successes in many computer vision tasks.

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 339–346, 2020.
https://doi.org/10.1007/978-3-030-65390-3_26

Most GAN models are trained from scratch. Although the use of pre-trained CNN in discriminative tasks has been demonstrated outperformed, or in the worst case, as well as CNN trained from scratch [2]. Transferring approach for generative model has received much less attention, possibly due to its great complexity, as two different networks of GANs are trained together. Recently, Wang et al. [3] studied fine-tune transferring GANs and showed that pre-trained model generally have better results than the scratch, and these valuable pre-trained weights cannot be learned when only limited data is available. However a similar approach is missing in medical image analysis, so far as our knowledge goes.

Computed tomography (CT) plays an import role in medical image while artifacts which are caused by multiple mechanisms seriously affect doctors accurate diagnoses. CT artifacts can be group into four categories according their underlying causes, including physics-based, patient-based, scanner-based, helical and multisection artifacts [4]. The goal of this work is to generate brain CT motion artifacts/artifact-free images using pre-trained model.

Research Questions. We mainly address three questions:

- How can we generate high quality brain CT artifact-free images given limited training images?
- How can we synthesize realistic brain CT motion artifacts images from artifact-free images?
- How can we improve the performance of brain CT artifacts detection using synthetic data examples?

Contributions. The contributions of our work are as follows:

- Image generation with small sample dataset. We use pre-trained StyleGAN for brain CT artifact-free images generation, and show pre-trained model can provide priori knowledge to overcome the small sample constrain.
- Two-stage transfer learning. We synthesize brain CT motion artifacts images using pre-trained weights from artifact-free model.
- Brain CT artifacts detection. We evaluate our synthetic data examples by training them with a ResNet classifier, which successfully detects CT artifacts and achieve a classification improvement of 5.59% in sensitivity using synthetic images.

2 Related Work

Generative adversarial networks were first described by Ian Goodfellow [5] in 2014 to tackle the generative tasks using deep learning methods. The GAN model involves a generator and discriminator module, which are trained in an adversarial manner.

As for GAN based data augmentations in medical imaging, Deep Convolution Generative Adversarial Networks (DCGAN) based 56 × 56 CT lung nodule generated samples were verified similar to the original samples [7]. Conditional

Progressive Growing of GANs (PGGAN) based 256 × 256 generated MRI images improved sensitivity in brain tumor diagnosis [8]. Also, there are some literatures reported that GAN based data augmentations could boost classification and segmentation from other medical scanners [9].

Our work is the first pre-trained GAN based DA in CT medical imaging. As the number of parameters in GANs is large, small sample medical imaging data is always not sufficient to train GAN network. Our proposed method is less likely to suffer from overfitting with embedded prior knowledge and boost the CT motion artifacts detection using synthetic data.

3 Proposed Approach

3.1 Datasets

This study included source domain brain MRI images and target domain brain CT images. Two popular MRI image datasets used for StyleGAN source model training: ACRIN-FMISO-Brain (ACRIN 6684) and LGG-1p19qDeletion. Both datasets are publically available at The Cancer Imaging Archive (TCIA). We selected a total number of 24,133 slices of 512 × 512 DICOM images, which contents both healthy individuals and tumor patients. Our CT datasets contains 1072 practical brain CT images with motion artifacts and 4285 artifact-free images. All the CT images are in the size of 512 × 512. When synthesized artifacts/artifact-free images are added to real dataset, the number of artifacts images to artifact-free images is 1:4. We split the data in training and validation sets (70%/30%, respectively) with stratified sampling.

3.2 Image Synthesis

The proposed method for small sample CT images data argument using pre-trained StyleGAN model is illustrated in Fig. 1. The details of proposed approach will be described in the following subsections.

StyleGAN-Based Source Model. For MRI source model training, we followed the StyleGAN architecture as described in [6], which is consisted of a novel style-controlled generator and a binary classifier discriminator. The StyleGAN generator differs from original GAN generator, as it involves a mapping network and the noise injections. The mapping network accepts a d-dimensional Gaussian latent vector z, and maps z to an intermediate space by an 8-layer MLP, which then controls the generator style through AdaIN operation. To introduce fine-grained variation, the Gaussian noise is injected to the feature maps prior the AdaIN operation. The loss function we used for training is the non-saturating loss with R1 regularization, which is given by:

$$L_D = \mathbb{E}_{x \sim p_{data}}[f(-D(x))] + \mathbb{E}_{z \sim p_z}[f(D(G(z))] \tag{1}$$

$$L_G = \mathbb{E}_{z \sim p_z}[g(-D(G(z))] \tag{2}$$

where $f(x) = g(x) = log(1 + e^x)$.

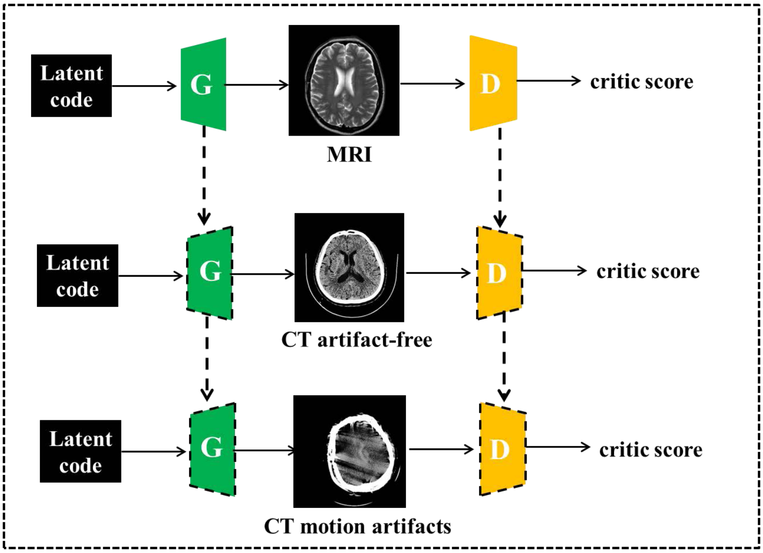


Fig. 1. Framework of our approach for pre-trained StyleGAN based data augmentation. MRI dataset is fed to the StyleGAN model to obtain the source model, then CT artifact-free images are synthesized with the StyleGAN architecture using pre-trained weights from MRI dataset. Finally, CT artifacts images are generated by using pre-trained weights on CT artifact-free dataset.

Pre-trained StyleGAN. Our goal is to generating both artifact-free and motion artifacts images given limited samples. As shown in Fig. 1, CT artifact-free images are synthesized with the StyleGAN architecture using pre-trained weights from MRI domain. CT artifacts images in are generated by using pre-trained weights from artifact-free images, considering the fact that CT artifacts and artifact-free images share basically structural information in common.

In our strategy, we used a straightforward but effective method to transfer embedded prior knowledge. We initialize both the discriminator and generator of target model using pre-trained weights from the source model. To overcome the problem of overfitting, an effective L2-SP regularization [10] is used during transferring process, which is given by:

$$\Omega(w) = \frac{\alpha}{2}||w - w^0||_2^2 + \frac{\beta}{2}||w||_2^2 \tag{3}$$

where w_0 and w respectively denote source network parameters and target network parameters.

Implementation Details. To obtain the source model, the MRI dataset was firstly trained for 200000 iterations with a batch size of 8, the learning rate for both generator and discriminator is 0.001, the Adam optimizer was used with momentum (beta1 = 0, beta2 = 0.99). Next, we used the MRI source model of resolution 256 × 256 for pre-training and fine-tuned on the CT brain artifact-free dataset for 100000 iterations without progressive training, the batch size was 8 and learning rate was 0.001. To generate CT motion images, we reused the pre-trained CT artifact-free model and fine-tuned it following the previous pre-training scheme. All the models described in this paper were implemented using Pytorch v 1.0.

3.3 Evaluation

We measured the ability and usefulness of generative image with a pre-trained ResNet18 network. As one of the standard ImageNet architecture, ResNet model has been extensively used in medical transferring learning tasks. When appending the generated images for training the ResNet classifier, the classification performance is evaluated by both sensitivity and specificity.

4 Experimental Results and Discussion

In this section we present a set of experiments and results. To obtain a proper source model for pre-training, we trained a StyleGAN network (see Sect. 3.2) from scratch by feeding the MRI dataset. Then, the StyleGAN source model was used for fine-tuning on CT artifact-free dataset. Next, we implemented a similar pre-training scheme for CT motion artifacts generation. Finally, we analyzed the effect of data augmentation for training a ResNet classifier using synthesized CT images.

4.1 StyleGAN Based MRI Source Model

In order to obtain a source model for pre-training, we trained a StyleGAN model with MRI dataset. Figure 2 illustrates examples of MRI images generated by StyleGAN model trained from scratch. From visual confirmation, it successfully captures the texture and structural information of original MRI images. The number of training dataset is critical to obtain realistic images, especially in high resolution. When trained StyleGAN model with a medium size dataset (∼2.4 W), synthetic MRI image maintains realism and diversity at resolution 256 × 256, which is a high level resolution for typical GAN models.

4.2 Pre-trained StyleGAN Based CT Images Generation

The second step of the experiment consisted of CT images generation for data augmentation using pre-trained StyleGAN model. We incorporated prior knowledge from MRI domain to adapt the CT data distribution. As can be seen in Fig. 3, when training dataset become scarce, training from scratch leads to blurry images. The skull boundary and tissue fine details in scratch result images are

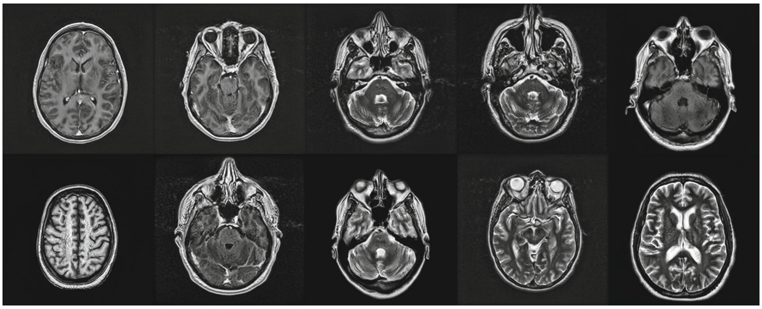


Fig. 2. Example synthetic 256 × 256 MRI image samples generated by StyleGAN model trained from scratch.

not as shaper as in pre-trained result images in Fig. 4. Besides, more blob artifacts resemble water droplets appear at tissue region, which are probably related to StyleGAN generator architecture.

Furthermore, since our CT artifacts dataset is too small for effective training, we only show the pre-trained result. Results in Fig. 5 shows that synthetic CT motion artifacts images vary in shapes, sizes, intensities and locations, and maintains structural information such as the long range blurring streaks at desired position. All these factors are critical for training a good classifier.

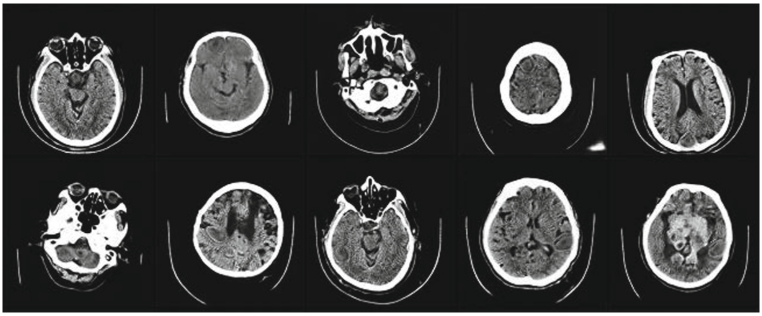


Fig. 3. Example synthetic 256 × 256 CT artifact-free image samples generated by StyleGAN model trained from scratch.

4.3 Image Classification Performance

Table 1 shows the CT artifacts detection results W/O artifacts DA. As expected, the sensitivity/specificity remarkably increase with the additional synthetic data. In practical CT artifacts scenarios, much attention is paid to the sensitivity of artifact images. For artifact samples, we see an improvement in the classification

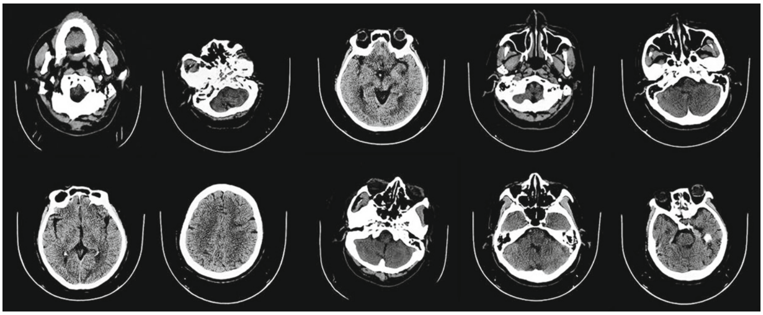


Fig. 4. Synthetic 256 × 256 CT image samples generated by the StyleGAN model pre-trained on MRI datasets.

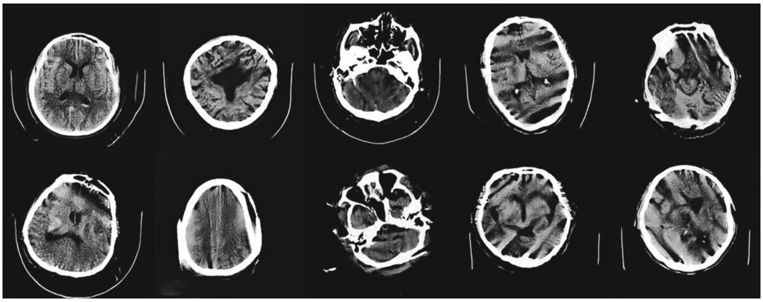


Fig. 5. Example synthetic 256 × 256 CT artifacts samples generated by the StyleGAN model pre-trained on CT artifact-free model.

Table 1. CT artifacts detection measurements using DA proposed in this paper

	Artifacts		Artifact-free	
	Sensitivity	Specificity	Sensitivity	Specificity
3000 real images	86.83%	83.21%	95.44%	96.53%
+1000 generated images	90.38%	85.71%	96.37%	97.65%
+2000 generated images	88.74%	86.99%	96.67%	97.16%
+4000 generated images	**92.42**%	**87.13**%	**96.74**%	**98.16**%
+6000 generated images	90.31%	86.35%	96.53%	97.62%
+8000 generated images	92.02%	84.02%	95.98%	98.13%

results from 86.83% to 92.42% in sensitivity, using the pre-trained StyleGAN-based DA, an increase of 5.59% in sensitivity is achieved. In our case, adding only 4000 synthetic images to 3000 real samples yield the best sensitivity improvement, adding more synthetic data leads to an decrease in the classification result. Our results indicates that appending additional synthetic images to real samples during training phase does not contribute to CT artifacts detection performance.

5 Conclusion

This work focused on generating synthetic CT images to enlarge sample datasets and improve the performance for CT artifacts detection. We presented a method that uses the pre-trained StyleGAN for data augmentation. The experiment was carried out on a limited dataset of CT motion artifacts/artifact-free images. We demonstrated this technique on a classification task and achieved an improvement of 5.59% in sensitivity using synthetic data, which indicates that pre-trained models can provide priori knowledge to overcome the small sample problem in medical image processing. In the future, we plan to extend our work to transfer knowledge from multiple domains to further improve the GAN model performance.

References

1. Anwar, S.M., Majid, M., Qayyum, A., Awais, M., Alnowami, M., Khan, M.K.: Medical image analysis using convolutional neural networks: a review. J. Med. Syst. **42**(11), 1–13 (2018). https://doi.org/10.1007/s10916-018-1088-1
2. Tajbakhsh, N., et al.: Convolutional neural networks for medical image analysis: full training or fine tuning? IEEE Trans. Med. Imaging **35**(5), 1299–1312 (2016)
3. Wang, Y., Wu, C., Herranz, L., van de Weijer, J., Gonzalez-Garcia, A., Raducanu, B.: Transferring GANs: generating images from limited data. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 218–234 (2018)
4. Boas, F.E., Fleischmann, D.: CT artifacts: causes and reduction techniques. Imaging Med. **4**(2), 229–240 (2012)
5. Goodfellow, I.J., et al.: Generative adversarial nets. Adv. Neural Inf. Process. Syst. **27**, 2672–2680 (2014)
6. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4401-4410 (2019)
7. Chuquicusma, M. J., Hussein, S., Burt, J., Bagci, U.: How to fool radiologists with generative adversarial networks? A visual turing test for lung cancer diagnosis. In: 2018 IEEE 15th International Symposium on Biomedical Imaging, pp. 240–244 (2018)
8. Han, C., Murao, K., Noguchi, T., Kawata, Y., Uchiyama, F., Rundo, L., Satoh, S.I.: Learning more with less: conditional PGGAN-based data augmentation for brain metastases detection using highly-rough annotation on MR images. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 119–127 (2019)
9. Ben-Cohen, A., et al.: Cross-modality synthesis from CT to PET using FCN and GAN networks for improved automated lesion detection. Eng. Appl. Artif. Intell. **4**, 186–194 (2019)
10. Li, X., Grandvalet, Y., Davoine, F. Explicit inductive bias for transfer learning with convolutional networks. arXiv preprint arXiv:1802.01483 (2018)

Motion Artifacts Detection from Computed Tomography Images

Xiaoyu Sun[1,2](✉), Feng Huang[3], Guanjun Lai[1,2], Dan Yu[1,2], Bin Zhang[1,2],
Baozhu Guo[1,2], and Zhuang Ma[1,2]

[1] Dalian Neusoft University of Information, Dalian, Liaoning, China
{jt_sunxiaoyu,laiguanjun,yudan,
zhangbin_jt,jt_guobaozhu,jt_mazhuang}@neusoft.edu.cn
[2] Dalian Neusoft Educaiton Technology Group Co. Limited, Dalian, Liaoning, China
[3] Neusoft Medical Systems Co. Limited, Shenyang, Liaoning, China
ethan@neusoftmedical.com

Abstract. Motion artifacts detection is essential for computed tomography (CT) imaging, and it can be concerned as a binary classification problem where images with artifacts are positive samples and images without artifacts are negative samples. However, there are two main challenges for this problem: (a) how to extract features of motion artifacts from CT images, and (b) with limited labeled data, how to ensure high sensitivity and generality of the training model. To address these challenges, we first develop a preprocessing procedure, the Motion Artifacts Enhancement Method (MAEM), to extract features effectively. Subsequently, a Motion Artifacts Detection Algorithm based on Convolutional Neural Network (MADA-CNN) is presented to construct the classification model. Performance is evaluated by the area under the receiver operation characteristics curve (AUC). Compared with traditional preprocessing method on single classifier, the MAEM shows the AUC of 0.9570 (improved +1.96%) and sensitivity of 92.66% (improved +3.59%). To validate the generality of the proposed method, the ensemble model shows the AUC of 0.9665 and sensitivity of 94.50%. Experimental results have demonstrated the effectiveness and generality of our method.

Keywords: Motion artifacts detection · Computed tomography images · Convolutional neural network

1 Introduction

Computed tomography (CT) has become a preferred technique in diagnosis and therapy for its high resolution and low acquisition time [1]. However, motion artifacts in CT can seriously affect the quality of images and even lead to misdiagnosis [2]. Figure 1(a)(b) and (c)(d) show examples of head CT images with motion artifacts and without artifacts respectively. These motion artifacts in CT images may confuse clinical diagnosis of lesions and even cause serious aftereffects on patients' health. Therefore, it is very important and meaningful to detect motion artifacts from CT images.

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 347–359, 2020.
https://doi.org/10.1007/978-3-030-65390-3_27

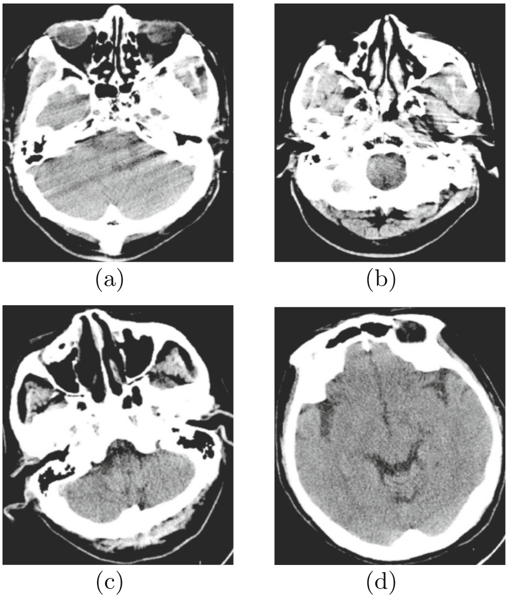


Fig. 1. Examples of head CT images. (a)(b) Images with motion artifacts. (c)(d) Images without artifacts.

In clinical practice, radiologists generally detect artifacts of CT images based on their experience. While manual detection is a time-consuming and laborious process, and evaluation criteria of artifacts are fixed and susceptible to subjective factors. Some studies use quantitative methods to assess image quality, such as PSNR (Peak Signal to Noise Ratio), MSE (Mean Square Error), Structural Similarity (SSIM), Feature Similarity Indexing Method (FSIM), UIQI (Universal Image Quality Index), FSIM (Feature Similarity Index Method), etc. However, these quantitative criteria have different applicability and limitations [3].

Currently, deep learning, especially convolutional neural network has many significant applications in medical imaging such as imaging reconstruction, classification, segmentation and registration, computer-aided detection and diagnosis [4]. However, there are two major challenges in dealing with artifacts detection by deep learning.

The first challenge is about how to effectively extract artifacts features from images. Many preprocessing methods such as filtering, sonograms, normalization, etc. mainly focus on artifact correction, artifact reduction or other aspects. Unfortunately, few studies can provide an effective preprocessing method to extract features for artifact detection problem. Besides, traditional data augmentation operations such as geometric transforms and color transforms are mostly used to process natural images, which are not exactly applicable for medical images.

The second challenge is about how to ensure high sensitivity and generality of the model with limited labeled training data. Three points should be taken into account to deal with this challenge: (a) the lack of large samples of labeled training data, (b) superior performance of the algorithm (especially sensitivity) in clinical practice requirements and (c) generality of the model.

Based on the above discussions, we propose a novel method to detect motion artifacts from CT images. Particularly, we aim to solve a binary classification problem (positive: motion artifact, negative: no artifact). We summarize contributions of our work as follows:

- We develop a novel motion artifacts enhancement method (MAEM) for image preprocessing to extract artifacts features.
- We present a motion artifact detection algorithm based on convolutional neural network (MADA-CNN). We transfer and fine-tune the CNN parameters to build multiple classifiers. Then, the top N classifiers are included to build an ensemble model.
- We apply our algorithm on a real-world data set to investigate the effectiveness and generality of the algorithm. Experiments show good performance on evaluation criteria.
- In medical clinical practice, we provide a precise computer-aided motion artifacts detection method for radiologists and efficiently optimize the quality inspection workflow.

The reminder of this paper is organized as follows: Sect. 2 presents the related works. Section 3 gives a description of our methodology in details. Section 4 describes experiments and evaluations. Finally, Sect. 5 remarks on conclusions of this paper.

2 Related Works

Artifacts can dramatically degrade the quality of CT images to make diagnostically unusable. In study of artifacts from medical imaging, most researchers focus on image reconstruction by reducing artifacts of projections. Compressive sensing (CS), the total variation (TV), dictionary learning algorithms (DL) and statistical nearest neighbors (SNN) are powerful approaches to suppress artifacts for image reconstruction. Additionally, with recent advances in medical imaging technologies, approaches of artifact reduction using deep learning have been widely used [5]. However, only a few studies devoted their attentions to artifacts detection problem. Stoeve *et al.* [6] proposed an approach to detect motion artifacts in Confocal Laser Endomicroscopy (CLE). They used histogram of oriented gradient (HOG) for preprocessing and then constructed artiNet for classification based on transfer learning from Inception v3 network. Wei *et al.* [7] extracted seven features from ROIs and used random forests to detect metal steak artifacts in head and neck CT images. Welch *et al.* [8] interpolated CT volumes using SimpleITK linear resampling image filter to reduce variability

Table 1. A brief summary of artifacts detection problem in the reviewed publications.

Ref.	Image type	Artifacts type	Preprocessing method	Classification
[6]	CLE	Motion artifacts	HOG	ArtiNet
[7]	CT	Metal artifacts	Features extraction and PCA	Random forest
[8]	CT	Dental artifacts	Geometrical transforms	3D CNN
[10]	CCTA	Motion artifacts	Coronary motion forward artifact model	ResNet
[11]	CT	Multi artifacts	FBP for projection reconstruction	CNN

within images and performed some geometrical transforms for data preprocessing. The three-dimensional CNN was trained to classify dental artifacts statuses. Each preprocessing approach mentioned above is specific to a particular problem. Particularly, different types of images, different types of artifacts may have greatly different preprocessing methods. Therefore, the existing preprocessing methods are not fully applicable to solve the task of motion artifacts detection from CT images.

In order to address a practical problem of limited availability of sample data, generative adversarial networks (GANs) and filtered back projection (FBP) are two promising approaches for synthetic data augmentation. GANs generates fake medical images based on game theory principles between generator and discriminator [9]. FBP algorithm introduces simulated artifacts to generate artifact samples. Lossau *et al.* [10] adopt filtered back projection algorithm as a forward model to generate motion artifacts data respectively. Then CNN was applied to classify CT motion artifacts in coronary arteries. Prakash *et al.* [11] developed a deep learning-based artifact detection method for CT images. Artifacts were introduced in the projection data as gain shift by using FBP algorithm for reconstruction. These studies generate large amounts of fake data for supervised learning [12]. Unfortunately, synthetic images, especially high-resolution images generation, can lead to distribution biases from the real ones. Simultaneously, it will cause inferior performance of the algorithm on real test samples and fail to meet clinical requirements.

Table 1 summarizes papers related to artifacts detection. As it can be seen from the table, most researches only use one single classifier (which is usually called base classifier) to solve classification problem. However, there are many theoretical studies show that overfitting, high variance and instability would happen in base classifier. Consequently, the generality ability of base classifier is not strong enough to get a better performance on test set.

3 Methods

The problem of motion artifacts detection of CT images is defined as a binary classification problem, where positive samples are motion-artifact images and negative samples are non-artifact images. We propose a Motion Artifacts Detection Algorithm based on Convolutional Neural Network (MADA-CNN) to detect motion artifacts from CT images automatically. The framework of the MADA-CNN method is shown in Fig. 2, which mainly includes four stages: image preprocessing, model construction, model selection and integration, and model evaluation.

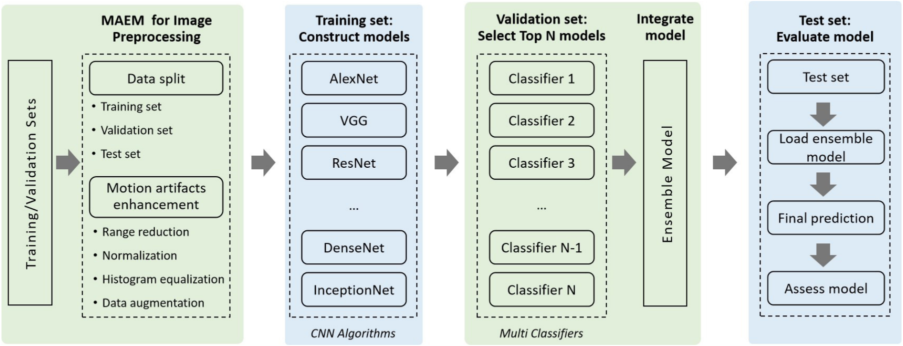


Fig. 2. The framework of the MADA-CNN method.

3.1 MAEM for Image Preprocessing

We present Motion Artifacts Enhancement Method (MAEM) for image preprocessing. The purpose of the MAEM is to extract and enhance the local and detailed features of motion artifacts, so as to improve the performance of classifiers.

Firstly, we divide the dataset into training set, validation set, and test set with the proportion of 80%, 10% and 10%. The number of samples in each data set is subject to the original sample proportion distribution.

Secondly, before training a deep learning network, it is particularly critical to extract artifact regions of interest through effective preprocessing method. According to the unique characteristics of motion artifacts, we extract and enhance features by the following steps: range reduction, normalization, histogram equalization and data augmentation. Detailed descriptions are given below.

Range Reduction. This step is to retain artifact features and remove irrelevant regions. The Hounsfield unit (HU) is a dimensionless unit frequently used in CT scan and its value quantifies the tissues absorption of X-rays beam. The range of a head CT image is usually from −3000 to +2000HU, but −3000HU represents four black border areas without any useful information. Thus, we reduce the range of each image to extract the ROIs of motion artifacts.

Normalization. Normalization generally speeds up training and leads to faster convergence. We scale data by min-max normalization.

Histogram Equalization. This step is to improve contrast and enhance artifact features of images. As artifacts of CT image are local features represented by close contrast values, histogram equalization can effectively stretch out the intensity range of the image.

Data Augmentation. Data augmentation can expand the diversity of images and improve the generality of deep learning models. We apply geometric transforms, such as random flip, rotation, translation and scaling to augment data utility.

Figure 3 shows an example of original image. After applying the MAEM for preprocessing, we get the resulting image in Fig. 4. It is obvious that areas of lower local contrast gain a higher contrast and obtain an enhanced artifact features. Moreover, these enhanced features are of great help for artifact recognition.

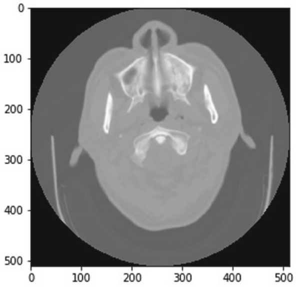


Fig. 3. Original image.

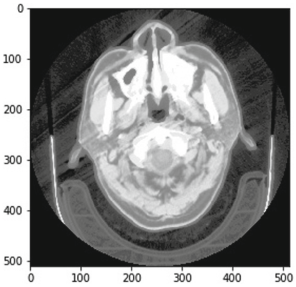


Fig. 4. MAEM preprocessed image.

3.2 Model Construction

We construct the model based on transfer learning. The CNN has made great success in image detection. In addition, in order to deal with the scarcity of labeled data in medical images, transfer learning is considered as an effective way to train CNN deep learning models. The process of training model based on transfer learning is: (a) pre-training: transfer the ImageNet pre-trained model as initial weights; (b) fine-tuning: adjust the fully connected layers and the number of classes, on this basis, use training set in our task to fine-tune network parameters for further training.

Network Architecture. We select different CNN architectures to verify the applicability of different transferred models in our problem. The network architectures include AlexNet, SqueezeNet, VGG, DenseNet, Inception v3 and ResNet.

Loss Function. Artifact detection is an imbalanced binary classification problem. Hence, we use weighted cross entropy as the loss function, and the weight is set to the inverse of the ratio of the number of two classes.

Other Parameters. Other parameters are set as follows.

- Optimizer: Stochastic gradient descent (SGD) optimizer. Because SGD can get better training results with lower computational complexity.
- Momentum: 0.9
- Weight decay: 6e−3
- Learning rate: 1e−3
- Batch size: 64
- Number of iterations: 200.

3.3 Model Selection and Integration

Model Selection. Due to the issue of class imbalance in our dataset, accuracy cannot objectively measure an algorithms performance, so that can result in a serious bias towards the majority class. To alleviate this issue, the AUC (the area under the receiver operating characteristic curve) is preferred as a typical criterion for evaluating the performance of algorithms with imbalanced data.

Therefore, the basis of model selection is the highest AUC of the validation set. For each network architecture, we choose the corresponding iteration with the highest AUC as the optimal classifier of this network. Since we have multiple network architectures, the multiple classifiers models are constructed.

Model Integration. In order to improve the generality of the algorithm, we apply ensemble learning to integrate multiple models. The ensemble model can obtain a better performance than any of the constituent classifiers alone does. According to the AUC ranking of validation set, we select the top N classifiers to construct the ensemble model. Finally, the majority voting of N classifiers computes the final prediction of the ensemble model, which expressed as Eq. (1).

Displayed equations are centered and set on a separate line.

$$H(x) = sign(\sum_{t=1}^{T} w_t h_t(y|x)) \tag{1}$$

Displayed equations are centered and set on a separate line.

$$h(x) \in \{-1, +1\} \tag{2}$$

Where t = $\{1, 2,$T$\}$ denotes the set of sub-classifier; x denotes the input vector of image; y denotes the probability of class y, w_t denotes the weight of classifier t. For a binary classification, $h_t(x)$ represents the prediction of classifier t (defined as Eq. (2)).

3.4 Model Evaluation

We evaluate the model performance by AUC, accuracy, sensitivity, and specificity on test set. Moreover, we analyze the results from two perspectives.

From an algorithm perspective, AUC is a typical criterion to measure the effectiveness of the model. That is, the higher AUC, the better performance of the model.

From a clinical perspective, sensitivity is preferred to be the most important indicator that radiologists would care most. In artifact detection, sensitivity measures the ability of a model to correctly identify those images with artifacts (also known as TPR, true positive rate). The higher sensitivity means the higher probability that artifact image is detected. Correspondingly, the smaller probability of missing detection. (Example: A test with 80% sensitivity will identify 80% of images that have artifacts, but will miss 20% of images that have artifacts.)

Therefore, we comprehensively consider both algorithm and clinical factors, and focus on sensitivity under the guarantee of AUC.

4 Experiments and Results

4.1 Dataset

A total of 10,552 head CT slices were collected from Neusoft Medical CT scanner. The ground-truth of images were labeled as a binary classification problem by a neuroradiologist with 7 years' experience, including 2,177 positive samples (with motion artifacts) and 8,375 negative samples (without artifacts). We split the dataset into training set, validation set and test set at a ratio of 80%:10%:10%. Table 2 shows the detailed information. All experiments are implemented in Python 3.7, 4 NVIDIA Tesla V100-SXM2-32 GB GPUs, and CUDA V10.1.

Table 2. Detailed information of dataset.

Dataset (10,552 images)	# Overall	# Positive (with artifacts)	# Negative (without artifacts)
Training set	8546	1763	6783
Validation set	950	196	754
Test set	1056	218	838

4.2 Experiment 1: Comparison Results on Single Classifier

To illustrate the effectiveness of our image preprocessing approach, we compare the MAEM method with baseline T-Pre method on different CNN backbones.

Baseline Method: The traditional image preprocessing method (T-Pre) is used to construct the baseline model (denoted as Baseline). The process of T-Pre mainly includes normalization and data augmentation.

Our Proposed Method: The MAEM described in Sect. 3.1 is used to construct the proposed model (denoted as Proposed).

Figure 5 and Fig. 6 show the ROC curves of baseline and the proposed method respectively. From the average evaluation values of different backbones, our method is superior to baseline method. The baseline method obtains mean AUC of 0.822, while our method obtains mean AUC of 0.923, which achieves average improvements by 12.29%. Obviously, the MAEM provides an effective image preprocessing approach that shows better performance than traditional method in different CNN networks.

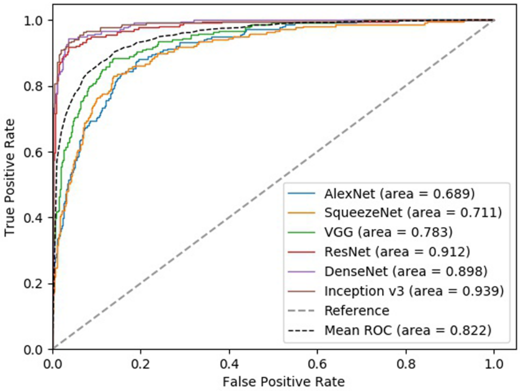


Fig. 5. ROC curves of baseline method.

Furthermore, we compare two methods that perform best on single classifier in Table 3 (with the highest AUC criterion). It can be seen that the proposed method presents an improvement of convergence rate by +62.79%, AUC by +1.96%, accuracy by +1.08%, sensitivity by +3.59%, and specificity by +0.49%. Our method shows the optimal performance under Inception v3 architecture. The improvement of convergence speed and AUC verify the efficiency and effectiveness of the proposed algorithm. It is worth noting that the sensitivity increased significantly by +3.59%. Since radiologists are most concerned about sensitivity criterion, the improvement of sensitivity is of great significance for clinical application.

In current experiment, our method even works well with single classifier model. In further experiment, we will consider multi-model selection and integration with our preprocessing approach.

Table 3. Comparison results on single classifier.

Method	Backbone	Best epoch		AUC		Accuracy		Sensitivity		Specificity	
		Value	Gain%	Value	Gain%	Value	Gain%	Value	Gain%	Value	Gain%
Baseline	Inceptionv3	86	–	0.94	–	0.97	–	0.89	–	0.98	–
Proposed	Inceptionv3	32	(+62.79)	0.96	(+1.96)	0.98	(+1.08)	0.93	(+3.59)	0.99	(+0.49)

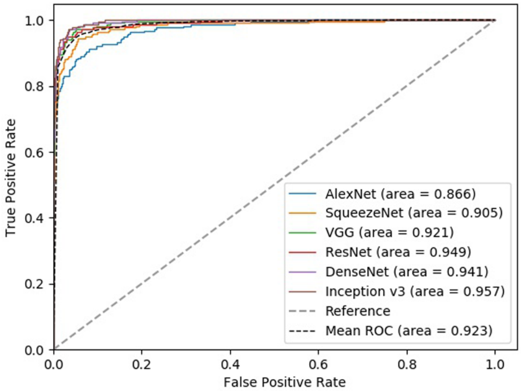


Fig. 6. ROC curves of the proposed method.

4.3 Experiment 2: Selection Model and Integration Model

Experiment 1 has proved that the MAEM performs well on image preprocessing and benefits for classification. On the basis of the MAEM, the goal of experiment 2 is to select the top N classifiers on validation set and then construct the final ensemble model.

Model Selection on Validation Set. Firstly, we rank models according to their performance on validation set. The ranking result of evaluation performance is shown in Table 4 (in reverse order of AUC). Where, the column Index = {1,2,M} represents the ranking of M models, when Index = 1 represents the optimal model (Backbone: ResNet). Therefore, if we select the top N models (N <= M), it can be expressed as the corresponding models of Index = {1,2,N}.

Table 4. Ranking results of AUC on validation set.

Index	Backbone	AUC
1	ResNet	0.9609
2	Inception v3	0.9601
3	DenseNet	0.9501
4	VGG	0.9367
5	SqueezeNet	0.9066
6	AlexNet	0.8741

Ensemble Model and Evaluation on Test Set. According to the ranking of validation set, we integrate the top N models and evaluate them on test set. When the values of N are different, we perform a set of comparative experiments, as shown in Table 5. In particular, when N = 1, it is the optimal single model selected on validation set (ResNet). We compare each ensemble model with the optimal single model, where "Gain" represents the improvements of ensemble model.

Table 5. Comparison results on single classifier.

Top N	AUC		Accuracy		Sensitivity		Specificity	
	Value	Gain%	Value	Gain%	Value	Gain%	Value	Gain%
N = 6	0.9460	(−0.31)	0.9735	(+0.85)	0.8991	(−2.29)	0.9928	(+1.71)
N = 5	0.9414	(−0.81)	0.9716	(+0.68)	0.8899	(−3.48)	0.9928	(+1.71)
N = 4	0.9665	(+1.83)	0.9792	(+1.47)	0.9450	(+2.49)	0.9881	(+1.23)
N = 3	0.9625	(+1.41)	0.9782	(+1.37)	0.9358	(+1.50)	0.9893	(+1.35)
N = 2	0.9644	(+1.61)	0.9678	(+0.29)	0.9587	(+3.98)	0.9702	(−0.60)
N = 1	0.9491	(0.00)	0.9650	(0.00)	0.9220	(0.00)	0.9761	(0.00)

Next, we compare results in two perspectives: algorithm performance and clinical application. Then we recommend different integration schemes according to different scenarios.

From the perspective of algorithm performance, the top 4 ensemble model (when N = 4) performs best on test set. Compared with optimal single model (when N = 1), all evaluation criteria have been improved. Specifically, AUC increased by +1.83%, accuracy increased by 1.47%, sensitivity increased by +2.49%, and specificity increased by +1.23%. However, the disadvantage is that loading four models for computation will have higher time complexity and space complexity, which demands higher computing power.

From the perspective of clinical application, since radiologists are more concerned about whether all artifacts images are detected, the top 2^{nd} ensemble model (when N = 2) obtains the maximum improvements of sensitivity by +3.98%. Although there will be a slight misjudgment (specificity is reduced by -0.60%), sensitivity has been more significantly improved by +3.98%. Moreover, the memory space and running speed will be improved than that of N = 4. If the ensemble model is embedded in CT scanning devices, then N = 2 will be more suitable for clinical application.

In summary, under high computing power environments, N = 4 can be selected to provide more accurate prediction results. While under clinical conditions, N = 2 can be chosen to identify more artifacts with the lowest missed detection rate.

On test data set, this study proves high performance and effectiveness of the algorithm. Basically, it can replace manual detection of CT motion artifacts and can assist radiologists to provide an automatic artifact detection method in clinical practice.

5 Conclusions

This paper presents an automatic detection method for motion artifacts from CT images. Firstly, the MAEM is performed for image preprocessing to enhance artifact features. Then, the MADA-CNN algorithm is applied to construct and integrate classification models. Finally, experiments have verified the effectiveness and generality of our method.

Our contributions are as follows:

- **Superior performance for artifacts detection.** We introduce the MAEM for preprocessing and construct binary classification model by the MADA-CNN algorithm to detect motion artifacts of CT images. High effectiveness of the proposed model can provide precise and real-time auxiliary support for clinical practice.
- **Ensemble model recommendation for different scenarios.** We integrate models to optimize single classifier, and recommend different combinations of ensemble models based on different application scenarios and conditions. With high computing power, more alternative models can be selected for integration to obtain the highest overall performance, as AUC of 0.9665 and sensitivity of 94.50%. In clinical environment, we can attain higher sensitivity with fewer integrated models, as AUC of 0.9644%, and sensitivity of 95.87%.
- **High generality of our model.** Generality is an important property in model migration and practical application. The ensemble model obtains an improvement of AUC by +1.83% and sensitivity by +2.49% compared with single classifier, which proves a strong robustness of our proposed ensemble model. That is of great significance for model application in different situations.

Our current solution only focused on motion artifacts detection of CT images. Our further study will enrich the diversity of training samples, such as collecting different types of medical images with different types of artifacts. Therefore, we may be able to train a unified model to solve artifact detection problem and to build the trained model into medical devices.

References

1. Larson, D.B., Johnson, L.W., Schnell, B.M., et al.: National trends in CT use in the emergency department. Radiology **258**(1), 164–173 (2011)
2. Boas, F.E., Fleischmann, D.: CT artifacts: causes and reduction techniques. Imaging Med. **4**(2), 229–240 (2012)
3. Sara, U., Akter, M., Uddin, M.S., Fleischmann, D.: Image quality assessment through FSIM, SSIM, MSE and PSNRA comparative study. J. Comput. Commun. **7**(3), 8–18 (2019)

4. Lee, J.G., Jun, S., Cho, Y.W., et al.: Deep learning in medical imaging: general overview. Korean J. Radiol. **18**(4), 570–584 (2017)
5. Haskins, G., Kruger, U., Yan, P.: Deep learning in medical image registration: a survey. Mach. Vision Appl. **31**(1), 1–18 (2020). https://doi.org/10.1007/s00138-020-01060-x
6. Stoeve, M., et al.: Motion artifact detection in confocal laser endomicroscopy images. Bildverarbeitung für die Medizin 2018. I, pp. 328–333. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-662-56537-7_85
7. Wei, L., Rosen, B., Vallires, M., et al.: Automatic recognition and analysis of metal streak artifacts in head and neck computed tomography for radiomics modeling. Phys. Imaging Radiat. Oncol. **10**, 49–54 (2019)
8. Welch, M.L., McIntosh, C., Purdie, T.G., et al.: Automatic classification of dental artifact status for efficient image veracity checks: effects of image resolution and convolutional neural network depth. Phys. Med. Biol. **65**(1), 015005 (2020)
9. Armanious, K., Jiang, C., Fischer, M., et al.: MedGAN: medical image translation using GANs. Comput. Med. Imaging Graph. **79**, 101684 (2020)
10. Lossau, T., Nickisch, H., Wissel, T., et al.: Motion artifact recognition and quantification in coronary CT angiography using convolutional neural networks. Med. Image Anal. **52**, 68–79 (2019)
11. Prakash, P., Dutta, S.: Deep learning-based artifact detection for diagnostic CT images. In: Medical Imaging 2019: Physics of Medical Imaging, pp. 109484C. International Society for Optics and Photonics (2019)
12. Zhang, Z., Zhu, Q., et al.: Discriminative margin-sensitive autoencoder for collective multi-view disease analysis. Neural Netw. **123**, 94–107 (2020)

Loners Stand Out. Identification of Anomalous Subsequences Based on Group Performance

Martha Tatusch(✉), Gerhard Klassen, and Stefan Conrad

Heinrich Heine University, Universitätsstr. 1, 40225 Düsseldorf, Germany
{tatusch,klassen,stefan.conrad}@hhu.de

Abstract. Time series analysis is a part of data mining and nowadays an important field of research due to the increasing amount of data that is recorded sequentially by various systems. Especially the identification of anomalous subsequences arouses great interest, since a manual search for errors or malfunctions is not possible in most cases. Often outliers are defined as points or sequences that deviate significantly from the course of one or multiple time series, yet there are also applications where the trend rather than the exact course of time series is relevant. In that case, there is an approach of clustering the time series per time point and analyzing their cluster transitions over time. Sequences that change their cluster members suddenly or often, indicate an anomaly.

In 2019, a novel approach for the detection of these transition-based outliers was introduced [19]. Now, we present an algorithm called DACT (**D**etecting **A**nomalies based on **C**luster **T**ransitions) that is able to identify outlier sequences of the same type. It is a simple approach that stands out due to different results, although a similar type of anomalies is targeted. In the evaluation, we examine and discuss the differences. Our experiments show, that the results are competitive and reasonable.

Keywords: Outlier detection · Time series analysis · Clustering

1 Motivation

Due to the increasing popularity of digital systems such as social platforms, online shops or simple database applications in various industries, data analysis is of steadily growing importance. The analysis of sequential data forms an important part of this field of research and is known as *time series analysis*. There are several applications which consider either single or multiple time series whereby these can be univariate or multivariate. In this work, we focus on multiple multivariate time series and the behavior of subsequences with regard to their peers. There are many applications where these conditions apply. For example, when investigating a drug's tolerance on humans, one time series per patient can be extracted whereby various features per timestamp are recorded. In our approach, we examine the trend of groups of time series rather than the exact course,

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 360–369, 2020.
https://doi.org/10.1007/978-3-030-65390-3_28

as it is not relevant in many applications. To do so, it is necessary to previously cluster the data for each point in time. Regarding the drug tolerance behavior, the patients may be grouped by their state of health. Since every human body is unique, these clusters may change over time. Some of these changes are normal, but if a patient shows any irregularity, action must be taken. In order to detect such irregularities automatically, we introduce DACT (**D**etecting **A**nomalies based on **C**luster **T**ransitions), an anomaly detection algorithm for transition-based outliers. To the best of our knowledge, the first approach regarding this type of outliers was published in 2019 [19]. Hence, in the following we will compare DACT with it.

2 Foundation

In order to provide a good basis for the comparison of the two methods, the same definitions as given in [19] are used in this work.

Definition 1 (Time Series). *A multivariate time series* $T = o_{t_1}, ..., o_{t_n}$ *is an ordered set of* n *real valued data points of arbitrary dimension. The data points are chronologically ordered by their time of recording.*

Definition 2 (Data Set). *A data set* $D = T_1, ..., T_m$ *is a set of* m *time series of same length and equivalent points in time. The set of data points of all time series at a timestamp* t_i *is denoted as* O_{t_i}.

Definition 3 (Subsequence). *A subsequence* $T_{t_i,t_j,l} = o_{t_i,l}, ..., o_{t_j,l}$ *with* $j > i$ *is an ordered set of successive real valued data points beginning at time* t_i *and ending at* t_j *from time series* T_l.

Definition 4 (Cluster). *A cluster* $C_{t_i,j} \subseteq O_{t_i}$ *at time* t_i*, with* $j \in \{1, ..., q\}$ *being a unique identifier (e.g. counter), is a set of similar data points, identified by a cluster algorithm or human.*

Definition 5 (Cluster Member). *A data point* $o_{t_i,l}$ *from time series* T_l *at time* t_i*, that is assigned to a cluster* $C_{t_i,j}$ *is called a member of cluster* $C_{t_i,j}$.

Definition 6 (Noise). *A data point* $o_{t_i,l}$ *from time series* T_l *at time* t_i *is considered as noise, if it is not assigned to any cluster.*

Definition 7 (Clustering). *A clustering is the overall result of a clustering algorithm or the set of all clusters annotated by a human for all timestamps. In concrete it is the set* $\zeta = \{C_{t_1,1}, ..., C_{t_n,q}\} \cup Noise$.

3 Related Work

There are various approaches for identifying irregularities in time series. In some applications, the detection of single anomalous data points is of interest.

This problem is for example addressed by prediction-based algorithms like autoregressive-moving-average (ARMA) models [2,6,15]. In other cases, the identification of so called *changing points* [7,13], which indicate a change of the previous course, are relevant. Although these techniques perform very well in most cases, they cannot be used for our purpose. First, in contrast to DACT, they target single data points, not subsequences. Second, they lack the correlation of one time series to others. There are also other algorithms for the detection of outliers, which decompose the time series with techniques like STL [4] before analyzing them. However, these methods only work if the considered time series can be actually decomposed. In many applications, this is not the case. When regarding anomalous subsequences, there are various works using dynamic time warping (DTW) [17] for the comparison of time series or neural networks [3,10,16]. Another approach is the detection of the most unusual subsequences (discords) using a symbolic aggregation of a time series [8,9,12]. Even though these methods are aiming at subsequences, they only consider single time series and therefore cannot be used in our case.

The most recent works for the detection of outlier subsequences in multiple time series use Probabilistic Suffix Trees (PST) [18] or Random Block Coordinate Descents (RBCD) [21] regarding the deviation of one time series to the others. In contrast to our approach, the behavior of a time series with regard to its peers is not analyzed here. We accomplish this analysis by clustering the time series data per timestamp and investigating a time series' transitions between clusters. Such an approach was already presented in 2019 [19]. However, the procedure has some particularities that might be unfavorable depending on the application. For example, the procedure in [19] only penalizes splits of a time series from a cluster, whereas merges of smaller clusters into larger ones do not have a negative influence on the outlier score of the sequences involved. In this paper we introduce a simple approach which resolves these difficulties.

4 Model Description

After the time series data has been clustered per timestamp using an arbitrary clustering algorithm like DBSCAN [5] or k-means [14], DACT can be applied. In short, the procedure is based on the analysis of the average number of points in time that a time series migrates with its peers, which indicates a subsequence's stability over time. The longer a sequence moves with its cluster members over time, the more stable it is.

For the following presentation of the components of DACT we first introduce the cluster identity function *cid* of a data point $o_{t_i,l}$, which returns the cluster of the time series l at the considered timestamp t_i:

$$cid(o_{t_i,l}) = \begin{cases} \emptyset & \text{if } o_{t_i,l} \text{ is not assigned to any cluster} \\ C_{t_i,a} & \text{else} \end{cases}$$

Now, we can calculate the number of time points in which two subsequences $T_{t_i,t_j,l}$ and $T_{t_i,t_j,x}$ share the same cluster. We call it the shared time points count *stc*:

$$stc(T_{t_i,t_j,l}, T_{t_i,t_j,x}) = |\{t_k | cid(o_{t_k,x}) = cid(o_{t_k,l}) \ \wedge \ t_k \in [t_i, t_j]\}|$$

with $x \neq l$. In order to get the average number of time points a time series $T_{t_i,t_j,l}$ moves with its cluster members, we need to compute the number of peers of the time series during the considered time period. It describes the amount of distinct time series that are at least once assigned to the same cluster as T_l during the period. It can be calculated by the peer count *pc*:

$$pc(T_{t_i,t_j,l}) = |\{T_x \ | \exists t_k \in [t_i, t_j] : cid(o_{t_k,x}) = cid(o_{t_k,l})\}|$$

with $x \neq l$. We can now express the over-time stability *OTS* of a subsequence $T_{t_i,t_j,l}$ by

$$OTS(T_{t_i,t_j,l}) = \frac{\sum_{p=1}^{m} stc(T_{t_i,t_j,l}, T_{t_i,t_j,p})}{pc(T_{t_i,t_j,l}) \cdot k}$$

with k being the number of timestamps where T_l holds data. In order to detect anomalies in time series, this score needs to be included in an outlier score, which indicates whether a subsequence is conspicuous or not. In the following we propose two concepts for building the outlier score. Since we believe, that this score is dependent on the behavior of a subsequence's peers (an unstable sequence is not as conspicuous regarding an unstable cluster as it is in a stable one), both variants focus on the scores of the considered cluster. Before introducing these two concepts, we define the term *intuitive outlier*:

Definition 8 (Intuitive Outlier). *A sequence $T_{t_i,t_j,l}$ is called an intuitive outlier if its data points are marked as noise for every timestamp $t_k \in [t_i, t_j]$.*

This is necessary as the outlier score can only be calculated for subsequences whose data point at the last timestamp is assigned to a cluster. If it is not, it is not possible to determine a meaningful reference value.

4.1 Variant 1

The first approach focuses on the best stability score achieved in a cluster $C_{t_j,a}$ regarding a time period from t_i to t_j. Formally, it can be expressed by

$$best\_score(C_{t_j,a}, t_i) = max(\{OTS(T_{t_i,t_j,l}) \mid cid(o_{t_j,l}) = C_{t_j,a}\}).$$

It describes the highest score obtained by subsequences from t_i to t_j ending in cluster $C_{t_j,a}$. The outlier score DACT of a subsequence is then given by the deviation of its stability score from the best score:

$$DACT(T_{t_i,t_j,l}) = best\_score(cid(o_{t_j,l}), t_i) - OTS(T_{t_i,t_j,l}).$$

Obviously, the *best_score* represents the upper bound for the outlier score within a cluster for a given time period. This causes, that clusters containing stable subsequences are more sensitive to deviations than the ones containing less stable sequences. Finally, an outlier can be formally described using the outlier score.

Definition 9 (Outlier – Variant 1). *Given a threshold τ, a sequence $T_{t_i,t_j,l}$ is called an outlier if*

$$DACT(T_{t_i,t_j,l}) > \tau\ .$$

Since the best subsequence score of a cluster influences the highest possible outlier score, the threshold τ often has to be chosen rather central in the interval $[0, 1]$. Additionally, the best threshold differs for data sets with different distributions of the data points. The more scattered the data, the lower the threshold.

4.2 Variant 2

The second approach follows the statistical assumption that anomalies can be found with the help of their deviation from the standard deviation. For this, the mean of a cluster's stability scores regarding the start time t_i has to be determined first. Regarding a cluster $C_{t_j,a}$ for the time period from t_i to t_j, it is given by

$$\mu(C_{t_j,a}, t_i) = \frac{1}{|C_{t_j,a}|} \cdot \sum_{o_{t_j,l} \in C_{t_j,a}} OTS(T_{t_i,t_j,l}).$$

The standard deviation of a cluster's stability scores regarding the start time t_i can then be calculated by

$$\sigma(C_{t_j,a}, t_i) = \sqrt{\frac{1}{|C_{t_j,a}|} \cdot \sum_{o_{t_j,l} \in C_{t_j,a}} (\mu(C_{t_j,a}, t_i) - OTS(T_{t_i,t_j,l}))^2}.$$

In order to compare it later with the standard deviation, we formulate the outlier score sDACT of a subsequence $T_{t_i,t_j,l}$ as the absolute difference of its stability score and the mean of its last cluster:

$$sDACT(T_{t_i,t_j,l}) = |\mu(cid(o_{t_j,l}), t_i) - OTS(T_{t_i,t_j,l})|.$$

We call it sDACT in order to express, that the *statistical* variant is used. In the following, this score can be used to detect outliers by inspecting the deviation of it from the standard deviation. With the help of a factor ρ it can be formally described.

Definition 10 (Outlier – Variant 2). *Given a threshold ρ, a sequence $T_{t_i,t_j,l}$ is called an outlier if*

$$sDACT(T_{t_i,t_j,l}) > \rho \cdot \sigma(cid(o_{t_j,l}), t_i)\ .$$

Again, the outlier score is highly dependent on the performance of the considered cluster's members. Since the standard deviation is considered, the outlier score is even less sensitive to deviations, especially in the case of a rather unstable cluster. Therefore in most cases the default value of $\rho = 3$ will probably be to high in order to detect inconsistencies. In our method, frequently a value of around $\rho \approx 2$ is recommended. This factor naturally is also dependent on the distribution of the data.

5 Experiments

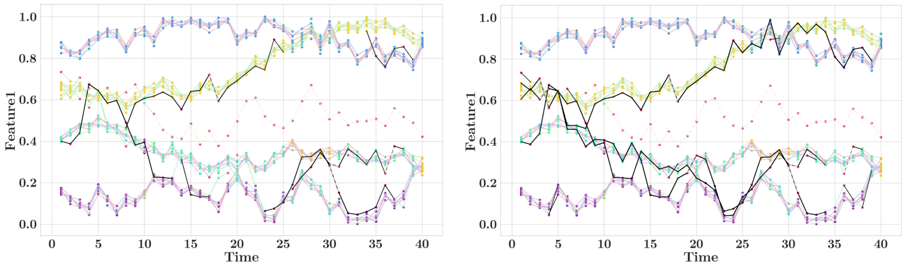
Following, experiments on a synthetic and a real world data set are discussed to evaluate the performance of the presented methods. In order to simplify referencing the approaches we will name them as follows:

- *referred method* – describes the approach from [19].
- *DACT* – stands for the presented method using variant 1 for the detection of outliers.
- *sDACT* – represents the approach using variant 2.

5.1 Artificially Generated Data Set

The first considered data set was artificially generated and contains 28 univariate time series (TS) with 40 timestamps. Initially four groups of TS were randomly generated. Afterwards, three targeted and one completely random outlier sequence were inserted. All data points of the completely random outlier TS were chosen randomly, whereby the distance between two consecutive points was set to not being greater than 0.1. The remaining outlier sequences were generated so that their data points were always located near to a cluster's centroid. An outlier sequence could change its cluster at the earliest if it was located for at least 5 time points in a cluster.

The experiment was performed with DACT and the referred method. In order to get comparable results, the same parameter settings for both approaches were chosen. For the clustering DBSCAN [5] was used with $\epsilon = 0.025$ and $minPts = 3$. The threshold τ was set to 0.55. Figure 1 shows the detected anomalies by DACT and the referred method. The colored dots represent cluster belongings whereby red dots indicate noise. The detected outlier sequences are illustrated as and intuitive outliers as dashed lines.



(a) DACT

(b) referred method

Fig. 1. Detected outliers on the generated data set with $\boldsymbol{\tau = 0.55}$, $\boldsymbol{minPts = 3}$ and $\boldsymbol{\epsilon = 0.025}$.

Both methods managed to detect the completely random as well as parts of the three targeted outliers. The referred method, however, marked a lot more parts as outliers than DACT. Regarding the uppermost outlier sequence from time point 10 to 39, there is a difference between both methods between time 25 and 34. DACT did not mark this part of the TS as an outlier although the referred method did. This can be explained by the fact, that the TS moves stably with most of its cluster members in this period. The merge of the two upper clusters causes lower stability scores, but since the size of both clusters is approximately the same, all cluster members are affected equally. The same applies to the split.

Considering the second lowest outlier sequence between timestamp 30 and 38, it is the other way around. While DACT marks the sequence as an outlier for the whole period, the referred method interprets the course between timestamp 34 and 36 as normal. On the one hand, this is caused by the decrease of the stability scores in the second lowest cluster. As there were merges and splits in the history of the cluster, all scores were negatively affected. On the other hand, there are only few members in the considered cluster and another sequence is marked as noise at time point 32, too. Between timestamp 34 and 36 the considered time series behaves stable, so that it does not stand out in contrast to its cluster members, regarding this short period. In contrast to that, DACT is more sensitive concerning short term changes, if only few time series are considered.

5.2 GlobalEconomy Data Set

The second data set is provided by the website theglobaleconomy.com [1]. It consists of over 300 indicators for different features of 200 countries for more than 60 years. For the experiments, we considered 20 different countries and two features (namely the education spendings and the unemployment rate) within the

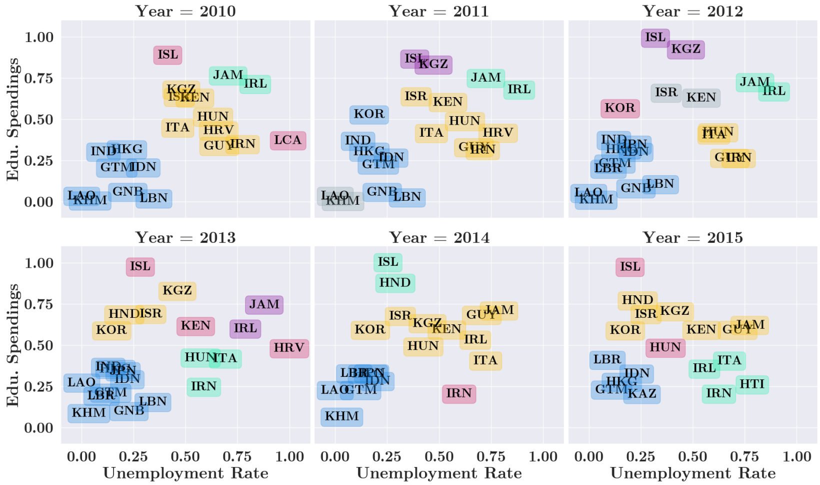


Fig. 2. Resulted clustering by DBSCAN with $\boldsymbol{minPts = 2}$ and $\boldsymbol{\epsilon = 0.19}$ on the GlobalEconomy data set.

period from 2010 to 2015 to enable a manageable illustration. Since the database is not complete for all country-year combinations, the amount of countries per timestamp may vary.

The experiment was run with all three methods using DBSCAN with $\epsilon = 0.19$ and $minPts = 2$. Since the underlying clustering for all three approaches is the same, it is illustrated separately in Fig. 2. Different colors represent different cluster belongings and noise data points are marked red. The resulting outlier sequences are listed in Table 1. The list was shortened so that in case of overlaps only the longest detected subsequence of a country is included per method. This time, the threshold parameters τ and ρ were chosen for all methods separately, as the first experiment showed that the same parameter setting led to considerably more outlier sequences with the referred method than with DACT. An individual parameter choice might therefore be appropriate.

Table 1. Resulting outlier sequences by DACT ($\boldsymbol{\tau = 0.3}$), sDACT ($\boldsymbol{\rho = 2}$) and the referred method ($\boldsymbol{\tau = 0.35}$) on the GlobalEconomy data set.

Country	Start	End	DACT	sDACT	referred
GUY	2012	2015	–	–	x
HND	2013	2015	x	–	–
HND	2014	2015	x	x	–
IRL	2010	2014	x	–	–
JAM	2010	2014	x	–	–
KEN	2010	2015	–	–	x
KEN	2013	2014	–	x	x
KGZ	2010	2014	–	–	x
KOR	2011	2014	x	–	x

It can be seen, that sDACT produces significantly less outlier sequences than DACT and the referred method. While those approaches detect both five anomalous subsequences, sDACT only finds two. This can be explained by the fact, that there are many clusters with only few cluster members. In addition, there are only a few TS, that are very stable over time. This causes, that the mean stability score per cluster is rather low. In order to stand out, a sequence needs therefore a very bad stability score. This only happens in two cases. First, Honduras (HND) does badly from 2014 to 2015, as it moves away from its only cluster member Iceland (ISL) and merges into a large cluster. The second case is Kenya (KEN) from 2013 to 2014, where it turns from noise to a large cluster's member. While the first anomaly sounds reasonable, the second one appears rather groundless, depending on the context. In contrast to DACT, which only found the first and not the second discussed outlier sequence, the referred method had exactly the opposite result. In fact, the only anomaly DACT and the referred method share, is the subsequence of Korea (KOR) from 2011 to 2014. This result is desired, since KOR changes its cluster members at every timestamp in this period.

The outlier sequences IRL and JAM show DACT's sensitivity regarding small clusters merging into large ones. Although those two countries stay stably together from 2010 to 2014, even when merging into the larger cluster, both are detected as outlier sequences. The referred method does not detect those sequences, because it does not penalize merges of clusters. However, although KEN stays with many cluster members over time, it is marked as outlier from 2010 to 2015. This is caused by the split from its cluster in 2012 and 2013. Another outlier detected by the referred method is Guyana (GUY) from 2012 to 2015. In 2013, the data is missing and this is the crucial point. In 2012 GUY is grouped with Hungary (HUN), Italy (ITA) and Iran (IRN). The merge into a larger cluster in 2014 is not penalized, but the following split from HUN, ITA and IRN in 2015 has a very negative effect on the stability, though.

6 Conclusion

In this paper, we introduced two approaches of finding transition-based outliers in time series databases. We examined the differences of the results and evaluated our methods against their competitor from [19], which targets the same problem definition. The results showed that both approaches find reasonable outliers, thus they differ in some characteristics. While the referred method does not penalize merges of clusters but only splits, DACT and sDACT treat both cases the same way. Furthermore, DACT is more sensitive regarding short term changes in small data sets. These differences lead to slightly different results, whereby the methods agree in clear cases. Depending on the application, both approaches provide a benefit.

We are aware of some shortcomings in DACT, that provide incentives for future work. For example, the handling of noise data points from the clustering could be improved. Currently, all subsequences consisting exclusively of noise data points are marked as intuitive outliers. In some cases, this behavior may not be legitimate. Furthermore, DACT is reliant on the assumption, that the underlying clustering is reasonable. Apart from inventing an evaluation measure for over-time clusterings [11,20] in order to support the user in finding the right parameter settings, a new clustering algorithm tailored to the intention of an over-time clustering with temporal linkage would be useful.

References

1. Global economy, world economy. https://www.theglobaleconomy.com/
2. Ahmar, A.S., et al.: Modeling data containing outliers using ARIMA additive outlier (ARIMA-AO). J. Phy.: Conf. Ser. **954**, 012010 (2018)
3. Chambon, S., Thorey, V., Arnal, P.J., Mignot, E., Gramfort, A.: A deep learning architecture to detect events in EEG signals during sleep. In: 28th International Workshop on Machine Learning for Signal Processing, pp. 1–6 (2018)
4. Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I.: STL: a seasonal-trend decomposition procedure based on loess (with discussion). J. Off. Stat. **6**, 3–73 (1990)
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996)
6. Hill, D.J., Minsker, B.S.: Anomaly detection in streaming environmental sensor data: a data-driven modeling approach. Environ. Model Softw. **25**(9), 1014–1022 (2010)

7. Kawahara, Y., Sugiyama, M.: Change-point detection in time-series data by direct density-ratio estimation. In: Proceedings of the 2009 SIAM International Conference on Data Mining, pp. 389–400. SIAM (2009)
8. Keogh, E., Lin, J., Fu, A.: Hot sax: efficiently finding the most unusual time series subsequence. In: Fifth IEEE International Conference on Data Mining (ICDM 2005), pp. 226–233 (2005)
9. Keogh, E., Lonardi, S., Chiu, B.Y.C.: Finding surprising patterns in a time series database in linear time and space. In: Proceedings of the 8th Int. Conference on Knowledge Discovery and Data Mining, pp. 550–556 (2002)
10. Kieu, T., Yang, B., Jensen, C.S.: Outlier detection for multidimensional time series using deep neural networks. In: 2018 19th IEEE International Conference on Mobile Data Management (MDM), pp. 125–134 (2018)
11. Klassen, G., Tatusch, M., Himmelspach, L., Conrad, S.: Fuzzy clustering stability evaluation of time series. In: Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU (2020)
12. Lin, J., Keogh, E., Ada Fu, Van Herle, H.: Approximations to magic: finding unusual medical time series. In: 18th IEEE Symposium on Computer-Based Medical Systems (CBMS 2005), pp. 329–334 (2005)
13. Liu, S., Yamada, M., Collier, N., Sugiyama, M.: Change-point detection in time-series data by relative density-ratio estimation. Neural Netw. **43**, 72–83 (2013)
14. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297 (1967)
15. Munir, M., Siddiqui, S.A., Chattha, M.A., Dengel, A., Ahmed, S.: FuseAD: unsupervised anomaly detection in streaming sensors data by fusing statistical and deep learning models. Sensors **19**(11), 2451 (2019)
16. Munir, M., Siddiqui, S.A., Dengel, A., Ahmed, S.: DeepAnT: a deep learning approach for unsupervised anomaly detection in time series. IEEE Access **7**, 1991–2005 (2018)
17. Salvador, S., Chan, P.: Toward accurate dynamic time warping in linear time and space. Intell. Data Anal. **11**(5), 561–580 (2007)
18. Sun, P., Chawla, S., Arunasalam, B.: Mining for outliers in sequential databases. In: ICDM, pp. 94–106 (2006)
19. Tatusch, M., Klassen, G., Bravidor, M., Conrad, S.: Show me your friends and i'll tell you who you are finding anomalous time series by conspicuous cluster transitions. In: Data Mining Communications in Computer and Information Science. AusDM 2019, vol. 1127, pp. 91–103 (2019)
20. Tatusch, M., Klassen, G., Bravidor, M., Conrad, S.: How is your team spirit? cluster over-time stability evaluation. In: Machine Learning and Data Mining in Pattern Recognition, MLDM (2020)
21. Zhou, Y., Zou, H., Arghandeh, R., Gu, W., Spanos, C.J.: Non-parametric outliers detection in multiple time series a case study: power grid data analysis. In: AAAI (2018)

Brain CT Image with Motion Artifact Augmentation Based on PGGAN and FBP for Artifact Detection

Che Wang[1,2](✉), Xiaoyu Sun[1,2], Bin Zhang[1,2], Guanjun Lai[1,2], Dan Yu[1,2], and Kang Su[3]

[1] Dalian Neusoft University of Information, Dalian, China
jt_wangche@neusoft.edu.cn
[2] Dalian Neusoft Education Technology Group Co. Limited, Dalian, China
[3] Neusoft Institute Guangdong, Foshan, China

Abstract. Synthesis of computed tomographic images with motion artifact has many applications in assisted medical diagnosis such as artifact detection and removal. However, one of the challenges is on how to synthesize high-resolution images with motion artifact while artifact features and tissues are naturally presented within image. In this paper, we propose a new method to solve the problem by combing filtered back-projection (FBP) and progressive growing of generative adversarial networks (PGGAN), while FBP is for artifact generation and feature extraction and PGGAN is for feature augmentation. Finally, we superimpose artifact features onto artifact-free data, so to obtain a set of pre-demanded and diversified images with all kinds of motion artifacts. We quantitatively evaluate the synthetic images by training models with synthetic data for artifact detection. Our extensive experiments demonstrated that the performance of our proposed method is superior over the state-of-the-art methods.

Keywords: Generative adversarial networks · Medical image synthesis · Motion artifact generation

1 Introduction

Synthesis of computed tomographic (CT) images can increase the amount and diversity of medical data that has small samples, so we can develop more robust machine learning algorithms. Due to the relatively complicated CT imaging process, when the scanned object has slight movement or other factors, some motion artifacts will inevitably appear in the CT imaging process as shown in Fig. 1, which will cause image quality to deteriorate, resulting in misdiagnosis, missed diagnosis, or even unable to judge.

Although several methods in machine learning have been proposed to detect artifact or remove artifacts automatically [2], the training process requires a large number of data samples. We believe that synthesis of CT images can facilitate the training process for robust machine learning to detect or remove artifact under desensitization conditions.

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 370–378, 2020.
https://doi.org/10.1007/978-3-030-65390-3_29

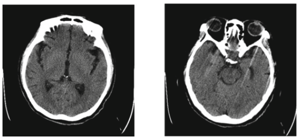


Fig. 1. CT images without or with motion artifact. (left: CT images without motion artifact, right: CT images with motion artifact)

Researchers generally adopt Generative Adversarial Network (GAN)-based Data Augmentation techniques in data synthesis, which can considerably increase the performance [1] since the generated images are completely new samples and can fill the distribution of real images. However, it's hard to synthesize CT images with motion artifact even for high resolution GAN algorithm due to artifact information and tissue structure are mixed presented within an image.

To overcome the challenge above, we propose an approach to solve the problem in synthesis of CT image with motion artifact by the combination of PGGAN and FBP algorithms, while FBP is for feature extraction and PGGAN for feature augmentation, as shown in Fig. 2. Finally, our experiment is based on 6,291 pieces of brain CT data collected by hospital, including 5,723 artifact-free images, 568 images with motion artifact. We evaluate the quality of the synthetic images by using the synthetic images in transfer learning to train an artifact detection model which improves classification accuracy as well as the performance of Area Under Curve (AUC).

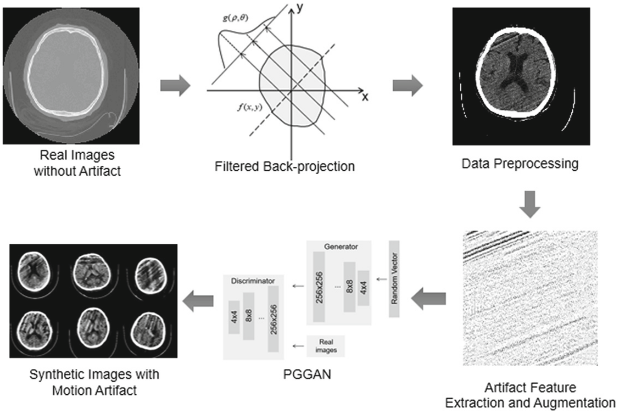


Fig. 2. Overview of our proposed method

2 Related Work

Application of GANs in medical image synthesis helps overcome the privacy issues related to diagnostic medical data and insufficient positive cases of pathology. Most current GAN techniques including DCGAN [5], are unable to generate high resolution images which are important for visualizing lesion with subtle pathological features. Karras *et al.* devised a training scheme for GAN called progressive growing of GANs (PGGAN) [3] that can create photorealistic images at high-resolution. Moreover, image-to-image GANs such as UNIT [6], SimGAN [7], CycleGAN [8] are proposed to obtain images with desired texture and shape. These methods can translate unpaired image-to-image either on corresponding pairs of images or large datasets which are often hard to access for research of medical images.

CT scans of different brain tissues may vary greatly and there is no obvious boundary between tissue and artifact so that we can only obtain some fuzzy artifact-free images when using GAN method mentioned above with a small number of samples. When simulating motion artifact, we consider to use a method of filtered back-projection algorithm [4] (FBP), which simulates the movement of scanned object during CT imaging and projects back to obtain the images with designed motion artifacts. But it can only synthesize a small number of images with motion artifact corresponding to original CT image.

The proposed approach will combine two algorithms together i.e., GAN and FBP described as follows:

Firstly we obtain limited pairs of images (with or without motion artifact) by FBP, extract artifact features by basic image manipulations and augment artifact features by PGGAN. Then we create unlimited number of synthetic images with motion artifacts by combining artifact features and artifact-free image. Finally, we evaluate the quality of the images by using synthetic images in transfer learning to improve the accuracy of artifact detection.

3 Proposed Approach

Without combining artifact features with artifact-free images, we can only obtain some fuzzy artifact-free images even by high-resolution GAN and can only obtain limited number of CT images with motion artifact by FBP, separately. In our approach, in order to augment the dataset, we design a three-step algorithm combining GAN and FBP: (1) obtaining image pairs (with and without artifact), (2) extracting and augment artifact features, (3) synthesizing images with motion artifact using artifact-free data and motion artifact features.

3.1 Corresponding Image Pairs by FBP

Projection of CT through the tissue is a tomographic measurement process in which signals attenuate through the organization, and the received signals reflect the internal structure of tissue. The tomographic projection in a certain direction can be represented by Eq. 1.

$$g(\theta, r) = f(x, y)\delta(x \cos \theta + y \sin \theta - r)dxdy \tag{1}$$

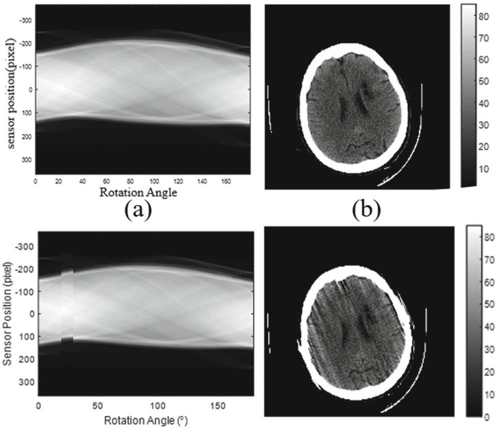
Where $f(x, y)$ is attenuation coefficient of (x, y), δ is projection direction, $g(\theta, r)$ is received signal. Since motion artifact is caused by slight movement of tissue in a certain direction during projection process, then the projection process in the corresponding direction becomes Eq. 2:

$$g'(\theta, r) = f(x + \Delta_i \cdot cos\,\theta, y + \Delta_i \cdot sin\,\theta)\delta(x\,cos\,\theta + y\,sin\,\theta - r)dxdy \qquad (2)$$

Where (θ, r) represents corresponding projection direction, Δ_i is moving distance. According to sinogram, CT image can be reconstructed by back projection. To ensure the reconstruction quality of CT image, the sinogram needs to be filtered and transformed during the back-projection process. The corresponding filtered back-projection reconstruction $X_{fbpMotion}$ is:

$$X_{fbpMotion} = \int \int_{\theta_{\mathrm{i}} - \Delta\theta}^{\theta_{\mathrm{i}} + \Delta\theta} s(\rho) \otimes g'(\theta, r) d\theta dr + \iint_{\theta_{rest}} s(\rho) \otimes g(\theta, r) d\theta dr \qquad (3)$$

Where $(\theta_i - \Delta\theta, \theta_i + \Delta\theta)$ is the scan angle range during motion, θ_{rest} is the remaining angle, $s(\rho)$ is Hamming filter. We can simulate the projection process to obtain corresponding data with motion artifact based on pre-ordered movement direction and distance (as Fig. 3).



(a) (b)

(c) (d)

Fig. 3. Corresponding sinogram and reconstructed CT images: (a) (b) are the spectrogram and artifact-free image, (c) (d) are the spectrogram and image with motion artifact

3.2 Artifact Feature Extraction

The quality of synthesized image is affected by data preprocessing techniques. Hounsfield unit (HU) is a measurement unit to measure the density of human body tissue. We preprocess the CT image into (0HU, 85HU), in order to make motion artifact features to be revealed obviously, so we can extract features successfully. Medical images before and after pre-processing are shown in Fig. 4.

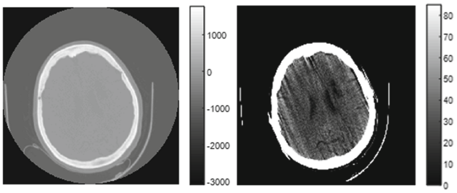


Fig. 4. Comparison between images before (left) and after (right) preprocessing

After data preprocessing, artifact feature can be extracted based on pairs by pixel difference and image cropping:

$$\Delta X_{ArtifactFeature} = F_{imageCrop}\left(P_{pre}\left(X_{fbpMotion}\right) - P_{pre}(X)\right) \tag{4}$$

Where $\Delta X_{ArtifactFeature}$ is motion artifact feature extracted, $X_{fbpMotion}$ is image with motion artifact simulated by FBP, X is original artifact-free CT image, $F_{imageCrop}$ is cutting off the irregular edges of the obtained artifact feature, leaving only the artifact stripe feature, P_{pre} is the image preprocessing. Compared with artifact-free images, there are some pixels lighter and some pixels darker in images with artifact. We can extract an artifact feature matrix by the difference between two images and decompose the matrix into bright and dark stripes as artifact feature (Fig. 5). And we can synthesize artifact features using PGGAN to obtain unlimited and diversified artifact features.

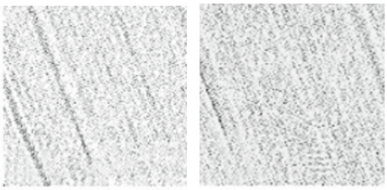


Fig. 5. Artifact feature extracted (left: bright stripes, right: dark stripes)

3.3 Superimpose Artifact Feature on CT Image

The lightness and darkness of artifact features are defined here as Artifact Coefficient including light artifact coefficient and dark artifact coefficient. We superimpose artifact features on artifact-free CT images by multiplying light and dark stripe feature by artifact coefficient:

$$X_{withArtifact} = F_{mask}\left(X_{withouArtifact} + \alpha_{light} * \Delta X_{AFL} + \alpha_{dark} * \Delta X_{AFD}\right) \tag{5}$$

Where ΔX_{AFL} is light artifact feature, ΔX_{AFD} is dark artifact feature, α_{light} is artifact light coefficient, α_{dark} is artifact dark coefficient, F_{mask} will turn the background and the bone area to the same pixel as the artifact-free image due to there is no motion artifact on bone and background, $X_{withArtifact}$ is the image with motion artifact generated by our approach. We can obtain a variety of synthetic data with diverse motion artifact as Fig. 6.

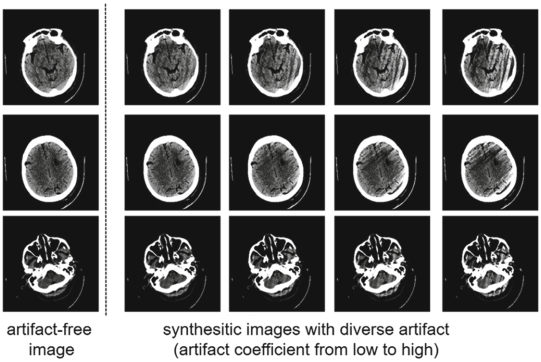

artifact-free image
synthesitic images with diverse artifact (artifact coefficient from low to high)

Fig. 6. Synthesis of CT images with motion artifact by adjusting artifact coefficient

Details of the approach is described in Algorithm 1.

Algorithm 1 CT images with motion artifact synthesis procedure.

```
Input: CT artifact-free images $X = x_1, x_2, \ldots, x_n$ (f(x, y) is distribution of $x_i$) , scan angle range during moving $\vartheta = \vartheta_1, \vartheta_2, \ldots, \vartheta_n(\vartheta_i = (\theta_i - \Delta\theta, \theta_i + \Delta\theta))$, moving distance $\Delta = \Delta_1, \Delta_2, \ldots, \Delta_n$, artifact light coefficient $\alpha_{light}$, artifact dark coefficient $\alpha_{dark}$
Output: Simulated CT image data with motion artifact $Y = y_1, y_2, \ldots, y_n$
Step 1: Calculate tomographic projection in all directions.
    if $\theta \epsilon (\theta_i - \Delta\theta, \theta_i + \Delta\theta)$:
        $g'(\theta, r) = f(x + \Delta_i \cdot cos\theta, y + \Delta_i \cdot sin\theta)\delta(x \cos\theta + y \sin\theta - r)dxdy$
    else:
        $g(\theta, r) = f(x, y)\delta(x \cos\theta + y \sin\theta - r)dxdy$
Step 2: Integrate projection in all directions and obtain corresponding image with motion artifact:
    $X_{fbpMotion} = \iint_{\theta_i - \Delta\theta}^{\theta_i + \Delta\theta} s(\rho) \otimes g'(\theta, r)\, d\theta dr + \iint_{\theta_{rest}} s(\rho) \otimes g(\theta, r)\, d\theta dr$
Step 3: Preprocess image pairs and extract artifact feature:
    $\Delta X_{ArtifactFeature} = F_{imageCrop}\left(P_{pre}(X_{fbpMotion}) - P_{pre}(X)\right)$
    $\Delta X_{AFL} = \|\Delta X_{ArtifactFeature}\|$
    $\Delta X_{AFD} = \|-\Delta X_{ArtifactFeature}\|$
Step 4: Adjust artifact coefficient and superimpose features on artifact-free images:
    $y_i = F_{mask}(X_{withouArtifact} + \alpha_{light} * \Delta X_{AFL} + \alpha_{dark} * \Delta X_{AFD})$
```

4 Experiments

In this section we discuss a set of experiments that we conducted to evaluate the quality of our results. Data used in experiments are from a total of 6,291 pieces of brain CT images collected by a private hospital, including 5,723 artifact-free images and 568 images with artifact. Figure 7 demonstrates comparisons between real images and synthetic images created by the state-of-the-art algorithm PGGAN, and by our proposed solution.

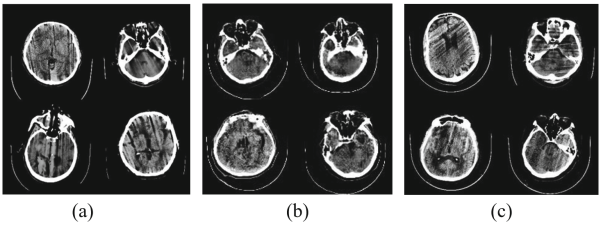


Fig. 7. Comparison between synthetic images with motion artifact and real data: (a) real images, (b) synthetic images by PGGAN, (c) synthetic images by our solution

To further verify the quality of synthetic images with artifact, we use synthetic data to train the classifier to detect artifact by transfer learning. There are 6,291 real images, including 5,723 images without artifact and 568 data with motion artifact. We use ResNet50 as a classifier for artifact detection by strategies shown in Table 1.

Table 1. Classifier training strategies.

Model name	Parent training set (artifact-free)	Parent training set (data with artifact)	Child training set	Test set
Model1	90% real data	90% real data	–	10% real data
Model2	ImageNet	ImageNet	90% real data	10% real data
Model3	90% real data	5,000 synthetic data (FBP)	90% real data	10% real data
Model4	90% real data	5,000 synthetic data (our solution)	90% real data	10% real data

We train the model by transfer learning except model 1. We train a parent model using parent training set, fine-tune with child a training set, and test with test set. Results of models above shown in Table 2, classification accuracy and performance of area under curve (AUC) of artifact detection model obtained by transfer learning have been significantly improved. AUC of model 4 increased from the original 86.18% to 97.13% with improvement 10.95%, accuracy from 90.72% to 95.48% with improvement 4.76%. Therefore, we can obtain an artifact detection model with higher accuracy and AUC by using data synthesized by our solution.

Table 2. Results in transfer learning.

Model	Sensitivity (%)	AUC (%)	Specificity (%)	Accuracy (%)
MODEL1	78.44	86.18	77.03	90.72
MODEL2	71.01	96.31	76.56	94.54
MODEL3	89.34	96.57	74.41	91.46
MODEL4	72.46	97.13	83.33	95.48

5 Conclusion

This paper proposes a data augmentation algorithm to solve small sample limitations when using deep learning to detect artifact in CT images. We proposed a solution using FBP and feature extraction algorithm combined with PGGAN to augment the amount and diversity of images with motion artifact.

Our contributions are as follows:

- **Integration of FBP and PGGAN.** To solve the challenge that high-resolution GAN algorithms cannot synthesize CT artifact images with motion artifact, we simulate motion artifact by FBP, extract and augment features by PGGAN and finally obtain a pre-ordered number of images with motion artifact.
- **Artifact Coefficient.** We propose an Artifact Coefficient as a control parameter to specify the lightness and darkness of motion artifact features. We can adjust the brightness of the artifact by using Artifact Coefficient.
- **Extract motion artifact feature by preprocessing and basic manipulations.** We can extract motion artifact successfully by preprocessing data and pixel difference to further augment the features.

Finally, synthetic data by our solution performs better than PGGAN and can be used in transfer learning, so to have improved the accuracy and AUC of artifact detection. Our extensive experiments demonstrated the superior performance of our proposed approach to the state-of-the-art methods.

References

1. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M.: Generative adversarial networks. Adv. Neural. Inf. Process. Syst. **3**, 2672–2680 (2014)
2. Elss, T., Nickisch, H., Wissel, T.: Deep-learning-based CT motion artifact recognition in coronary arteries. In: Image Processing (2018)
3. Karras, T., Aila, T., Laine, S.: Progressive growing of GANs for improved quality, stability, and variation (2017)
4. Cherry, S.R., Dahlbom, M.: PET: physics, instrumentation, and scanners, pp. 70–93 (2004)
5. Yi, X., Walia, E., Babyn, P: Generative adversarial network in medical imaging: a review (2018)

6. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: Advances in Neural Information Processing Systems (NIPS), pp. 700–708 (2017)
7. Shrivastava, A., Pfister, T., Tuzel, O.: Learning from simulated and unsupervised images through adversarial training. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2107–2116 (2017)
8. Zhu, J.Y., Park, T., Isola, P.: Unpaired image-to-image translation using cycle-consistent adversarial networks (2017)

Recursive RNN Based Shift Representation Learning for Dynamic User-Item Interaction Prediction

Chengyu Yin, Senzhang Wang(✉), Jinlong Du, and Meiyue Zhang

Nanjing University of Aeronautics and Astronautics, Nanjing, China
{chengyuyin,szwang,kingloon,meiyuezhang}@nuaa.edu.cn

Abstract. Accurately predicting user-item interactions is critically important in many real applications including recommender systems and user behavior analysis in social networks. One limitation of most existing approaches is that they use the sparse user-item interaction relationships directly, but ignore the second order user-user and item-item relationships. Another limitation is that they generally embed users and items into different embedding spaces in a static way, but cannot capture the dynamic and evolving dependency between users and items and embed them into a unified latent space. In this paper, we aim to learn dynamic embedding vector trajectories rather than static embedding vectors for users and items simultaneously. A Recursive RNN based Shift embedding method called RRNN-S is proposed to learn the continuously evolving embeddings of users and items for more accurately predicting their future interactions. Specifically, we first propose to quantize the user-user and item-item relationships from the original user-item interaction graph, which can be used as auxiliary information to enrich the sparse user-item interaction graph. A recursive RNN is proposed to iteratively and mutually learn the dynamic user and item embeddings in the same latent space based on their historical interactions. A shift embedding module is next proposed to predict the future user embedding. To predict the item which a user will interact with, we innovatively output the item embedding instead of the pairwise interaction probability between users and items, which is much more efficient. Through extensive experiments on two real-world datasets, we demonstrate that RRNN-S achieves superior performance by comparison with several state-of-the-art baseline models.

Keywords: Recursive RNN · User-item interaction · Shift embedding · GCN

1 Introduction

Online user behavior analysis has been widely studied recently due to the rising popularity of various online platforms such as social networks (e.g., a user answering a question in Quora) [1], multimedia websites (a user watching a video or listening to a song in Lasm.fm) [2], and e-commerce platforms (a customer

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 379–394, 2020.
https://doi.org/10.1007/978-3-030-65390-3_30

purchasing a commodity in Amazon) [3,4]. Especially, accurately predicting the interactions between users and items is critically important and can facilitate for many online applications such as recommendation [5–8], user behavior analysis [3,4,9] and sales forecasting [10,11]. Figure 1(a) shows an example of user-item interaction on an e-commerce platform. Each arrow represents an interaction from a user to an item, e.g. a user buying or browsing a commodity on e-commerce platform Taobao, with each interaction associated with a time stamp t and a feature vector f (such as interaction types, user and commodity features). An interaction between a user and an item can be a user clicking, buying or browsing a certain commodity on the online shopping website. Figure 1(b) shows the sequential interactions between two users (Alice and Tom) and the items. For example, Alice first buys a pair of shoes, and shortly she buys a T-shirt. Thus, we can infer that she is more likely to buy an overcoat rather than a chair in the near future.

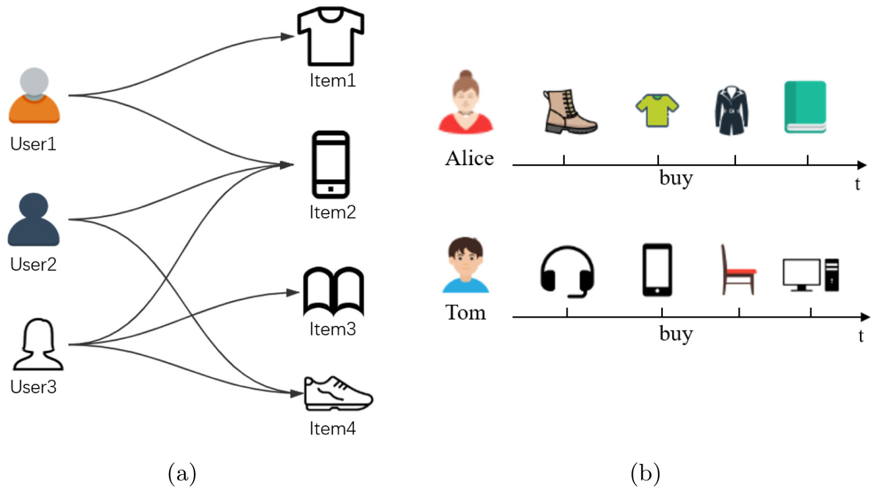


Fig. 1. (a) A toy example of an interaction network containing three users and four items. Each arrow represents an interaction from a user to an item. Each interaction is associated with a timestamp t and a feature vector f (such as the feature of the commodity). (b) Two user interaction sequences: Alice buys a pair of shoes, T-shirt, overcoat and a book; Tom buys a headset, iphone, chair and a computer, successively.

Recommendation systems and temporal network embedding which are relevant to our work have been extensively studied and enjoyed considerable success recently. [5] proposed a cross domain multi-view deep learning approach and utilized social relations among users as auxiliary information. [2] applied graph neural network architecture to knowledge graph (KG), and learned the edge weights in KG. [12] modeled recommendations as a probabilistic mixture over several heterogeneous item-to-item relationships. However, there are three major issues when applying such methods directly to our studied problem. First, most existing works only consider the interaction relationships between users and items, but ignore the user-user and item-item relationships [2,5], which are important auxiliary information to solve the data sparsity issue. Second, the user-item interactions are sequentially dependent as shown in Fig. 1(b). Users' successive online behaviors are usually highly correlated, but are largely ignored by existing collaborative filtering based recommender systems [12,13]. Intuitively, if a user buys a football first, he is more likely to buy a pair of sneakers in the near future. Third, the user's preference and the item popularity evolve over time [14–16]. Existing works mostly learn a static represent vector for each user and item, but fail to capture the dynamic representations of users and items that evolve over time.

To address the above issues, we propose a Recursive RNN based Shift embedding method named RRNN-S to more effectively learn the dynamic representations of users and items, and use them to more accurately predict the future user-item interactions. More specifically, RRNN-S contains a GCN module to learn the user and item representations over the constructed user-user and item-item relationship graphs. The user-user and item-item graphs are constructed based on the second order proximity from the user-item interaction graph. A recursive RNN module is also designed to catch the sequential dependence of user-item interactions simultaneously by mapping users and items into the same latent representation space. The embeddings of users and items are mutually and iteratively updated by the proposed recursive RNN. Then, a shift embedding module is designed to predict the continuous future embedding of a user through the time interval, and then predict the user embedding. Finally, we predict the embedding of the item and identities the item whose embedding vector is closest to it in the embedding space.

To summarize, our primary contributions are as follows:

- To enrich the sparse user-item interaction graph, the user-user and item-item graphs are constructed based on the second order proximity of the user-item graph. Then GCN is applied on the two graphs to learn user and item representations.
- A recursive RNN module is proposed to mutually learn the dynamic embeddings of users and items based on their historical interactions. The proposed recursive RNN can project users and items to the same embedding space.
- A shift embedding module is also designed to predict the evolving user embeddings over time. This module amins to update the future embedding trajectory of the user.
- Comprehensive experiments on two real interaction graph datasets demonstrate the effectiveness of our method against five competitive baselines. Moreover, it leads to 4.2% and 9.2% performance improvement over the state-of-the-art methods on the two datasets, respectively.

The rest of the paper is organized as follows. We first discuss related works in Sect. 2. Then we give some notation and formally define the studied problem in Sect. 3. Section 4 introduces the proposed model RRNN-S and objective function. In Sect. 5, we evaluate our approach and report the results. Finally, we conclude the work in Sect. 6.

2 Related Work

This work is relevant to the research areas of recommender systems and temporal network embedding. Next we will review related works from the two aspects.

2.1 Recommender Systems

Recommender systems (RS), which can facilitate the decision-making process in complex information overload scenarios, play a vital and indispensable role in numerous domains such as multimedia websites, e-commerce, and entertainment. Recently, deep learning has obtained tremendous success in many domains, which causes a surge of interest in applying deep learning techniques to recommendation system.

The recent deep learning based recommendation models can be roughly categorized into RS with neural building blocks (neural building black means MLP, AE, RNN, CNN, etc.) and RS with deep hybrid models (e.g., RNN+CNN, AE+CNN, etc.) [17]. For the first category, [18] designed an end-to-end model to integrates factorization machines and MLP seamlessly, which can severally model the high-order feature interactions and low-order interactions via deep neural networks and factorization machines. [6] designed a CNN-based framework to combine the entity embeddings and work embeddings together for news recommendation. Another deep learning-based recommender models utilize heterogeneous deep learning technique. [7] designed a co-attention-based hashtag recommendation model that integrated both visual and textual information. [19] proposed a model with two corresponding components to encode long-term behavior sequences.

However, these works mostly learn static user and item embeddings, which is not suitable for temporal user-item interaction network. The sequential dependence in temporal user-item interaction, reflects the evolvement of user preference and item popularity over time, which should not be ignored. Our work aims to learn the sequence of dynamic embeddings of users and items.

2.2 Temporal Network Embedding

Recently, temporal network embedding methods are widely studied. For example, [20,21] equated the dynamic evolution of the network to the constant change of the matrix than update the embedding according to the matrix perturbation theory. Similarly, [22] generated the embedding based on the non-negative matrix factorization of a series of time-evolving adjacency matrices with the smoothness. However, these algorithms learn embeddings from a sequence of graph snapshots, which is not applicable to our setting of the successive interaction data. Some other works are based on skip-gram model. For example, [23] proposed to generate temporal random walk-based embedding methods to embed the continuous-time dynamic network in a unique representation, but it generated one final static embedding of the nodes. [24] studied on the neighborhood formation sequence through Hawkes process to capture the influence of historical neighbors on the current neighbors. Whereas the used temporal network is different from our temporal user-item interaction network. [25] introduced triad as the basic mechanism of dynamic changes of the network, to capture the temporal of the network and learned the node representation at different time. It cannot be applied in our study either as triad does not exist in the studied user-item interaction network.

3 Preliminary and Problem Definition

In a typical interaction scenario, $\boldsymbol{u_t} \in \mathbb{R}^n\ \forall \boldsymbol{u} \in \mathcal{U}$ denotes the user embedding and $\boldsymbol{i_t} \in \mathbb{R}^n\ \forall \boldsymbol{i} \in \mathcal{I}$ denotes the item embedding. $\mathcal{U}$ and $\mathcal{I}$ denote the sets of users and items, respectively. An ordered sequence of user-item interaction $\mathcal{S}$ is composed by $S = (u, i, t, f) \in \mathcal{S}$, where u and i denote a user and an item in $\mathcal{U}$ and $\mathcal{I}$, separately. t is the timestamp of interaction S, and each interaction is associated with a feature vector f (e.g. the embedding of user, item or interaction information). Table 1 lists the symbols and their descriptions used in this paper. Based on the above notations, we formally define the studied problem as follows.

Problem Definition: Given a set of historical user-item interactions $\mathcal{S}$, we aim to learn the future embeddings $\boldsymbol{u_{t+\Delta}}$ and $\boldsymbol{i_{t+\Delta}}$ for users and items, and predict that user u will interact with which item at a given future time $t + \Delta$.

Table 1. Symbols and descriptions.

Symbol	Meaning
$\mathcal{S}$	The data set of historial user-item interaction
$\mathcal{U}$ and $\mathcal{I}$	The set of users and items
$\boldsymbol{u}_t$ and $\boldsymbol{i}_t$	The dynamic embedding of user u and item i at time t
$\overline{\boldsymbol{u}}$ and $\overline{\boldsymbol{i}}$	The static embedding of user u and item i
$\boldsymbol{E}_u$ and $\boldsymbol{E}_i$	The static embedding matrices for users and items
$\widehat{\boldsymbol{u}_{t+\Delta}}$	The predicted embedding of user u at time $t + \Delta$
$\widetilde{\boldsymbol{j}_{t+\Delta}}$	The predicted embedding of item j at time $t + \Delta$

4 Recursive RNN Based Shift Representation Learning

In this section, we introduce the proposed model RRNN-S. As shown in Fig. 2, RRNN-S consists of three modules: GCN module over the user-user and item-item relationship graphs to learn their representations, which are constructed from the original user-item interaction graph based on the second order proximity. Recursive RNN module, which aims to mutually learn user and item dynamic embeddings, and projects them into the same embedding space; and shift embedding module, which predicts the future user embedding trajectory.

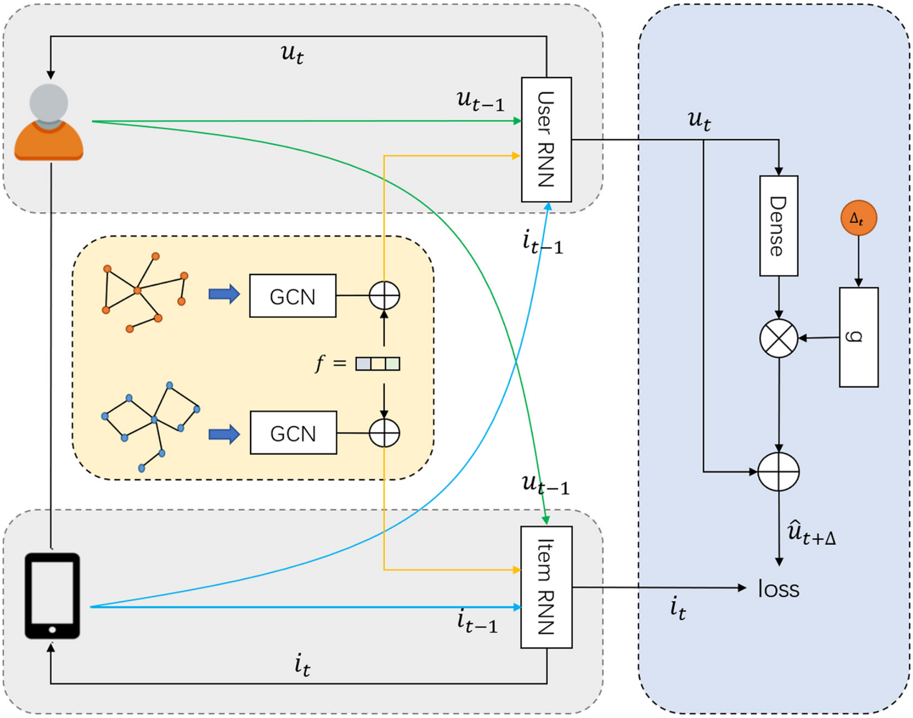


Fig. 2. Overview of the RRNN-S model, which consists of three parts. Two recursive RNNs (in gray box) dynamically update user and item embeddings, respectively; the GCN (in yellow box) captures the auxiliary information from user-item interaction data; and the shift embedding (in blue box) predicts the user embedding at a future time $t + \Delta$. (Color figure online)

4.1 GCN over Second Order Proximity Graph

The data sparsity issue of user-item interactions causes a big dilemma for accurately predicting users' future behavior. To address this issue, we transform the original user-item interaction graph into more robust weighted user-user relation graph and item-item relation graph based on the second order proximity graph. For example, if user1 and user2 have both interacted with the item2, a link between user1 and user2 is constructed. Similarly, if two items (item1 and item3) have been interacted by the same user (user3), they are more likely to be correlated and thus a link between them is constructed as show in Fig. 3. For instance, if user1 and user2 both engage (e.g., buying, browsing) the No. 24 jerseys, we think user1 and user2 are similar (second order proximity), as they are probably fans of NBA superstar Kobe Bryant. In the ordered sequence of temporal user-item interaction, we can generate lots of user-user relationships based on the second-order proximity from different items. In the user-user graph, the edge weight between two connected users are measured by the number of common items that they both interacted with.

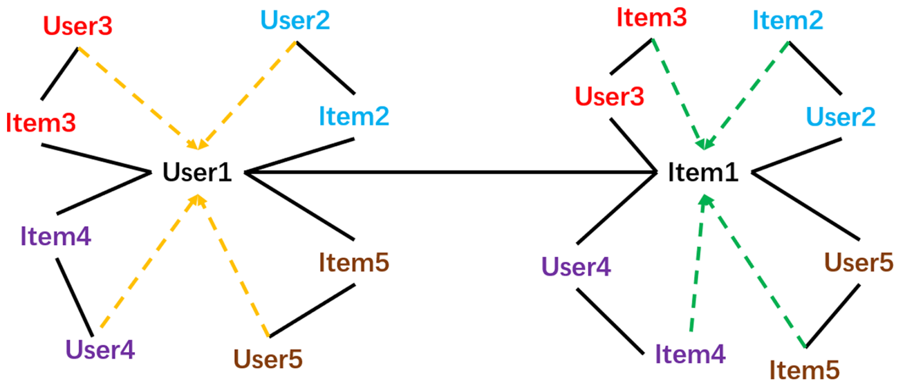


Fig. 3. An illustration of transforming the original user-item interaction graph into the user-user graph and item-item graph based on second order proximity. The black lines represent the interactions between users and items, the yellow lines represent the user-user second-order proximity relations and the green lines represent the item-item second-order proximity relations. (Color figure online)

Given the aforementioned method, user-item interaction sequence $\mathcal{S}$ therefore can be transformed into user-user and item-item second order proximity graph adjacency matrix $A_u \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{U}|}$ and $A_i \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}$. The (i,j)-entry A_u^{ij} is the number of items that user i and user j both interacted with. $|\mathcal{U}|$ and $|\mathcal{I}|$ are the numbers of users and items in $\mathcal{U}$ and $\mathcal{I}$, respectively. We also denote the static feature matrix of user and item as $E_u \in \mathbb{R}^{|\mathcal{U}|\times d_u}$ and $E_i \in \mathbb{R}^{|\mathcal{I}|\times d_i}$, where d_u, d_i are feature dimensions. Then we use feed forward layers to update the static representation matrices by aggregating representations of neighboring entities. Specifically, the layer-wise forward propagation can be expressed as

$$\begin{aligned} H_u^{l+1} &= \sigma(D_u^{-1/2} A_u D_u^{-1/2} H_u^l W_u^l),\ l = 0,1,...,L-1. \\ H_i^{l+1} &= \sigma(D_i^{-1/2} A_i D_i^{-1/2} H_i^l W_i^l),\ l = 0,1,...,L-1. \end{aligned} \tag{1}$$

In Eq. 1, H_u^l and H_i^l are the matrices of hidden representations of users and items in layer l, and $H_u^0 = E_u$, $H_i^0 = E_i$. A_u and A_i are adjacency matrices for user u and item i. D_u and D_i are diagonal degree matrices of A_u and A_i. W_u^l and W_i^l are the layer-specific trainable weight matrices, σ is a non-linear activation function, and L is the number of layers. The final outputs h_u^L and h_u^L concatenate with user-item interaction feature vector f, named o_u and o_i, as the auxiliary feature used in learning dynamic user and item embeddings.

4.2 Recursive RNN

Recursive RNN module, as shown in the gray boxes of Fig. 2, is used to generate user and item dynamic embeddings according to their historical interactions. It consists of two parts: User RNN and Item RNN, which are shared by users and items, to mutually learn dynamic embeddings of users and items. A user/item RNN is composed of RNN layers, and the hidden states of the user/item RNN are used to represent user/item embeddings.

In the recursive RNN module, user and item embeddings will be updated whenever user-item interaction occurs. User RNN updates user embedding u_t by using the user embedding u_{t-1}, item embedding i_{t-1} and auxiliary feature vector o_u at the previous time $t-1$. The item embedding is updated with the item RNN in the similar way to the user RNN. In this way, the interactions between users and items are encoded into both the user and item embeddings. The recursive RNN module maps user and item embeddings into the same latent space. Note that the embeddings of users and items evolve with the dynamic user-item interactions. The proposed recursive RNN module has two advantages. First, users and items are embedded into the same latent space, and thus the similarity between users and items can be easily obtained through measuring their distance in the same embedding space. For example, if two users have similar interactions with similar items, the two users will be closer to each other in the embedding space. Similarly, if two items have the interactions with similar users, we consider the two items are also similar. Based on this idea, the embeddings of users and items can be updated by the following formula iteratively.

$$\begin{aligned} u_t &= \sigma(W_1^u u_{t-1} + W_2^u i_{t-1} + W_3^u o_u) \\ i_t &= \sigma(W_1^i i_{t-1} + W_2^i u_{t-1} + W_3^i o_i) \end{aligned} \tag{2}$$

In Eq. 2, u_t and u_i are dynamic embeddings of user u and item i at time t. o_u and o_i are auxiliary features learned from the GCN module. σ is a sigmoid function. $W_1^u, ..., W_3^u$ are the parameter matrices of user RNN and $W_1^i, ..., W_3^i$ are the parameter matrices of item RNN.

4.3 Shift Embedding

Shift embedding module is designed as a embedding projection operation, which predicts the embedding of the user in the future. The predicted embedding can then be used for downstream tasks, such as link predict or recommendation. Existing works cannot continuously update the user-item interaction embeddings over time [26]. The embeddings are only updated discretely when new interactions occur. We argue that even there is no new interactions between a user and a item, their embeddings still smoothly and continuously change over time because the interests of users and the popularity of items can both evolve over time continuously.

The part in the blue box of Fig. 2 shows the shift embedding module. By considering the elapsed time information, the shift embedding module is able to capture the temporal dynamics of user embedding. If the time gap between two successive interactions of one user is large, the user embedding learned from the previous interaction is not appropriate to predict the current interaction. Therefore, we propose a shift embedding method that can smoothly and continuously adjust the user embeddings between the time interval of two successively interactions with items. Inspired by [27], the feedback loop of RNN keeps the previous information of hidden states as an internal memory. First, we use a linear layer to obtain the internal memory needed to be adjusted u_t^S. Then it is adjusted by the elapsed time $\widetilde{u_t^S}$. Finally, to compose the shifted user embedding, the adjusted internal memory is combined with the original user embedding ($\widehat{u_{t+\Delta}} = u_t + \widetilde{u_t^S}$). Details of the shift embedding module is given below:

$$\begin{aligned} u_t^S &= \sigma(W_s u_t + b) \\ \widetilde{u_t^S} &= u_t^S * g(\Delta) \\ \widehat{u_{t+\Delta}} &= u_t + \widetilde{u_t^S} \end{aligned} \tag{3}$$

where Δ denotes the time interval since last previous user-item interaction and W_s is the parameter matrix of the linear layer, b is bias. The function $g(\Delta) = W_p \cdot log(e + \Delta)$ is used to convert Δ to a time-context vector, and W_p is trainable parameters. $\widehat{u_{t+\Delta}}$ is the predicted user embedding at time $t + \Delta$.

4.4 Overall Objective Function

In our model, we aim to predict the embedding $\widetilde{j_t}$ of the item that a user will interact with. RRNN-S directly outputs the item embedding vector, instead of finding the highest interaction probability in different user-item pairs. Most existing models, which given the highest interaction probability among all user-item pairs, need to do the neural-network forward process for each item, which is very time consuming. In contrast, RNN-S only needs to do forward process once and outputs a predicted item embedding in the shift embedding module, and then identities the item whose embedding vector is closest to it in the embedding space. Thus our model is much more efficient that existing models. The item embedding prediction function is given below:

$$\widetilde{j_{t+\Delta}} = W_1 \widehat{u_{t+\Delta}} + W_2 \overline{u} + W_3 i_{t+\Delta-1} + W_4 \overline{i} + b \tag{4}$$

where $W_1, ..., W_4$ are trainable parameters and b is a bias vector, $\overline{u}$ and $\overline{i}$ represent static embeddings of user and item, respectively. $\widehat{u_{t+\Delta}}$ is the output of shift embedding module, but it is noteworthy that $i_{t+\Delta-1}$ is the dynamic item embedding before $t + \Delta$.

For training the parameters of the model, we minimize the L_2 difference between the predicted item embedding $\widetilde{j_t}$ and the ground truth item embedding j_t at every interaction. We aim to minimize the following loss.

$$minimize \sum_{(u,j,t,f)\in S} \left\| \widetilde{j_t} - j_t \right\|_2 + \lambda_U \left\| u_t - u_{t-1} \right\|_2 + \lambda_I \left\| i_t - i_{t-1} \right\|_2 \tag{5}$$

The first loss term is the error of the predicted embedding vector. To prevent the user and item embedding changes sharply, the last two terms are embedding smoothness regularization, and λ_I and λ_I are scaling parameters.

5 Experiments

In this section, we evaluate our model on two real-world scenarios: Wikipedia edits and JingDong online business. We first introduce the datasets, the baselines and the experiment setup, and then present the experiment results.

5.1 Dataset

We utilize the Wikipedia dataset and the JingDong dataset for evaluation, and the statistics of the two datasets are presented in Table 2.

Table 2. Statistics of the two datasets.

Data	#Users	#Items	#Interactions
Wikipedia	8,227	1,000	157,474
JingDong dataset	10,692	303,150	1,198,735

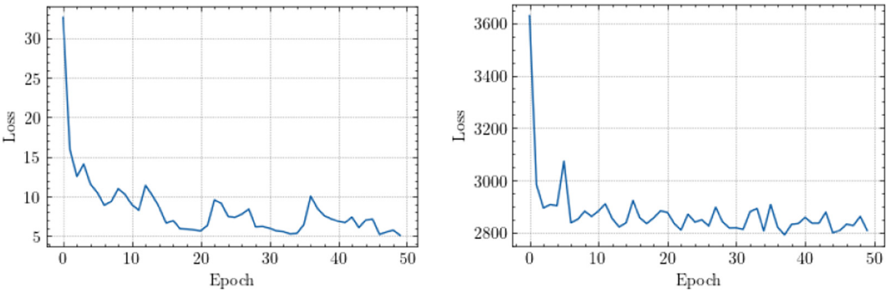
- **Wikipedia editing dataset**: This dataset contains one month of editions made on Wikipedia pages. We select the 1000 pages that get the most editions as items and editors who made at least 5 edits as users. In total we have 8,227 users. There are 157,474 interactions between the selected users and pages in total, and the edited text is considered as features.
- **JingDong dataset**: This dataset is extracted from JD.com, which contains records of users' online behaviors in JiongDong website. It contains 1,198,735 interactions between 10,692 users and 303,150 items from March 2020 to April 2020.

5.2 Baselines and Evaluation Metrice

The compared algorithms in our experiments include:

- **LSTM** [28] is an important ingredient of RNN architectures. Here we simply record the sequence of items, dropping of the time information.
- **Time-LSTM** [28] is a new LSTM variant, which equips LSTM with time gates to model time intervals.
- **Jodie** [26] is a coupled recurrent neural network model to learn dynamic embeddings of users and items. Here we ignore the one-hot embedding for item in Jodie, because it cannot be utilized in a great number of items (e.g., our JingDong dataset has 303,150 items).
- **NGCF** [29] is a recommendation framework based on graph neural network, which explicitly encodes the collaborative signal in user-item bipartite graph by performing embedding propagation.
- **LightGCN** [30] is state-of-the-art collaborative filtering based method. It simplifies the design of GCN to make it more concise and appropriate for recommendation.

We use 80% data for training, 10% validation, and the remaining 10% for testing. We adopt mean reciprocal rank (MRR) and Recall@K defined as follows as the evaluation metric. **MRR** is a statistic measure for evaluating any process that produces a list of possible responses to a sample of queries, ordered by probability of correctness. It is the average of the reciprocal ranks of results for a sample of queries Q: MRR $= \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\text{rank}_i}$, where $rank_i$ refers to the rank position of the first relevant document for the i-th query. **Recall@K** measures the fraction of the total amount of relevant instances that are actually retrieved.



(a) Wikipedia editing dataset

(b) JingDong dataset

Fig. 4. The training loss curve of RRNN-S over the two datasets

The embedding dimension is set to 128, the learning rate is 0.001, and the model is trained with Adam Optimizer with a weight decay of 0.001. The loss curve of the training process in two datasets are shown in Fig. 4, we can see that the model converges quickly. Within around 10 epochs, the training loss first drops quickly, and then becomes stable.

5.3 Results

Table 3 shows the results of our model and baseline models. We observe that RRNN-S significantly outperforms all the baseline on the two datasets.

Table 3. Experiment results

Method	Wikipedia		JingDong	
	MRR	Recall@10	MRR	Recall@10
LSTM	0.332	0.401	0.039	0.057
Time-LSTM	0.351	0.452	0.047	0.064
Jodie	0.741	0.803	0.080	0.131
NGCF	–	0.198	–	0.010
LightGCN	–	0.248	–	0.013
RRNN-1	0.748	0.806	0.083	0.136
RRNN-2	0.751	0.828	0.087	0.143
RRNN-S	**0.756**	**0.837**	**0.089**	**0.142**

Among the baselines, NGCF and LightGCN are collaborative filtering based method, which cannot catch the time information in the temporal user-item interaction network. Thus the two methods are not suitable for dynamic user-item interaction prediction and their performance is the worst among all the methods. Although LSTM does not consider the time interval of two successive interactions with items, it records the sequence of items. Therefore, LSTM performs better than NGCF and LightGCN. Time-LSTM, as a variant of LSTM, considers the time information instead of sequential order, outperforms LSTM by 12.7% in wikipedia and 12.2% in Jingdong dataset. Jodie takes dynamic and sequentially dependent between user-item interaction into consideration, thus it performs better than other baselines. However, compared with Jodie, RRNN-S improves the performance by 4.2% and 9.2% on the two datasets, respectively. This is mainly because it better captures the user-user and item-item relationships.

Ablation Experiment. To illustrate the validity of our proposed module GCN in the second order proximity graph and shift embedding in RRNN-S, we further compare RRNN-S with the following two variants:

- **RRNN-1** drops the GCN module over the item-item and user-user graphs. Only the user-item interaction feature vector f is fed to recursive RNN.
- **RRNN-2** drops the shift embedding module, we directly utilize the output of recursive RNN to predict the final interacted item embedding.

In Table 3, it shows that the RRNN-S outperforms all variants in both Wikipedia and JingDong, which means that our proposed two modules are able to improve performance.

Parameters Sensitivity Analysis. Figure 5 shows the MRR under different dimension of embedding on Jingdong dataset. One can see that the best performance is achieved when the dimension is set to 128. We find that the performance is initially improved with the increase of dimension, because more dimension in embedding can encode more useful information. However, the performance drops when the dimension further increases, as a too large number of dimensions may introduce noise which will mislead the subsequent prediction.

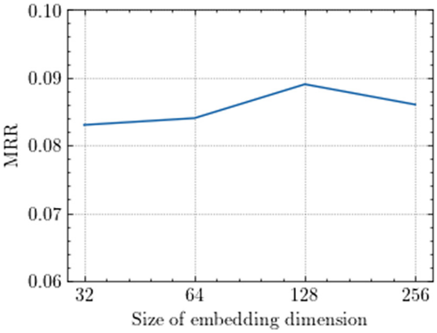


Fig. 5. Robustness to dynamic embedding size.

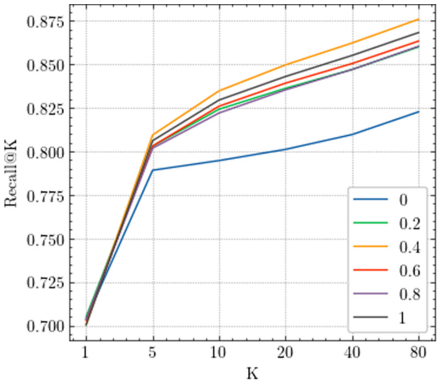


Fig. 6. Recall@K with different λ_I

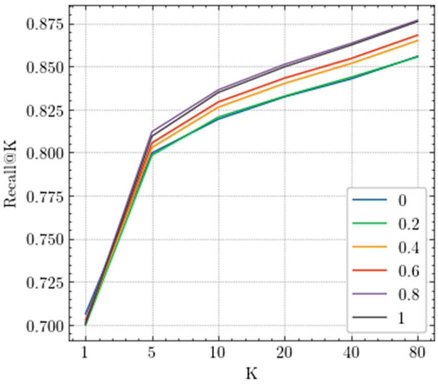


Fig. 7. Recall@K with different λ_U

We next investigate the influence of parameters λ_I on RRNN-S by varying λ_I from 0 to 1, while keeping $\lambda_U = 1$. The results are presented in Fig. 6. We observe that RRNN-S achieves the best performance among all recall@K when $\lambda_U = 0.4$. The model achieves the worst performance when $\lambda_\mathrm{I} = 0$. The large gap between $\lambda_{\mathrm{I}} = 0$ and others shows the effectiveness of embedding smoothness regularization for item embeddings. Then, we further investigate the influence of parameters λ_U, while fixing $\lambda_{\mathrm{I}} = 0.4$. The results are show in Fig. 7. Obviously, when $\lambda_U = 0.8$, RRCNN-S achieves the best performance.

6 Conclusion

In this paper, we propose a recursive RNN-based shift embedding method to predict the item embedding that user will interact with in the future. RRNN-S overcomes the limitations of existing works in exploiting user-item interaction history sequences and the dynamics of entity embedding in temporal interaction network. Over two real-world datasets, our proposal has shown significant performance improvement over user-item interaction prediction. For the future, it would be interesting to further study how to exploiting more auxiliary information and heterogeneous relationships between different entities to boost the performance of temporal interaction network prediction.

Acknowledgments. This work is supported by National Key R&D Program of China 2018YFB1003900, NSF of Jiangsu Province BK20171420, CCF-Tencent Open Research Fund and the Fundamental Research Funds for the Central Universities (No.: NZ2020014).

References

1. Zhao, Y., Min, C., Han, X., Deng, S., Wang, H., Li, J.: Listening to the user's voice: a temporal analysis of autism-related questions on Quora. Proc. Assoc. Inf. Sci. Technol. **56**(1), 513–516 (2019)
2. Wang, H., et al.: Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 968–977 (2019)
3. Guo, L., Hua, L., Jia, R., Zhao, B., Wang, X., Cui, B.: Buying or browsing?: predicting real-time purchasing intent using attention-based deep network with multiple behavior. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1984–1992 (2019)
4. Huang, C., et al.: Online purchase prediction via multi-scale modeling of behavior dynamics. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2613–2622 (2019)
5. Elkahky, A.M., Song, Y., He, X.: A multi-view deep learning approach for cross domain user modeling in recommendation systems. In: Proceedings of the 24th International Conference on World Wide Web, pp. 278–288 (2015)
6. Wang, H., Zhang, F., Xie, X., Guo, M.: DKN: deep knowledge-aware network for news recommendation. In: Proceedings of the 2018 World Wide Web Conference, pp. 1835–1844 (2018)
7. Zhang, Q., Wang, J., Huang, H., Huang, X., Gong, Y.: Hashtag recommendation for multimodal microblog using co-attention network. In: IJCAI, pp. 3420–3426 (2017)

8. Quadrana, M., Karatzoglou, A., Hidasi, B., Cremonesi, P.: Personalizing session-based recommendations with hierarchical recurrent neural networks. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp. 130–137 (2017)
9. Lei, C., Ji, S., Li, Z.: TiSSA: a time slice self-attention approach for modeling sequential user behaviors. In: The World Wide Web Conference, pp. 2964–2970 (2019)
10. Xin, S., et al.: Multi-task based sales predictions for online promotions. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 2823–2831 (2019)
11. Qi, Y., Li, C., Deng, H., Cai, M., Qi, Y., Deng, Y.: A deep neural framework for sales forecasting in e-commerce. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 299–308 (2019)
12. Kang, W.C., Wan, M., McAuley, J.: Recommendation through mixtures of heterogeneous item relationships. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 1143–1152 (2018)
13. Han, X., Shi, C., Wang, S., Philip, S.Y., Song, L.: Aspect-level deep collaborative filtering via heterogeneous information networks. In: IJCAI, pp. 3393–3399 (2018)
14. Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q.Z., Orgun, M.: Sequential recommender systems: challenges, progress and prospects. arXiv preprint arXiv:2001.04830 (2019)
15. Wang, S., Hu, X., Yu, P.S., Li, Z.: MMRate: inferring multi-aspect diffusion networks with multi-pattern cascades. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1246–1255 (2014)
16. Yin, C., Wang, S, Hao, M.: Recursive LSTM with shift embedding for online user-item interaction prediction. In: Proceedings of the IEEE International Conference on Cloud Computer (2020)
17. Zhang, S., Yao, L., Sun, A., Tay, Y.: Deep learning based recommender system: a survey and new perspectives. ACM Comput. Surv. (CSUR) **52**(1), 1–38 (2019)
18. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: DeepFM: a factorization-machine based neural network for CTR prediction. arXiv preprint arXiv:1703.04247 (2017)
19. Lv, F., et al.: SDM: sequential deep matching model for online large-scale recommender system. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 2635–2643 (2019)
20. Li, J., Dani, H., Hu, X., Tang, J., Chang, Y., Liu, H.: Attributed network embedding for learning in a dynamic environment. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 387–396 (2017)
21. Zhu, D., Cui, P., Zhang, Z., Pei, J., Zhu, W.: High-order proximity preserved embedding for dynamic networks. IEEE Trans. Knowl. Data Eng. **30**(11), 2134–2144 (2018)
22. Zhu, L., Guo, D., Yin, J., Ver Steeg, G., Galstyan, A.: Scalable temporal latent space inference for link prediction in dynamic social networks. IEEE Trans. Knowl. Data Eng. **28**(10), 2765–2777 (2016)
23. Nguyen, G.H., Lee, J.B., Rossi, R.A., Ahmed, N.K., Koh, E., Kim, S.: Continuous-time dynamic network embeddings. In: Companion Proceedings of the Web Conference 2018, pp. 969–976 (2018)
24. Zuo, Y., Liu, G., Lin, H., Guo, J., Hu, X., Wu, J.: Embedding temporal network via neighborhood formation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2857–2866 (2018)

25. Zhou, L., Yang, Y., Ren, X., Wu, F., Zhuang, Y.: Dynamic network embedding by modeling triadic closure process. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
26. Kumar, S., Zhang, X., Leskovec, J.: Predicting dynamic embedding trajectory in temporal interaction networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1269–1278 (2019)
27. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. **5**(2), 157–166 (1994)
28. Zhu, Y., et al.: What to do next: modeling user behaviors by time-LSTM. In: IJCAI, vol. 17, pp. 3602–3608 (2017)
29. Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 165–174 (2019)
30. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: LightGCN: simplifying and powering graph convolution network for recommendation. arXiv preprint arXiv:2002.02126 (2020)

Computational Methods for Predicting Autism Spectrum Disorder from Gene Expression Data

Junpeng Zhang[1], Thin Nguyen[2], Buu Truong[3], Lin Liu[3], Jiuyong Li[3], and Thuc Duy Le[3](✉)

[1] School of Engineering, Dali University, Dali, Yunnan, China
zhangjunpeng_411@yahoo.com

[2] Centre for Pattern Recognition and Data Analytics, Deakin University, Geelong, Australia
thin.nguyen@deakin.edu.au

[3] UniSA STEM, University of South Australia, Adelaide, Australia
trumt001@mymail.unisa.edu.au, {lin.liu,jiuyong.li,thuc.le}@unisa.edu.au

Abstract. Autism Spectrum Disorder (ASD) is defined as polygenetic developmental and neurobiological disorders that cover a variety of development delays in social interactions. In recent years, computational methods using gene expression data have been proved to be effective in predicting ASD at the early stage. Feature selection methods directly affect the prediction performance of the ASD prognosis methods. With the advances of computational methods and exploding of high-dimensional ASD gene expression data, there is a need to examine the performance of different computational techniques in predicting ASD. In this paper, we review and conduct a comparison study of 22 different feature selection methods for predicting ASD from gene expression data. The methods are categorised into traditional methods (14 methods) and network-based methods (8 methods). The experimental results have shown that the network-based methods generally outperform the traditional feature selection methods in all three accuracy measures, including AUC (area under the curve), F1-score, and Matthews Correlation Coefficient.

Keywords: Autism · Feature selection · ASD prognosis · Gene expression

1 Introduction

Autism Spectrum Disorder (ASD) is a set of neurodevelopmental disorders characterised by 2 major deficits 1) lack of verbal and non-verbal communication and

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-65390-3_31) contains supplementary material, which is available to authorized users.

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 395–409, 2020.
https://doi.org/10.1007/978-3-030-65390-3_31

social interaction and 2) restricted/repetitive patterns of behaviors, interests and activities [1]. Symptoms of ASD are commonly aware at the second year of life and become clearer at the third year. On the other hand, some less severe presentation of ASD may not be found in children until the age of four to six by their parents or teachers [2].

The pathogenetic mechanism of ASD is not well understood. In response to this question, prior studies have noted the importance of genetic cause [3], neurobiological factors [4], parental age [5], environmental and perinatal factors [6]. However, the general consensus has had a long-term interest in genomic etiology of ASD that may refine the brain development, particularly affect the neural connectivity which influences social communication and provokes restricted interests and repetitive behaviors [7,8].

Investigating genomic abnormalities is also a continuing concern within ASD discovery. ASD is believed to be developed by one of the major "epigenetic theory" which claims that an abnormal gene affects the expression of other genes without changing their DNA sequences [9]. Several studies have found that *GABRB3*, *UBE3A* and *MECP2* genes have consistently indicated epigenetic dysregulation [9–11].

In addition, there is an emerging of transcriptomic studies which have been involved to explore differentially expressed genes and to classify based on these signatures. These experiments identify a number of dysregulated genes in different tissues likely to be associated with ASD pathogenesis, for instance, *IF116* [12], *A2BP1/FOX1* [13] in brain, *SGLT1*, *GLUT2* [14] in intestine and *DRD4* [15], *FOXP1* [16] in blood samples. However, these experiments may neglect the global effect of gene expression regulation. In addition, types of tissue source are disadvantages since using fresh brain tissue is not always possible. Therefore, blood tissue is a common source of currently available ASD data, and it may open some debates regarding precision using peripheral blood cells [16,17].

Traditionally, ASD is diagnosed using combination of clinical symptoms, such as facial expression, and body gestures, and non-specific symptoms. However, those clinical features are not generally applicable to different groups with distinct ages and individual abilities in intelligence. Moreover, some of the core symptoms such as lack of social communication and restricted behaviors are only manifested clearly at the age of 3 years old, and therefore making it difficult for the early intervention to improve the outcomes.

Recently, gene expression data has been proved to be effective in predicting ASD [18,19]. The common strategy of those approaches are to select a set of small genes for diagnosis. With the advances of machine learning techniques in the last few decades, there is a need to examine the performance of different computational techniques in predicting ASD using gene expression data.

In this paper, we review and conduct a comparison study for 22 feature selection methods for predicting ASD. We aim to provide a portrait of various feature selection methods categorised into traditional methods and network-based methods. The results of the comparison study suggest that feature selection methods in the network-based methods are more stable with smaller standard deviation than in the traditional methods.

2 Predicting ASD with Traditional Feature Selection Methods

Feature (gene) selection, as an effective technique for dimensionality reduction of high-throughput gene expression data, is the crucial first step for ASD prediction. In this section, we review the traditional feature selection methods for gene expression data. These traditional feature selection methods evaluate the importance of each gene by using gene expression data as input.

2.1 Area Under the Curve (AUC)

The Area Under the Curve (AUC) model [19] ranks genes by using the AUC value of each gene. The value of AUC is equal to the probability that a classifier will rank a randomly chosen positive sample higher than a randomly chosen negative one. Let A_i is the distribution of scores the classifier produces for samples that are actually in the positive class using a gene g_i, and B_i is the distribution of scores the classifier produces for samples that are actually in the negative class using a gene g_i. The value of AUC for each gene g_i is calculated as follows:

$$AUC(g_i) = P(A_i > B_i) \tag{1}$$

2.2 Chi-Square

Chi-square method [20] uses independence test to evaluate whether the gene is independent of the class label. Given a gene g_i with r different gene expression values, the Chi-square score of the gene is calculated as follows:

$$Chi\_square\_score(g_i) = \sum_{j=1}^{r} \sum_{k=1}^{s} \frac{(n_{jk} - \mu_{jk})^2}{\mu_{jk}} \tag{2}$$

where n_{jk} is the number of samples with the j-th gene expression value in class k. The expected number of samples with the j-th gene expression value in class k is $\mu_{jk} = \frac{n_{*k} n_{j*}}{n}$, where n_{j*} is the number of expression data samples with the j-th gene expression value, n_{*k} represents the number of expression data samples in class k.

2.3 Conditional Mutual Information Maximization (CMIM)

The Conditional Mutual Information Maximization (CMIM) score [21] is a measure that iteratively selects genes which can maximize the mutual information with the class labels given the selected genes. The CMIM score for each new unselected gene g_i can be formulated in the following:

$$CMIM(g_i) = \min_{g_j \in S}[I(g_i; c|g_j)] \tag{3}$$

where S denote the current selected gene set that initially empty, c represents the class labels, and $g_j \in S$ is a specific gene in the current S. $I(g_{i;}c|g_j)$ indicates the conditional mutual information of g_i and c given g_j. It is noted that the value of $I(g_{i;}c|g_j)$ is small if g_i is not closely correlated with the class label c or if g_i is redundant when S is known. The higher the value of $CMIM(g_i)$, the more important the gene g_i is. During the selection process that maximizes the minimum value, it not only guarantees that the selected gene has a strong predictive ability, but reduces redundancy to the selected genes.

2.4 Fisher-Score

Fisher score [22] selects the genes where the gene expression values of samples within the same class are small while the gene expression values of samples from different classes are large. Given a gene g_i, the Fisher score is evaluated as:

$$Fisher - score(g_i) = \frac{\sum\limits_{k=1}^{c} n_k(\mu_{ik} - \mu_i)^2}{\sum\limits_{k=1}^{c} n_k \sigma_{ik}^2} \tag{4}$$

where $n_k, \mu_i, \mu_{i,k}$ and σ_{ik}^2 denote the number of samples in class k, mean value of gene g_i, mean value of gene g_i for samples in class k, and variance value of gene g_i for samples in class k, respectively.

2.5 F-Score

F-score [23] is used to test if a gene is able to separate samples from different classes. The F-score of a gene g_i can be computed as:

$$F - score(g_i) = \frac{\sum\limits_{k=1} \frac{n_k}{c-1}(\mu_k - \mu)^2}{\frac{1}{n-c}\sum_k (n_k - 1)\sigma_k^2} \tag{5}$$

Where $n_k, \mu, \mu_k, \sigma_k$ indicate the number of samples from class k, the mean gene expression value, the mean gene expression value on class k, the standard deviation of gene expression value on class k, respectively.

2.6 Gini-Index

Gini index [24] is used to quantify if a gene can separate samples from different classes. Given a gene g_i with r different gene expression values, for the j-th gene expression value, let W_1 denote the set of samples with the gene expression value no more than the j-th gene expression value, let W_2 denote the set of samples with the gene expression value larger than the j-th gene expression value. If the j-th gene expression value can separate the expression data into W_1 and W_2, the Gini index for the gene g_i is computed in the following:

$$\begin{aligned} Gini\_index\_score(g_i) = \min_{W_i} \Bigg(& p(W_i)(1 - \sum_{k=1}^{s} p(C_k|W_1)^2 \\ & + p(W_2)(1 - \sum_{k=1}^{s} p(C_k|W_2)^2 \Bigg) \end{aligned} \tag{6}$$

where C_k denotes that the class label is k. And $p(W_1)$ and are the probability of the set of W_1 and W_2, respectively. $p(C_k|W_1)$ and $p(C_k|W_2)$ indicate the conditional probability of class k given the set of W_1 and W_2, respectively.

2.7 Interaction Capping (ICAP)

The Interaction Capping (ICAP) [25] is a similar feature selection method as CMIM. Different from CMIM, ICAP restricts the term $I(g_j; g_i) - I(g_j; g_i|c)$ to be nonnegative as:

$$ICAP(g_i) = I(g_i; c) - \sum_{g_j \in S} max[0, I(g_j; g_i) - I(g_j; g_i|c)] \tag{7}$$

where S denote the current selected gene set that initially empty, c represents the class labels, and $g_j \in S$ is a specific gene in the current S. $I(g_i; c)$ is the mutual information between g_i and c, $I(g_j; g_i)$ is the mutual information between g_i and g_j, and $I(g_j; g_i|c)$ is indicates the conditional mutual information of g_j and g_i given c.

2.8 Joint Mutual Informatio (JMI)

The Joint Mutual Information (JMI) [26] is able to increase the complementary information that is shared between new unselected gene and selected genes given the class labels. The JMI score for each new unselected gene g_i can be calculated:

$$JMI(g_i) = \sum_{g_j \in S} I(g_i, g_j; c) \tag{8}$$

where S denote the current selected gene set that initially empty, c represents the class labels, and $g_j \in S$ is a specific gene in the current S.

2.9 LL-L21

The LL-L21 model [27] uses $l_{2,1} - norm$ regularizer to conduct gene feature selection. Assume that $\boldsymbol{X}$ is a gene expression data matrix with n samples and

d genes, and $\boldsymbol{y}$ denotes the label vector which includes s different class labels $\{c_1, c_2, ..., c_s\}$. Firstly, $\boldsymbol{y}$ is transformed into a one-hot label matrix $\boldsymbol{Y}$ where if $\boldsymbol{y}_i = c_j$ then the j-th element in the corresponding row vector $\boldsymbol{Y}(i,:)$ is 1. Suppose that the linear classification problem is parameterized by a weight matrix $\boldsymbol{W}$ where the j-th column of $\boldsymbol{W}$ includes the gene coefficient for the j-th class label, and $\boldsymbol{w}$ denote gene coefficient. y_i is the class label of the i-th sample, and the logistic loss function is defined as follows:

$$loss(\boldsymbol{w}; X, y) = \sum_{i=1}^{n} log(1 + exp(-y_i \boldsymbol{w'x_i})) \tag{9}$$

The LL-L21 model is formulated in the following:

$$\min_{\boldsymbol{W}} ||\boldsymbol{XW} - \boldsymbol{Y}||_F^2 + \alpha ||\boldsymbol{W}||_{2,1} \tag{10}$$

where α is used to balance the contribution of the loss function and the regularization term. By solving the optimization problem, we can obtain a gene coefficient matrix $\boldsymbol{W}$. For the i-th gene g_i, we can calculate the $l_2 - norm$ of each row vector $||\boldsymbol{W}(i,:)||_2$.

2.10 LS-L21

Similar with the LL-L21 model, the LS-L21 model also uses $l_{2,1} - norm$ regularizer to conduct gene feature selection. The difference is that the LS-L21 model uses the least square loss function as follows:

$$loss(\boldsymbol{w}; X, y) = \sum_{i=1}^{n} (y_i - \boldsymbol{w'x}_i)^2 \tag{11}$$

If the least square loss function is specified, the LS-L21 model is formulated:

$$\min_{\boldsymbol{W}} ||\boldsymbol{XW} - \boldsymbol{Y}||_F^2 + \alpha ||\boldsymbol{W}||_{2,1} \tag{12}$$

We can also calculate the $l_2 - norm$ of each row vector $||\boldsymbol{W}(i,:)||_2$ to reflect the importance of the i-th gene g_i.

2.11 ReliefF

ReliefF [28] is a variant method of Relief method [29] for tackling multi-class classification problem. Suppose that l data samples are randomly selected among all n samples, the gene score of g_i in ReliefF is defined in the following:

$$\begin{aligned} ReliefF\_score = \frac{1}{s} \sum_{j=1}^{l} \Bigg(&-\frac{1}{m_j} \sum_{g_r \in SC(j)} d(\boldsymbol{X}(j,i) - \boldsymbol{X}(r,i)) \\ &+ \sum_{k \neq k_j} \frac{1}{h_{jk}} \frac{p(k)}{1 - p(k)} \sum_{g_r \in DC(j,k)} d(\boldsymbol{X}(j,i) - \boldsymbol{X}(r,i)) \Bigg) \end{aligned} \tag{13}$$

where $\boldsymbol{X}$ is gene expression data matrix, s is the number of classes, $SC(j)$ and $DC(j,k)$ denote the nearest data samples to x_j in the same class and a different class k, respectively, and their sizes are h_{jk} and m_j. $p(k)$ is the ratio of samples with class label k.

2.12 Robust Feature Selection (RFS)

Robust Feature Selection (RFS) method [30] employs a joint $l_{2,1} - norm$ minimization on both the loss function and the regularization. To achieve group sparsity, a $l_{2,1} - norm$ regularizer is included in the $l_{2,1} - norm$ loss function. The object function of RFS is formulated as:

$$\min_{\boldsymbol{W}} ||\boldsymbol{XW} - \boldsymbol{C}||_{2,1} + \alpha ||\boldsymbol{W}||_{2,1} \tag{14}$$

where α parameter is used to balance the contribution of the loss function and regularization term, $\boldsymbol{X}$ is a gene expression data matrix with n samples and d genes, $\boldsymbol{C}$ is a one-hot label matrix. By solving the object function of RFS, we can obtain a sparse matrix $\boldsymbol{W}$ where the j-th column contains the gene coefficient for the j-th class label. By selecting some rows, it achieves joint gene selection across different class labels.

2.13 Simons Foundation Autism Research Initiative (SFARI)

Similar with the AUC model [19], the Simons Foundation Autism Research Initiative (SFARI) model ranks ASD genes by using the AUC value of each ASD-related gene. Based on the SFARI database, the SFARI model focuses on the ASD genes for gene selection. Similarly, let A_i is the distribution of scores the classifier produces for samples that are actually in the positive class using an ASD gene g_i, and B_i is the distribution of scores the classifier produces for samples that are actually in the negative class using an ASD gene g_i. The value of AUC for each ASD gene g_i is calculated as follows:

$$AUC(g_i) = P(A_i > B_i) \tag{15}$$

2.14 Trace-Ratio

The trace ratio [31] is used to select global optimal gene subset by computing a trace ratio norm. S_w and S_b are two affinity matrices to describe within-class and between-class expression data similarity. Let $\boldsymbol{W} = [\boldsymbol{w_{i_1}}, \boldsymbol{w_{i_2}}, ..., \boldsymbol{w_{i_k}}] \in^{d \times k}$ be the selection indicator matrix where only the i_j-th entry in $\boldsymbol{w}_{i_j}$ is 1 and all the other entries are 0. The trace ratio of k genes in G is:

$$Trace\_ratio(G) = \frac{tr(\boldsymbol{W'X'L_bXW})}{tr(\boldsymbol{W'X'L_wXW})} \tag{16}$$

where $\boldsymbol{X}$ is a gene expression data matrix with n samples and d genes, $\boldsymbol{L}_w$ and $\boldsymbol{L}_b$ are Laplacian matrices of $\boldsymbol{S}_w$ and $\boldsymbol{S}_b$, respectively.

3 Predicting ASD with Network-Based Feature Selection Methods

The traditional feature selection methods only consider gene expression data as input for ranking genes independently, ignoring the fact that they are interdependent. Actually, some other "prior knowledge" (e.g. gene ontology annotations, protein-protein interactions, and gene co-expression network) can capture the relationships between genes, and help to assess the importance of genes. The network-based feature selection methods use both gene expression data and network as input. If the expression data is specified, the network-based feature selection methods are closely associated with prior biological network. In this section, we focus on reviewing several networks or graphs used in the network-based feature selection methods.

3.1 Co-expression

A gene co-expression network is an undirected graph, where a pair of genes is linked with an edge if there is a similar expression pattern across samples between them. Since co-expressed genes may be controlled by the same transcriptional regulatory program or members of the same pathway or biological process [32], gene co-expression network is of biological interest. Two genes are linked if there is a significant co-expression relationship between them.

3.2 HgncToDO

The Disease Ontology (DO) [33] is a formal ontology of human disease, and is developed to address the need for a purpose-built ontology that covers the full spectrum of disease concepts annotated within biomedical repositories within an ontological framework. Since genes are connected if they share an annotation defined by the DO, we can use DO data to construct hgncToDO based gene-gene interaction network.

3.3 HgncToGObp

The Gene Ontology (GO) [34] defines the universe of concepts associated with gene functions. A GO biological process is achieved by a particular set of molecular functions carried out by specific genes (or macromolecular complexes) [34]. Genes are connected if they share a GO biological process annotation. Therefore, the GO biological process data can be utilized to construct hgncToGObp based gene-gene interaction network.

3.4 HgncToGOcc

A GO cellular component describes a location occupied by a macromolecular machine when it carries out a molecular function, and is the parts of a cell or its extracellular environment [34]. Genes are connected if they have a common GO cellular component annotation. Thus, the GO cellular component data is also able to construct hgncToGOcc based gene-gene interaction network.

3.5 HgncToGOmf

A GO molecular function describes the elemental activities of a gene product at the molecular level, such as binding or catalysis [34]. Genes are connected if they share a GO molecular function annotation. Hence, we can utilize GO molecular function data to construct hgncToGOmf based gene-gene interaction network.

3.6 HgncToHPO

The Human Phenotypes Ontology (HPO) [35] is a formal ontology of human phenotypes, and provides a standardized vocabulary of phenotypic abnormalities encountered in human disease. Genes are connected if they share a HPO annotation. The HPO data can be regarded as 'prior knowledge' to construct hgncToHPO based gene-gene interaction network.

3.7 Protein-Protein Interactions (PPI)

PPIs are the physical interactions between two or more protein molecules. Since PPIs are important "prior knowledge", we can also infer gene-gene interaction network based on protein-protein interaction network. An edge is linked for two genes if their coding proteins interact with each other.

3.8 Pubmed

By using word embedding algorithms which attempt to capture relationships between words [36], gene-gene interaction network can also be automatically estimated from literatures in PubMed (http://www.ncbi.nlm.nih.gov/pubmed). Given a disease (e.g. autism), we can obtain articles associated with the disease. The resulting articles are then transformed into word vectors using the word2vec model [36]. The output word vectors for genes of interest are used to identify gene-gene interactions.

4 A Comparison Case Study in Predicting ASD

We use the largest gene expression datasets on ASD (GEO accession: GSE18123) from Kong et al. [19] for the comparison experiments. Gene expression profiles of the training dataset (P1) were prepared using Affymetrix Human Genome U133 Plus 2.0 Array, and those of the validation dataset (P2) were prepared using Affymetrix Human Gene 1.0 ST Array. In total, we have expression profiles of 17626 genes in 66 ASD and 33 control samples for P1, and 104 ASD and 82 control samples for P2. A list of 1007 ASD genes is obtained from SFARI. We also obtain DO, GO and HPO annotations of genes in expression data. The PPI data is from Human Integrated Protein-Protein Interaction rEference (HIPPIE) [37]. The Partial Least Squares (PLS) [38] is used as the classifier for the classification. We use four classification measures (ACC: Accuracy, AUC:

Area Under the Curve, F1_Score, and MCC: Matthews Correlation Coefficient) to compare the performance of different feature selection methods. To find the best performing feature selection method, we compare the mean ACCs, AUCs, F1_Scores and MCCs between feature selection methods using the top N genes incremented by 10 up to 500. The codes of different methods and all datasets are available at https://github.com/thinng/asd_feat.

5 Results

5.1 Comparison Results of Traditional Feature Selection Methods

We compare the performance of 14 traditional feature selection methods using ACC, AUC, F1-Score and MCC. The classification results of top N genes (incremented by 10 up to 500) for each traditional feature selection method can be seen in Fig. 1. Overall, when we select different numbers of top genes for classification, the performance of each feature selection method is different. Therefore, to have a better performance, it is necessary to select an appropriate number of top genes for classification. In terms of mean ACC, AUC, F1_Score and MCC, the LL-L21 model performs the best in terms of four measures. The CMIM and ICAP have the same performance. The reason is that the two models are similar feature selection methods. In addition, the Fisher-score and F-score models also perform the same in terms of four measures. The reason is that they all consider the within class variance and between class variance to select important genes. The detailed comparison results can be seen in Supplementary file 1.

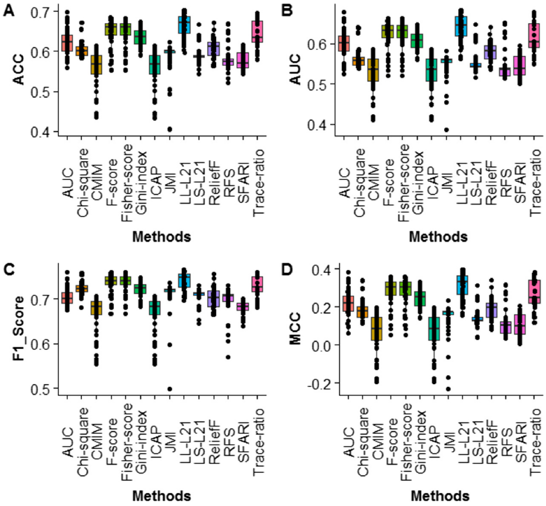


Fig. 1. Comparison results of traditional feature selection methods in terms of ACC, AUC, F1_Score and MCC.

5.2 Comparison Results of Network-Based Feature Selection Methods

In addition to gene expression data, network-based feature selection methods consider network as input. In this work, we use a gene ranking algorithm called GeneRank [39] to rank genes. To balance the contribution of gene expression data and gene-gene interaction network, GeneRank uses a free parameter d with interval [0, 1]. If $d = 0$, the ranking results of genes are based solely on gene expression data. As for $d = 1$, the ranking results of genes are only based on gene-gene interaction network. To find an appropriate parameter d for each network-based feature selection method, we compare the mean ACCs, AUCs F1_Scores and MCCs under different values of d incremented by 0.05 from 0 to 1. As a result, the co_expression_60 ($d = 0.6$), hgncToDO_65 ($d = 0.65$), hgncToGObp_5 ($d = 0.05$), hgncToGOcc_35 ($d = 0.35$), hgncToGOmf_25 ($d = 0.25$), hgncToHPO_100 ($d = 1$), PPI_40 ($d = 0.40$), and pubmed_65 ($d = 0.65$) perform the best for each network-based feature selection method. Similar with the traditional feature selection methods, the network-based feature selection methods perform differently with different number of top genes. In terms of mean ACC, AUC, F1_Score and MCC, the pubmed_65 method performs the best (see Fig. 2). Unlike the traditional feature selection methods, all the network-based feature selection methods have different classification results with each other. The detailed comparison results can be seen in Supplementary file 2.

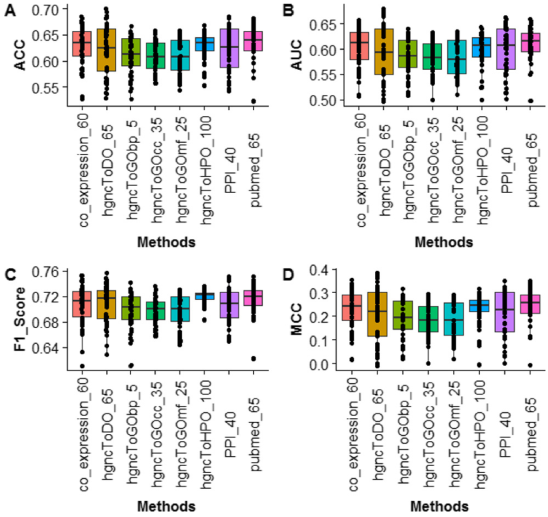


Fig. 2. Comparison results of network-based feature selection methods in terms of ACC, AUC, F1_Score and MCC.

5.3 Comparison Between Traditional and Network-Based Feature Selection Methods

In this section, we compare the performance between traditional and network-based feature selection methods in predicting ASD. As shown in Fig. 3, the feature selection method called LL-L21 in the traditional category performs the best in terms of mean ACC, AUC, F1_Score and MCC. However, as shown in Table 1, comparing with the methods in the traditional category, the feature selection methods in the network-based category are more stable with smaller standard deviation between different methods. Moreover, the average values of mean ACCs, AUCs, F1_Scores and MCCs in the network-based feature selection methods are slightly higher than those of the traditional feature selection methods in predicting ASD. In addition, we compare the number of SFARI genes in the top 500 genes for the two categories. Excepting the SFARI method in the traditional category, the average number of SFARI genes in the traditional category (30.86) is larger than that of the network-based category (26.25). Since the best method LL-L21 only contains 21 SFARI genes, it indicates that more SFARI genes in the feature selected genes may not result in higher performance for feature selection methods. The detailed number of SFARI genes for the two categories can be found in Supplementary file 3.

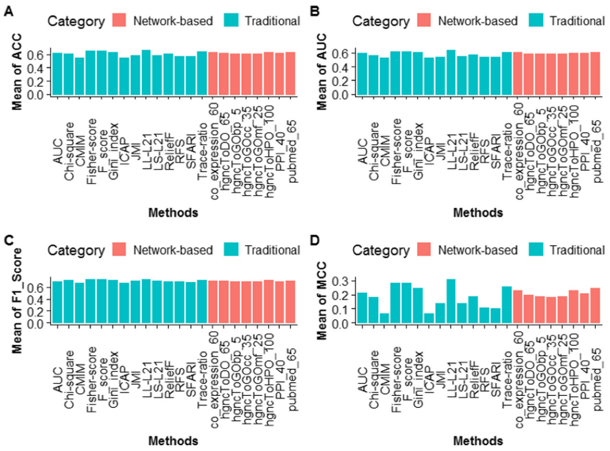


Fig. 3. Comparison results between two categories (traditional and network-based) in terms of mean ACC, AUC, F1_Score and MCC.

Table 1. Comparison results in terms of mean and std of mean ACC, AUC, F1_Score and MCC. mean and std represent the average value and standard deviation of mean ACCs, AUCs, F1_Scores and MCCs, respectively.

Category	ACC (mean, std)	AUC (mean, std)	F1_Score (mean, std)	MCC (mean, std)
Traditional	(0.61, 0.04)	(0.58, 0.04)	(0.71, 0.02)	(0.18, 0.08)
Network-based	(0.62, 0.01)	(0.59, 0.01)	(0.71, 0.01)	(0.21, 0.02)

6 Discussion and Conclusion

In the comparison study, we select top N genes incremented by 10 up to 500 for classification. Although the heuristic way can help us pick the number of selected genes from the top 10 to 500 genes that have the best classification, the optimal number of selected genes is still unknown. Moreover, the whole process of selecting top genes and classification for each feature selection method is computationally expensive. In the future, it is an open challenge to choose the optimal number of genes for classification.

The performance of both traditional and network-based feature selection methods is still room for improvement. For example, the values of ACC, AUC, F1_Score and MCC for the traditional and network-based feature selection methods are only around 0.6, 0.6, 0.7 and 0.2, respectively. The unfavourable result may be caused by the following three aspects. The first aspect is that the built prediction model in the training ASD dataset cannot be scaled to the validation ASD dataset. Thus, choosing a proper prediction model for gene expression data in ASD is preferred. The second aspect is the limited number of samples in ASD gene expression data and would cause an over-fitting problem of classification. For the issue, it is necessary to enlarge the number of samples in both training and validation dataset. The third aspect is the cause of feature selection methods. Since gene expression data is different from other data, it is urgent to develop new feature selection methods appropriate for it.

In conclusion, we provided a review of different feature selection methods and conducted a comparison study for ASD prediction. The paper would serve as a guide and benchmark for novel ASD diagnosis methods.

Funding. This work has been supported by the National Natural Science Foundation of China (61702069, 61963001), the Yunnan Fundamental Research Projects (202001AT070024), the NHMRC Grant (1123042), and the Australian Research Council Discovery Grant (DP170101306).

References

1. American Psychiatric Association et al. Diagnostic and statistical manual of mental disorders (DSM-5®). American Psychiatric Pub (2013)
2. McConachie, H., Le Couteur, A., Honey, E.: Can a diagnosis of asperger syndrome be made in very young children with suspected autism spectrum disorder? J. Autism Dev. Disord. **35**, 167–176 (2005)
3. Sahin, M., Sur, M.: Genes, circuits, and precision therapies for autism and related neurodevelopmental disorders. Science **350**(6263), aab3897 (2015)
4. Lai, M.-C., Lombardo, M.V., Baron-Cohen, S.: Autism. The Lancet **383**(9920), 896–910 (2014)
5. Gabis, L., Raz, R., Kesner-Baruch, Y.: Paternal age in autism spectrum disorders and ADHD. Pediatr. Neurol. **43**(4), 300–302 (2010)
6. Muhle, R.A., Reed, H.E., Stratigos, K.A., Veenstra-VanderWeele, J.: The emerging clinical neuroscience of autism spectrum disorder. JAMA Psychiatry **75**(5), 514 (2018)

7. Baron-Cohen, S.: Two new theories of autism: hyper-systemising and assortative mating. Arch. Dis. Child. **91**, 2–5 (2006)
8. Ecker, C., Bookheimer, S.Y., Murphy, D.G.M.: Neuroimaging in autism spectrum disorder: brain structure and function across the lifespan. Lancet Neurol. **14**(11), 1121–1134 (2015)
9. Hall, L., Kelley, E.: The contribution of epigenetics to understanding genetic factors in autism. Autism **18**(8), 872–881 (2013)
10. Wong, C.C.Y., et al.: Methylomic analysis of monozygotic twins discordant for autism spectrum disorder and related behavioural traits. Mol. Psychiatry **19**(4), 495–503 (2013)
11. Nagarajan, R.P., Hogart, A.R., Gwye, Y., Martin, M.R., LaSalle, J.M.: Reduced MeCP2 expression is frequent in autism frontal cortex and correlates with aberrant MECP2 promoter methylation. Epigenetics **1**, e1–e11 (2006)
12. Garbett, K., et al.: Immune transcriptome alterations in the temporal cortex of subjects with autism. Neurobiol. Dis. **30**(3), 303–311 (2008)
13. Voineagu, I., Eapen, V.: Converging pathways in autism spectrum disorders: interplay between synaptic dysfunction and immune responses. Front. Hum. Neurosci. **7**, 738 (2013)
14. Walker, S.J., Fortunato, J., Gonzalez, L.G., Krigsman, A.: Identification of unique gene expression profile in children with regressive autism spectrum disorder (ASD) and ileocolitis. PLoS One **8**(3), e58058 (2013)
15. Emanuele, E., et al.: Increased dopamine DRD4 receptor mRNA expression in lymphocytes of musicians and autistic individuals: bridging the music-autism connection. Neuro Endocrinol. Lett. **31**, 122–125 (2010)
16. Chien, W.-H., et al.: Increased gene expression of FOXP1 in patients with autism spectrum disorders. Mol. Autism **4**(1), 23 (2013)
17. Zhang, Z., Zhu, Q., Xie, G.-S., Chen, Y., Li, Z., Wang, S.: Discriminative margin-sensitive autoencoder for collective multi-view disease analysis. Neural Netw. **123**, 94–107 (2020)
18. Oh, D.H., Kim, I.B., Kim, S.H., Ahn, D.H.: Predicting autism spectrum disorder using blood-based gene expression signatures and machine learning. Clin. Psychopharmacol. Neurosci. **15**(1), 47–52 (2017)
19. Kong, S.W., et al.: Characteristics and predictive value of blood transcriptome signature in males with autism spectrum disorders. PLoS One **7**(12), e49475 (2012)
20. Liu, H., Setiono, R.: Chi2: Feature selection and discretization of numeric attributes. In: 1995 Proceedings of the Seventh International Conference on Tools with Artificial Intelligence, pp. 388–391. IEEE (1995)
21. Fleuret, F.: Fast binary feature selection with conditional mutual information. J. Mach. Learn. Res. **5**(2004), 1531–1555 (2004)
22. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley, Hoboken (2012)
23. Wright, S.: The interpretation of population structure by f-statistics with special regard to systems of mating. Evolution **19**(3), 395–420 (1965)
24. Gini, C.W.: Variability and mutability, contribution to the study of statistical distributions and relations. Studi cconomico-giuridici della r. Universita de cagliari (1912). Reviewed in: Light, R.J., Margolin, B.H.: An analysis of variance for categorical data. J. Am. Stat. Assoc. **66**, 534–544 (1971)
25. Jakulin, A.: Machine learning based on attribute interactions. Ph.D. thesis, University of Ljubljana (2005)
26. Yang, H.H., Moody, J.: Data visualization and feature selection: new algorithms for nongaussian data. In: Advances in Neural Information Processing Systems, pp. 687–693 (2000)

27. Tang, J., Alelyani, S., Liu, H.: Feature selection for classification: a review. In: Data Classification: Algorithms and Applications, pp. 37–64 (2014)
28. Robnik-Šikonja, M., Kononenko, I.: Theoretical and empirical analysis of ReliefF and RReliefF. Mach. Learn. **53**(1–2), 23–69 (2003)
29. Kira, K., Rendell, L.A.: The feature selection problem: traditional methods and a new algorithm. In: AAAI, vol. 2, pp. 129–134 (1992)
30. Nie, F., Huang, H., Cai, X., Ding, C.H.: Efficient and robust feature selection via joint L2, 1-norms minimization. In: Advances in Neural Information Processing Systems, pp. 1813–1821 (2010)
31. Nie, F., Xiang, S., Jia, Y., Zhang, C., Yan, S.: Trace ratio criterion for feature selection. In: AAAI, vol. 2, pp. 671–676 (2008)
32. Weirauch, M.T.: Gene coexpression networks for the analysis of DNA microarray data. Appl. Stat. Netw. Biol.: Methods Syst. Biol. **1**, 215–250 (2011)
33. Bello, S.M., et al.: Disease ontology: improving and unifying disease annotations across species. Dis. Models Mech. **11**(3), dmm032839 (2018)
34. Ashburner, M., et al.: Gene ontology: tool for the unification of biology. Nat. Genet. **25**(1), 25 (2000)
35. Köhler, S., et al.: The human phenotype ontology in 2017. Nucleic Acids Res. **45**(D1), D865–D876 (2016)
36. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
37. Alanis-Lobato, G., Andrade-Navarro, M.A., Schaefer, M.H.: HIPPIE v2.0: enhancing meaningfulness and reliability of protein-protein interaction networks. Nucleic Acids Res. **45**(D1), D408–D414 (2016)
38. Wold, S., Sjöström, M., Eriksson, L.: PLS-regression: a basic tool of chemometrics. Chemometr. Intell. Lab. Syst. **58**(2), 109–130 (2001)
39. Morrison, J.L., Breitling, R., Higham, D.J., Gilbert, D.R.: GeneRank: using search engine technology for the analysis of microarray experiments. BMC Bioinf. **6**(1), 233 (2005)

Recommender Systems

Declarative User-Item Profiling Based Context-Aware Recommendation

Rosni Lumbantoruan(✉), Xiangmin Zhou, and Yongli Ren

School of Science, RMIT University, Melbourne, Australia
{rosni.lumbantoruan,xiangmin.zhou,yongli.ren}@rmit.edu.au

Abstract. Context-aware recommendation has attracted much attention due to its ability to effectively finding the items that a target likes out of an abundance of online items. Different users may characterize different contexts to items since they also consider different contexts when they select items. Comprehensive identification of the declarative dominant contexts for both items and users can significantly affect the quality of the recommendation, which is often overlooked by the existing research. In this paper, we propose a new recommendation approach, which identifies the dominant contexts as declared by users on their previous transactions. Firstly, we identify the significant contexts from both item and user perspectives and construct the user-item profile in a personalized manner. Secondly, we propose a new context-aware recommendation model that seamlessly incorporates both declarative profiles into the recommendations. Finally, we demonstrate the effectiveness of the proposed method by conducting comprehensive experiments over two real benchmark datasets. The experimental results show that the proposed method outperforms the state-of-the-art methods.

Keywords: Context-aware recommender · User and item profiling · Dominant contexts

1 Introduction

The rapid development of Web 2.0 and smart mobile devices have resulted in prolific numbers of online activities. It ranges from online shopping, entertainment, education, e-government, and so on. In April 2020, active Internet users reach almost 4.57 billion people, encompassing 59% of the global population[1]. The growth of online users is linear with the abundance of online items offered for them to choose. A key issue of a recommender system in facing the abundance of users, online items, and user-generated contents, is to help the users to find the items that match their preferences as much as possible. Many internet sites, such as Amazon.com, YouTube, Netflix, Spotify LinkedIn, Facebook, Tripadvisor and Yelp offer practical recommender systems for this issue and incorporate contextual information to better understand user's preferences. In

[1] https://www.statista.com/statistics/617136/digital-population-worldwide/.

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 413–427, 2020.
https://doi.org/10.1007/978-3-030-65390-3_32

practice, different users concern different aspects or contexts when selecting an online item. Similarly, item contexts might also be differently characterized by each of the users. For example, Joe who likes to exercise might see a game console as another way of working out at home by imitating the character's movement on the game. Meanwhile, Lian who likes to read, see it purely as an entertainment and fun item. Given the above scenario, identifying the item's contexts enables the recommendation of the game console to users whose context is either working out or entertainment. Ignoring the item profiling might hinder the capability of the context-aware recommendation to return a more accurate item prediction. It becomes worthwhile incorporating item profiling apart from user profiling in recommendation for higher system performance.

In this paper, we study declarative user-item profiling based context-aware recommendation for e-business application. Given a user u and an item set I, declarative user-item based recommendation aims to capture the dominant contexts of user u and item i and return a list of items with the best match between u and i profile. As formulated above, first we need to identify the dominant contexts for both user and item. Second, by recalling the huge number of users and items, we treat user and item with a similar profile in the same group. Finally, we design an item recommendation that incorporates both user and item profile into the recommendation to increase the recommendation quality.

Incorporating context as in context-aware recommender system has been shown to be an effective approach in recommender system. There has been considerable research in this area [1,8,9,19]. Baltrunas et al. treat the time and location of an event as contexts and generate recommendation to users based on these contexts sub-profiles. [1]. In [19], Zhou et al. identify the optimal features by removing the redundancy from a feature to a feature set correlation. Lumbantoruan et al. in [9] capture the current contexts of users through users' feedback, while, in [8], the personalized contexts for each user are identified and the recommendation is generated based on the most relevant contexts to the target user. Although the aforementioned works have proved their superiority in CARS, none of them considers the complete declarative profiling of users and items to improve the recommendation quality. Motivated by the limitation of the current approaches, we propose a Declarative User-Item Profiling Based Context-Aware Recommender System (DecPro-CARS) that declaratively identifies user and item profile and integrates them to improve the effectiveness of the item recommendation. Specifically, we first identify user and item profile based on their review text by enhancing the functionality of UW-NMF topic modeling. Then, we ensure the model to handle the profiles with poor contextual information by collaborating the group of users and items with similar profiles. Later, we propose an item recommendation that seamlessly incorporates these grouping into the item recommendation. In summary, the key contributions of this work are as follows:

1. We propose a new framework that exploits the personalized and dominant contexts of user and item for the item recommendation. While the declarative

user and item contexts respectively represent the profile of user and item, ignoring the less important contexts avoid unnecessary computation cost.

2. We learn the personalized dominant contexts not only for the user but also the dominant contexts of items, based on which the declarative item profile is constructed. The declarative item profile will help the target users to choose items based on his dominant contexts.
3. We propose a new context-aware item recommendation that incorporates the user and item profile to the item recommendation. This new context-aware item recommendation will also perform on those profiles with few contextual information by incorporating group of users and items based on the similarity of profiles.
4. We conduct extensive experiments of the proposed solution on two public datasets to verify the effectiveness and efficiency performance of the proposed solution.

The rest of the paper is organized as follows: Sect. 2 reviews the related work on context-aware recommender system. Section 3 presents our proposed context-aware recommender with its declarative user and item profiling. Extensive experimental results are reported in Sect. 4. Finally, Sect. 5 concludes the whole paper.

2 Related Works

We review two related research that closely related to our work, including content-based filtering and context-aware recommender system.

Content-Based Filtering: Many approaches have been proposed for item recommendation. Among them, collaborative filtering-based recommendation is the most popular strategy adopted in existing recommender systems. Traditional collaborative filtering approaches exploit the user-item interaction to recommend items to a target user based on his friends' highly-rated items [4,6,7,11]. However, this rating-based collaborative filtering can not work well because of the sparsity of ratings in online communities. This propelled researchers towards content-based filtering methods for recommendation by incorporating contexts to enhance the quality of recommender systems. Content-based filtering techniques typically learn user and item profiles from item descriptions or user reviews [1,13]. In [1], Baltrunas et al. incorporated time and location of the event as contexts and create sub-profiles based on these contexts and perform KNN to recommend items for each of the profiles. In [13], Said et al. generate item prediction for a user by grouping similar users and create sub-profiles for each group. However, none of these approaches exploits significant contexts for each user that lies in user text reviews.

Context-Aware Recommender System: Context-aware recommender system appears to be a promising solution to alienate the important contexts between users. Content and contextual information have been embedded in various recommender systems. Typical examples include predefined and static attribute sets [2], dynamic context [12,15,17,18], and dynamic contexts with importance difference [8,19]. Baltrunas et al. [2] proposed a context-aware matrix factorization (CAMF), which introduced three models that represent the degrees of context and rating interaction. CAMF can outperform Tensor Factorization when the influence of the context is light. It also proves that grouping the items per category has a beneficial effect comparing with other CAMF alternatives. In [12], Ren et al. use location, queries, and web content for making a contextual recommendation. Zhang et al. [15] identifies features that a user concerns and recommends a product that performs well on those features. Zhou et al. [17] fuse the content and social relevance to identify the relevant video over online sharing communities. Meanwhile, Zhou et al. [18] exploit the contextual information of social users to enhance the video recommendation to multiple users. In [20], Zhou et al. embed the temporal user interest prediction and entity matching for the streaming item recommendation. These approaches pre-set the contexts used for the final recommendation, which ignores personalization of contexts, thus the contexts used may not be optimal for the real-world situations.

Recent approaches proposed to support the different importance of contextual information [8,9,19]. In [9], Lumbantoruan et al., learn the different context's priority and update them through interaction. Zhou et al. [19] identify the optimal features based on the correlation between a feature to a feature set to remove the feature redundancy. An approximately optimal feature set is identified for the content-context interaction graph-based social recommendation. Yet, they select the same set of contexts for all target users when conducting recommendation, thus the user preference with respect to contexts can not be captured. This problem has been considered in [8] which declaratively personalized contexts for each user by using UW-NMF topic modeling, but it only works best for top 30% of the active users and not considering users with less contextual information. In this work, we aim to solve the problems caused by the context preference bias of different target users for effectively social item recommendation. Work on top of UW-NMF topic modeling in [8], we identify the dominant contexts of both users and items which represent user's and item's profile. Later we seamlessly combine the declarative user and item profiling to the new proposed Declarative User-Item Profiling Context-Aware Recommender System (DecPro-CARS).

3 Framework of DecPro-CARS

In this section, we present a new context-aware recommender system with declarative user and item profiling (DecPro-CARS), and the overview is shown in Fig. 1. Specifically, the input of our framework is user-item metadata, user's historical transaction including the user review text that incorporates the rating, namely in the format of which user u rates which item i with review text d_i. DecPro-CARS aims to reveal the *user and item (hidden) contexts* accordingly from user u's reviews for a set of items and item i's reviews by a set of users. Given both user and item declarative profile, the system will incorporate these

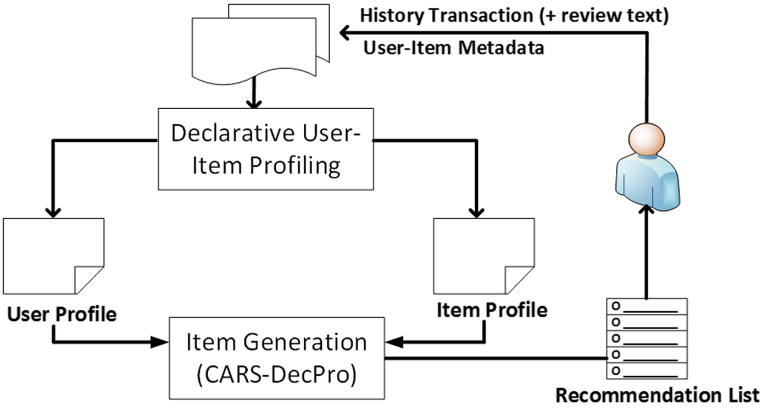

History Transaction (+ review text)
User-Item Metadata
Declarative User-Item Profiling
User Profile
Item Profile
Item Generation (CARS-DecPro)
Recommendation List

Fig. 1. Framework of our user-item declarative profile recommendation

profiles to the recommendation to generate the item that will most likely match the user's preference. Finally, DecPro-CARS returns the most relevant recommendation to the target user. There are two components in DecPro-CARS:

- **Declarative User and Item Profiling** identifies the most dominant contexts for both user and item from previous user transaction, especially from user text reviews. We assume that user will only write the important contexts in their review about an item, and so the item will also be reviewed by their dominant contexts. The declarative profiling for both user and item considers the different contexts for each user and each item.
- **Item Recommendation** generates the item recommendation by seamlessly incorporate the declarative profile of both user and item. The information of user and item with the closest profile is used to improve the recommendation effectiveness in handling both user and item with poor contextual information.

The notation used in this paper is listed in Table 1 for easy reference.

3.1 Declarative User and Item Profiling

It is a fact that when selecting online items, each user puts different contexts into consideration, vary from personal user contexts or the dominant contexts that the item has. Lumbantoruan et al. [8] propose UW-NMF to declaratively identify user profile for user u on item i based on their review text, transaction history, and the corresponding user and item content as below.

$$p_{ui} = [\underbrace{\tilde{w}_{u1}, \cdots, \tilde{w}_{ul}}_{\text{user context}}, \underbrace{c_1^U, \cdots, c_q^U}_{\text{user content}}, \underbrace{c_1^I, \cdots, c_j^I}_{\text{item content}}] \qquad (1)$$

User contexts are denoted as w_u, meanwhile, user's and item's contents, extracted from metadata, denoted as C^U and C^I accordingly. It is also a fact

Table 1. Notation

Symbol	Definition and description	Symbol	Definition and description
U	Set of m users: $U = \{u_1, \cdots, u_m\}$	$\mathbf{W}$	$review \times terms$ weighting matrix
I	Set of n items: $I = \{i_1, \cdots, i_n\}$	w_{ij}	Weight of term t_j in review d_i
I_u	Set of items rated/reviewed by u	C_i	Set of contents for item i
r_{ui}	Rating r on item i by user u	C_u	Set of contents for user u
$\hat{r}$	Rating prediction	T_u	Set of terms in D_u
D_u	Set of reviews for user u	d_{ui}	Review d from user u on item i
$t \in T_u$	A term from review $d \in D_u$: it is defined as any unbroken string of characters in a query delimited by the whitespace symbol; and stop words such as "and", "are", and "the", are removed	P	Set of declarative profiles: $P = \{p_{ui} \mid u \in U, i \in I, r_{ui} \neq \oslash, d_{ui} \neq \oslash\}$
$\lvert \cdot \rvert$	The cardinality of a set	p_{ui}	Declarative profile for u on item i
k	The number of latent factors (topics)	zp_u	Learning parameter for u on cluster p_u
n	The number of terms in each topic	zp_i	Learning parameter for item i on cluster p_i

that user puts the contexts/aspects of an item into consideration when doing the item selection and each item has different dominant contexts compared to others.

Inspired by the approach of [8] where each user has different important contexts when consuming items, we also believe that each item has different dominant contexts. Working on top of UW-NMF, we also identify the item profile based on the reviews the item received from users. Given i, item declarative profile P_{iu} is defined by each user u that gives rate and review to item i. Specifically, P_{iu} is generated declaratively from u's reviews. The corresponding part from u's reviews is defined as the *item (hidden) context*. So, in order to identify item profile, for n items, we have n-disjoint item review-window. So given D_i, the set of reviews for item i, and the corresponding set of review terms T_i, we constructed $review \times term$ matrix $\mathbf{W}$ for item i, and define w_{uj}, the weight of term t_j in review d_u:

$$w_{uj} = f_{uj} \cdot \left(\log \frac{1 + |D_i|}{1 + |D_i^j|} + 1 \right), \tag{2}$$

where f_{uj} is the frequency of term t_j appears in review d_u, $|D_i|$ is the number of total reviews for item i, D_i^j is the number of reviews containing term t_j. This weighting function is defined in a format of Term Frequency-Inverse Document Frequency (TF-IDF). Then, we applied $L2$ norm to reduce the biased contribution from either common or rare terms. NMF was applied in each item window $\mathbf{W}$, which decomposes $\mathbf{W}$ into two k-dimensional latent factors $\mathbf{X}$ and $\mathbf{H}$ with property that all three matrices have no negative elements:

$$\mathbf{X} \times \mathbf{H} \approx \mathbf{W}, \text{ where } w_{ij} \approx \sum_{k=1}^{k} x_{ik} h_{kj}, \tag{3}$$

where $\mathbf{X}$ is an $|D_u| \times k$ matrix, representing the *reviews* $*$ *topics* latent subspace; $\mathbf{H}$ is a $k \times |T_u|$ matrix, representing the *topics* $*$ *term* latent subspace. We employed Non-Negative Double Singular Value Decomposition (NNDSVD) [3] to enhance the initialization of matrix $\mathbf{W}$ and $\mathbf{H}$. Then, $\mathbf{X}$ and $\mathbf{H}$ are calculated as follows:

$$\mathbf{X} \leftarrow \mathbf{X} \times \frac{\mathbf{W}\mathbf{H}^T}{\mathbf{X}\mathbf{H}\mathbf{H}^T} \tag{4}$$

$$\mathbf{H} \leftarrow \mathbf{H} \times \frac{\mathbf{X}^T\mathbf{W}}{\mathbf{X}^T\mathbf{X}\mathbf{H}} \tag{5}$$

We define the declarative item profile based on the item i reviewed by user u as follows:

$$p_{iu} = [\underbrace{\tilde{w}_{i1}, \cdots, \tilde{w}_{il}}_{\text{item context}}] \tag{6}$$

Note, given u, $\tilde{w}_{u1}$ is the weight of the first ranked term $t \in T_u$ for topic-1 that was *declared* by user u via his/her reviews, which is calculated as below,

$$\tilde{w}_{uj} = \sum_{j=1}^{|D_i|} w_{uj} \tag{7}$$

where for each review term $t \in T_i$, we inspected whether t appears in the generated item contexts. If the term t exists in the item contexts, we get term t's value by summing all its value across all reviews.

As shown in (6), item profiling covers item's dominant contexts. Note that in item profile, we do not define the item content and user contexts as part of the profile to avoid redundancy with the declarative user profile as defined in (1). Declarative item profiling using UW-NMF can be seen in Algorithm 1, meanwhile please refer to [8] for detail technique and algorithm of the declarative user profile using UW-NMF.

3.2 Item Recommendation with DecPro-CARS

In the previous Sect. 3.1, we have defined personalized declarative item profile which covers the item's dominant contexts and user profiles that consists of user's context, user's content, and the corresponding rated/reviewed item's content. In this section, we incorporate both user and item profiles into the recommendation. Existing work has considered the importance of context for both user or item in item recommendation [10]. However, it justifies the user rating either by the user topics or the item topics and assumes the similar contexts over users and items. Meanwhile [8] only incorporates the user's but not the item profiles. In this work, the DecPro-CARS algorithm generates the item recommendation

Algorithm 1. Declarative Item Profiling

```
Input: $D_i$, $T_i$, $k$: the number of topics
Output: item profile ($P$)
Construct declarative item contexts
for each item $i$ do
    Initialise $\mathbf{W}$ with Eq. 2 and applied L2 norm.
    initialize $\mathbf{X}$ with NNDSVD [3]
    initialize $\mathbf{H}$ with NNDSVD [3]
    update $\mathbf{X}$ and $\mathbf{H}$ with Eq. 4 and Eq. 5.
    select item context from matrix $\mathbf{H}$ by selecting the top $k$ ranked terms for each
    topic (row).
end for
Construct $\tilde{\mathbf{W}}$ and $P$
for each item $i$ do
    for each review $d \in D_i$ do
        for each term $t \in T_i$ do
            if $t$ in item contexts then
                calculate $\tilde{w}$ with Eq. (7)
            end if
        end for
        $p_{iu} = [\tilde{w}_{i1}, \cdots, \tilde{w}_{il}]$
    end for
end for
Return $P = \{p_{iu} | i \in I, u \in U, r_{iu} \neq \oslash, d_{iu} \neq \oslash\}$
```

by matching the item profile with the user profile. Specifically, on the basis of Matrix Factorization, we incorporate declarative user profile (p_{ui}) to the user latent factors (v_u) and declarative item profile (p_{iu}) to the item latent factors (q_i). From this point, user and item profile are incorporated to enhance latent representations of user and item respectively. We can define this officially as follows:

$$\hat{r}_{ui,[c_1 \ldots c_g]} = \mu + b_i + b_u + (v_u \odot z_{p_u})^T \cdot (q_i \odot z_{p_i}), \tag{8}$$

where $\hat{r}_{ui,[c_1 \ldots c_g]}$ denotes the predicted rating of user u on item i in the contexts of $[c_1 ... c_g]$, retrieved from user text review, such as price, colour, and brand. $\odot$ denotes the element-wise multiplication or Hadamard product. While z_{p_u} and z_{p_i} are the parameters which represent a set of similar users (p_u) and similar items (p_i) based on their profiles. When two users u_a and u_b preferred same items, we can assume both users share similar interests. Therefore, if user u_a liked an item i_j that has not been consumed by user u_b, we can recommend item i_j to user u_b. Similarly, we can assume that user u_a will tend to consume another item i_k which shares similar contexts or features to item i_j that user u_a previously enjoyed.

Grouping users based on the similarity of their profile and items to the similarity of their dominant contexts can be done in many different ways. In this work, we assume that contexts were attached each time a user consumes an item and vice versa, and are captured via the *user context* component in p_{ui}:

$[\tilde{w}_{u1}, \cdots, \tilde{w}_{ul}]$ and *item context* component in p_{iu}: $[\tilde{w}_{i1}, \cdots, \tilde{w}_{il}]$. Then, we group the users and items based on the similarity of their contexts. Given an active user u_a or item i_j. First we retrieved the declarative user-item profiling for all users and items (refer to (1)). Later, we identified p_u, c group of users that similar to u_i and p_i is a group of items that similar to u_i in accordance with their contexts, where $p_u \in \{1, 2, \cdots, c\}$ and $p_i \in \{1, 2, \cdots, c\}$. In this paper, we deploy k-means to find the similar the users and similar items based on their profiles. The model tries to minimize the prediction error as below.

$$\mathcal{L}(\theta) = \frac{1}{2} \sum_{u,i,c \ \in \ \kappa} (r_{ui,[c_1 \ldots c_g]} - \hat{r}_{ui,[c_1 \ldots c_g]})^2 + \frac{\lambda}{2} |\theta|_2 \tag{9}$$

where θ is the set of unknown parameters to learn (μ, b_i, b_u, v_u, q_i, p_u, p_i), κ is the set of all user, item, rating, and context pairs, while $\lambda = (\lambda_v, \lambda_q, \lambda_{p_u}, \lambda_{p_i})$ is the set of the regularization parameters. Stochastic gradient descent (SGD) was conducted to update the parameters as follows:

$$\begin{aligned}\frac{\delta \mathcal{L}(\theta)}{\delta v_u} &= -\big(r_{ui,[c_1 \ldots c_g]} - \mu - b_i - b_u - (v_u \odot p_u)^T \cdot \\ &\quad (q_i \odot p_i)\big) \cdot (p_u \odot q_i \odot p_i) + \lambda_v v_u.\end{aligned} \tag{10}$$

$$\begin{aligned}\frac{\delta \mathcal{L}(\theta)}{\delta q_i} &= -\big(r_{ui,[c_1 \ldots c_g]} - \mu - b_i - b_u - (v_u \odot p_u)^T \cdot \\ &\quad (q_i \odot p_i)\big) \cdot (p_i \odot v_u \odot p_u) + \lambda_q q_i.\end{aligned} \tag{11}$$

$$\begin{aligned}\frac{\delta \mathcal{L}(\theta)}{\delta p_u} &= -\big(r_{ui,[c_1 \ldots c_g]} - \mu - b_i - b_u - (v_u \odot p_u)^T \cdot \\ &\quad (q_i \odot p_i)\big) \cdot (q_i \odot p_i) \cdot v_u + \lambda_p p_u.\end{aligned} \tag{12}$$

$$\begin{aligned}\frac{\delta \mathcal{L}(\theta)}{\delta p_i} &= -\big(r_{ui,[c_1 \ldots c_g]} - \mu - b_i - b_u - (v_u \odot p_u)^T \cdot \\ &\quad (q_i \odot p_i)\big) \cdot (v_u \odot p_u) \cdot q_i + \lambda_p p_i.\end{aligned} \tag{13}$$

Like most recommender systems, the proposed DecPro-CARS is trained based on users historical transaction data (e.g. ratings) and identified the profile for both user and item to generate recommendations.

4 Experimental Evaluation

In this section, we discuss the effectiveness and efficiency of our DecPro-CARS by conducting comprehensive experiments on two real datasets, Yelp, and TripAdvisor and demonstrate its superior performance.

4.1 Experimental Setup

We conduct experiments over two publicly available datasets, Yelp and TripAdvisor which are common benchmark dataset for recommender systems. Yelp dataset is a subset of the benchmark set from Yelp Challenge Dataset round 10[2], which includes restaurant review by users over Yelp online site. TripAdvisor is a hotel review data set crawled from TripAdvisor (www.tripadvisor.com) and have been used in [14]. The details of these datasets are described in Table 2. All datasets were created by keeping the users who have rated more than the average reviews.

Table 2. Dataset

Dataset	Users	Items	Transact	Avg. review	Size (MB)
Yelp	495	16,618	136,419	153	2,230
TripAdvisor	2,203	1,640	76,045	3	220

For the recommendation, we selected the ground truth data for each user. The ground truth data for a target user are those rated by this user in the test set. We evaluate whether the recommendation is successful to each user by calculating the MSE, RMSE, and MAE of the predicted rating with the rating in ground truth data. We apply 5-fold cross-validation in this study, and all the baselines are well-tuned as proposed in the author's paper.

4.2 Evaluation Methodology

We have conducted extensive experiments to evaluate the effectiveness and efficiency of our DecPro-CARS. Following the optimal parameter from previous work [16], we define 10 as the optimal topic number k and the number of top terms kept for each topic n. We evaluate the effectiveness of DecPro-CARS using the optimal k and n values. For each dataset, we compare our proposed DecPro-CARS with the five existing competitors which can be categorized into two groups: i) 3 traditional collaborative filtering and ii) 2 content/contextual recommendations:

1. SVD++ [6], exploit both rating information and other implicit feedback.
2. timeSVD [7] that track the time changing behaviour of customer throughout the life span of data.
3. ItemKNN [4] which returns item recommendation based on the similarity of item consumption of users, and
4. EFM [15], using specific product features to the user's interests and the hidden features learned for recommendations.
5. D-CARS [8] is by far the best amongst the methods that consider the different dominant contexts for each user by incorporating item's and user's contents and user's review text to define user profile.

[2] https://www.yelp.com/dataset_challenge/.

LibRec Package [5] is used to run the experiment for SVD++, timeSVD, ItemKNN, and EFM. For EFM, we treat the contexts as positive/negative sentiment based on the existence of the contexts in rating. All the baselines are well-tuned as proposed in the literature. The effectiveness of the system is evaluated by a popular metric *Means Average Error* (MAE), *Means Square Error* (MSE), and *Root Means Square Error* (RMSE), which is defined as formula 14.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(r_{ui,[c_1...c_g]} - \hat{r}_{ui,[c_1...c_g]})^2 \tag{14}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(|r_{ui,[c_1...c_g]} - \hat{r}_{ui,[c_1...c_g]}|) \tag{15}$$

$$RMSE = \frac{1}{n}\sum_{i=1}^{n}\overline{(\sqrt{r_{ui,[c_1...c_g]} - \hat{r}_{ui,[c_1...c_g]})}^2 \tag{16}$$

The three evaluation metrics measures the difference between the set of predicted ratings, $r_{ui,[\hat{c_1}...c_g]}$, and the corresponding ground truth that is the set of ratings, $r_{ui,[c_1...c_g]}$, in the n-size test dataset. MSE indicates the average of the squares of the errors between the predictions and the ground truth. MAE measures the average error, while RMSE indicates the root square of the error. A smaller MSE, RMSE, and MAE indicate the better performance of a system for rating prediction.

4.3 Evaluation on Effectiveness

Effectiveness Comparison with Existing Competitors. We compare the effectiveness of six recommendation methods, DecPro-CARS, D-CARS, SVD++, timeSVD, itemKNN, and EFM by performing recommendation over two datasets to the same sets of target users. We set all methods to their optimal parameter settings, and recommend the predicted items to the target users. Table 3 and 4 depict the comparison result of our DecPro-CARS with the other baselines in terms of MSE, RMSE, and MAE for Yelp and Tripadvisor datasets accordingly.

Clearly, all our proposed DecPro-CARS shows better performance by returning the lowest error in item prediction. SVD++ with traditional collaborative filtering recommendation has similar performance to D-CARS, context-aware recommendation with declarative user profile. It seems that only incorporating the user profile is not enough to boost the D-CARS performance for these two datasets. Meanwhile, we can see that DecPro-CARS can outperform all the baselines by incorporating both user and item profile in the item recommendation. Further investigation of user and item profiling is discussed in Sect. 4.3.

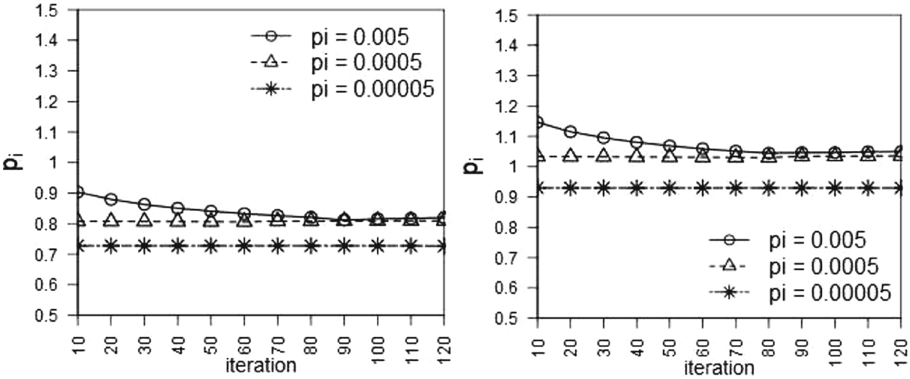
Effect of p_i Parameter. DecPro-CARS has many parameters to learn (μ, b_i, b_u, v_u, q_i, p_u, p_i) as depicted in Eq. 8. In this section, we evaluate the effect of

Table 3. Effectiveness comparison with existing competitors for *Yelp* Dataset in terms of MSE, RMSE, and MAE (the lower the value, the better).

Method	MSE	RMSE	MAE
SVD++	0.9065	0.9374	0.7345
timeSVD	1.4024	1.1842	0.9123
ItemKNN	1.0462	1.0228	0.7997
EFM	1.3579	1.1653	0.8930
D-CARS	0.9255	0.9357	0.7341
DecPro-CARS	**0.8738**	**0.9238**	**0.7295**

Table 4. Effectiveness comparison with existing competitors for *Tripadvisor* Dataset in terms of MSE, RMSE, and MAE (the lower the value, the better).

Method	MSE	RMSE	MAE
SVD++	1.1069	1.0335	0.7964
timeSVD	1.7822	1.3349	0.9893
ItemKNN	1.1633	1.0785	0.8018
EFM	1.9391	1.3925	1.0897
D-CARS	1.3662	0.9602	0.8009
DecPro-CARS	**0.8764**	**0.9361**	**0.7304**



(a) *MAE*

(b) *MSE*

Fig. 2. Effect of p_i parameter

item factor parameter to see the effect of p_i value for the item recommendation. The values we have experimented with are 0.005, 0.0005, and 0.00005. Other parameter settings were kept the same as the baselines.

Table 5. Effectiveness comparison of user-item profiling in recommendation.

User ID = 0		
User Profile = ***fries***, ***place***, *tart, thai, fish, food, minutes, yogurt, pad, media,* ***ordered***, *people, flavors, curry, ak, came,* ***time***, *toppings, panang, yelpcdn,* ***burger***, *room, froyo, rice, bphoto*		
	Item ID	**Item Profile**
DecPro-CARS Recommendation	462 (**)	**time**, friend, garlic, breakfast, salmon, calzone, broguth, pretty, baseball, **burger**, went, eggs, good, im, **food**, pretty, plate, yelp, **place**, **fries**, good, bartenders, sat, know
	8079 (*)	table, you've, iced, hour, dessert, plactic, chills, unhealthy, lunch, order, calories, tea, buffalo, love, bill, experience, **burger**, honey, really, overcooked, matter, damnit, looked, appetizer, avocado
D-CARS Recommendation	21774 (**)	cheese, cook, **place**, go, meal, really, lighting, swiss, dark, long, pot, dessert, good, definitely, course, **food**, awesome, extra, chocho
	23975 (*)	pretty, go, **food**, buffet, theres, indian, paneer, diner, buddy, **ordered**, right, **place**, coffee, reason, looked
User ID = 154		
User Profile = *eggplant, shrimp,* ***fries***, ***food***, *pho,* ***sauce***, *rice,* ***burger***, *indian,tofu, bread, egg,* ***good***, *lamb, broth, fried, potato,* ***chicken***, ***beef***, ***sandwich*** *,****soup***, *cheese, vindaloo,* ***friend***		
	Item ID	**Item Profile**
DecPro-CARS Recommendation	6552 (**)	bronze, house, place, quinoa, stars, cafe, banana, groupon,**chicken**, payment, **sandwich**, mized,number, order, bronze, green, lunch, im, cafe, **food**, book, ⋯, **friend**, got, handle, low, hummus, greens, **good**, free, stafff, vegan, town, certificate, close, **sandwiches**
	11493 (*)	mimis, **good**, **soup**, great, cooked, mozzarella, turkey, ok, french, salad, salmon, breakfast, onion, artichoke, tomato, steak, looked, bf, dressing, potatoes, baked, meal, location, pancakes, ⋯, **fries**, **burger**, app, fresh
D-CARS Recommendation	19449 (**)	meat, lomito, **sandwich**, **good**, egg, **beef**, **sauce**, chicken, barbeque, prices, frites, restaurant, stuffed, priced, sweetness, chinese, drink, grill, soft, ⋯, **chicken**
	28349 (*)	walmart, beewax, lot, pickup, lock, items, fabric, hundreds, employees, back, club, location, go, package, guards, charged, ⋯, staff

From the results in Fig. 2, it can be seen that small p_i parameter increase the prediction accuracy with the lowest MSE and MAE. However, the smaller the value the process becomes longer. In this work, we use the best p_i parameter is 0.00005 for the DecPro-CARS.

Effectiveness Comparison of User and Item Profiling for Item Recommendation. We evaluate the effectiveness of incorporating both user and item profiling in DecPro-CARS to D-CARS in item recommendation which incorporates user profiling online. For both methods, we choose the highly recommended items and the unfavourably recommended items. Next, we compare the item profile of these items to the current user's profile. Due to the space limitation, we display the item recommendation for 2 most active users, 0 and 154 with one item each to represent highly (**) and unfavourably (*) recommended items as shown in Table 5.

As we can see, DecPro-CARS recommends items with more similar profile to the user profile compare to D-CARS. For user 0, DecPro-CARS returns item 462 that has 5 matched terms with the user 0's profile (*time, burger, food, place, fries*), while D-CARS recommends item 21774 that has only 2 similar terms (*place, food*). On the other hand, the lowest recommendation by DecPro-CARS (item: 8079) has a less similar term to the item profile (*burger*). While, the lowest item recommendation (item: 23975) by D-CARS has 3 similar terms (*food, ordered, place*) that are even more compared to its highly recommended item with 2 similar terms. It shows that incorporating both user and item profile in item recommendation increase the recommendation quality by returning more similar items to the user profile.

5 Conclusion

In this paper, we proposed a context-aware recommender system that declaratively defines both user's and item's contexts and effectively recommends items over online sites. First, we proposed a declarative user and item profiling that extracting personalized contexts for each of the users and items from historical transactions. Then we propose an item recommendation algorithm that effectively incorporates the group of similar users and items based on their profile to generate the item recommendation. We have conducted extensive experiments to evaluate our proposed recommendation approach on two real datasets. The experimental results have proved that our proposed approach outperformed the existing methods in terms of efficacy.

Acknowledgments. This work is supported by ARC Discovery Project DP200101 175, the Indonesia Endowment Fund for Education (LPDP), and Del Institute of Technology, Indonesia.

References

1. Baltrunas, L., Amatriain, X.: Towards time-dependant recommendation based on implicit feedback. In: Workshop on CARS 2009 (2009)

2. Baltrunas, L., Ludwig, B., Ricci, F.: Matrix factorization techniques for context aware recommendation. In: Proceedings of the RecSys, pp. 301–304. ACM (2011)
3. Boutsidis, C., Gallopoulos, E.: SVD based initialization: a head start for nonnegative matrix factorization. Pattern Recogn. **41**(4), 1350–1362 (2008)
4. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst. **22**(1), 143–177 (2004)
5. Guo, G., Zhang, J., Sun, Z., Yorke-Smith, N.: LibRec: a Java library for recommender systems. In: UMAP Workshops, vol. 4 (2015)
6. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the ACM SIGKDD, pp. 426–434 (2008)
7. Koren, Y.: Collaborative filtering with temporal dynamics. In: Proceedings of the ACM SIGKDD, pp. 447–456. ACM (2009)
8. Lumbantoruan, R., Zhou, X., Ren, Y., Bao, Z.: D-cars: a declarative context-aware recommender system. In: International Conference on Data Mining (ICDM), pp. 1152–1157. IEEE (2018)
9. Lumbantoruan, R., Zhou, X., Ren, Y., Chen, L.: I-cars: an interactive context-aware recommender system. In: International Conference on Data Mining (ICDM), pp. 1240–1245. IEEE (2019)
10. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of the RecSys, pp. 165–172. ACM (2013)
11. Qin, D., Zhou, X., Chen, L., Huang, G., Zhang, Y.: Dynamic connection-based social group recommendation. IEEE Trans. Knowl. Data Eng. **32**(3), 453–467 (2020)
12. Ren, Y., Tomko, M., Salim, F.D., Chan, J., Clarke, C.L., Sanderson, M.: A location-query-browse graph for contextual recommendation. TKDE **30**(2), 204–218 (2018)
13. Said, A., Luca, E.W.D., Albayrak, S.: Inferring contextual user profiles - improving recommender performance (2011)
14. Wang, H., Lu, Y., Zhai, C.: Latent aspect rating analysis without aspect keyword supervision. In: Proceedings of the ACM SIGKDD, pp. 618–626. ACM (2011)
15. Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., Ma, S.: Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In: Proceedings of ACM SIGIR, pp. 83–92 (2014)
16. Zhou, X., Chen, L.: Event detection over Twitter social media streams. VLDB J. **23**(3), 381–400 (2013). https://doi.org/10.1007/s00778-013-0320-3
17. Zhou, X., Chen, L., Zhang, Y., Cao, L., Huang, G., Wang, C.: Online video recommendation in sharing community. In: Proceedings of the ACM SIGMOD, pp. 1645–1656 (2015)
18. Zhou, X., et al.: Enhancing online video recommendation using social user interactions. VLDB J. **26**(5), 637–656 (2017). https://doi.org/10.1007/s00778-017-0469-2
19. Zhou, X., Qin, D., Chen, L., Zhang, Y.: Real-time context-aware social media recommendation. VLDB J. **28**(2), 197–219 (2018). https://doi.org/10.1007/s00778-018-0524-7
20. Zhou, X., Qin, D., Lu, X., Chen, L., Zhang, Y.: Online social media recommendation over streams. In: 35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8–11, 2019, pp. 938–949. IEEE (2019)

HisRec: Bridging Heterogeneous Information Spaces for Recommendation via Attentive Embedding

Jingwei Ma[1], Lei Zhu[1(✉)], Jiahui Wen[2], and Mingyang Zhong[3]

[1] Shandong Normal University, Jinan, China
leizhu0608@gmail.com
[2] National Innovative Institute of Defense Technology, Beijing, China
[3] The University of Queensland, Brisbane, Australia

Abstract. A large volume of knowledge has been accumulated with the prevalence of social networks. Although extra knowledge can mitigate the problem of data sparsity, it is still a challenging task to integrate data across different information spaces for recommendation. In this paper, we propose to address recommendation that involves heterogeneous information spaces, namely interactive space (i.e. user-item), structural space (user-user) and semantic space (user-attribute). Instead of modeling each information space independently, we propose to seamlessly integrate information across heterogeneous spaces. To do this, we propose an attention mechanism in which the users/items attend differently to their structural neighbors (structural space) for learning compact representations. The attentions are parameterized by the interactions between user/item attributes (semantic space), and they are collaboratively learned for the recommendation task (interactive space). In this way, information across different spaces can be complementary to each other for boosting recommendation performance. We also prove that the proposed attentive embedding method is a generalization of traditional social regularization and network embedding methods. We validate the effectiveness of the proposed model with two real world datasets, and show that the proposed model is able to outperform state-of-the-art recommender models significantly.

Keywords: Recommendation · Attentive embedding

1 Introduction

Collaborative Filtering (CF) [23] is one of the most widely employed recommendation techniques. However, it suffers from data sparseness, as it mainly relies on user-item interactions for learning user preferences. In practical scenarios, there are heterogeneous information spaces available, and they can potentially alleviate data sparseness. Each information space is characterized by space-specific entities and their interactions. For example, Fig. 1 involves three types of infor-

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 428–443, 2020.
https://doi.org/10.1007/978-3-030-65390-3_33

mation spaces, namely interactive information space (i.g. user-item), structural information space (e.g. user-user) and semantic information space (e.g. user-attribute).

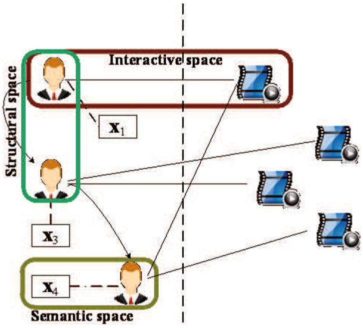


Fig. 1. An example of heterogeneous information spaces.

To mitigate data sparseness, many previous works [19,28,32] incorporate auxiliary information for recommendation. For example, some research [5,11,29] explore social networks for propagating user preferences, while others [30,31] leverage content data for bridging similar items. However, there are some limitations with those methods. First, most of them involve learning representations within a single information space or employ representations from another information space as regularizations. Second, representations are modeled independently in each information space in most of the previous works, while interactions among different information spaces are ignored.

To address those problems, we propose a model (HisRec for short) that involves representation learning from heterogeneous information spaces. To bridge information across multiple spaces, we propose an attention mechanism for learning user/item representation, so that each entity can attend differently to its structural neighbors for informative representation learning. The attention weight between an entity and its neighbor is calculated based on their respective attributes, and hence an entity pays more attention to its informative structural neighbors for learning its representation. Therefore, instead of learning representations for each space independently, we bridge knowledge across heterogeneous information spaces for learning compact representations, and information of different spaces can compensate each other for accurate recommendation. Finally, we incorporate collaborative filtering and attentive representation learning into a unified model. Experimental results demonstrate that the proposed model outperforms the state-of-the-art baselines by a large margin.

The contributions of this work can be concluded as follows.

– We propose to bridge heterogeneous information spaces for recommendation. Information spaces are seamlessly bridged by the proposed attentive representation learning method, which is able to identify the most informative structural neighbors based on the attribute data.

- We incorporate collaborative filtering and attentive representation learning into a probabilistic model, and jointly optimize the model objective for learning compact user/item representations.
- We validate the effectiveness of the proposed model with real datasets, and demonstrate its advantage over the state-of-the-art baselines with extensive experiments.

2 Related Work

One successful collaborative filtering technique widely used by recommender systems is matrix factorization that factorizes rating matrix into low-dimensional user/item feature vectors. However, traditional MF-based methods suffer from data sparsity problem as a user usually gives few ratings compared to the large item set. To solve this problem, many previous works proposed to incorporate additional information for better learning user/item representation for the tasks. For example, in [13], the authors exploit positive and negative emotional information in the reviews for bridging users with similar emotion. Xu et al. [1] make use of autoencoders for extracting high-level hidden representations of tags, and then hybrid them with neural networks for calculating semantic similarity-based reference scores between users and items. The model in [6] has the similar architecture as that in [1]. However, the auxiliary information that they employ autoencoders to learn representations from is user profile and item content information. DeepCoCNN [25] extract hidden representations from user- and item-generate textual reviews with deep neural network, and concatenate the representation pairs in a forward neural network for further modeling their interactions.

Some other works propose to exploit structural user/item information for better learning user/item representations for the recommendation task. For example, TrustSVD [7] leverages additional trust network for learning representations for trustees and trusters, and users and trusters are assumed to share the same latent representations to bridge them together. Ma et al. [12] model social network information as regularization terms, and jointly optimize the objective function of matrix factorization and the social regularization. In [24], the authors predict user/item representations with their respective contexts based on Skipgram model [14]. The learned representations are able to preserve the 2nd proximity of the user- and item-network, as users/items with similar contexts are supposed to be in a proximity closed to each other in the latent space. Wang et al. [22] address the cross-domain recommendation by leveraging information from social domain for propagating user representations in the information domain. However, the drawback of the aforementioned models is that most of them involve single information space or incorporate knowledge from the second information space as regularization. Therefore, there are not effective for learning comprehensive user/item representations and are inapplicable in our case.

Some researchers [4,18] claim to solve recommendation problem in heterogeneous information networks. The basic idea of these works is to measure the similarities between the users based on different meta paths, and aggregate the ratings of similar users for estimating the preference of the target user. However, those methods are mainly based on structural information space, and they are inapplicable in semantic information space. Another work, Attentive Collaborative Filtering (ACF) [2] proposes a hierarchical attention mechanism for item recommendation. Specifically, items attend differently to their components and users attend differently to their interacted items for user preferences modeling. However, like aforementioned models, ACF first extract item representations in the semantic information space, and interact them with user in the interactive information space for user preferences modeling. Therefore, it cannot address challenging cross-heterogeneous information space recommendation.

Our work is related to previous recommendation models that consider heterogeneous data sources. Representative works include CKE [26] and JRL [27]. CKE discovers hidden high-level representations from each data modality (e.g. text, image), and then generate unified item representation from the linearly combination of the heterogeneous representations. On the contrary, JRL models user-item interaction in each data source, and adds up the interaction scores to rerank the top-N items. Therefore, an item can be ranked higher in the final recommendation list as long as user preference is properly profiled in any one of the data sources. Even though those methods are able to address data heterogeneity, they cannot be applied directly to recommendation with heterogeneous information spaces, as data of different modalities are in the same information space (i.g. semantic information space). Hence they cannot solve the challenges of recommendation with heterogeneous information space discussed in Sect. 1.

The proposed model is also related to network embedding [3] that learns low-dimensional vector representations of nodes in a network for information retrieval tasks such as node classification, node clustering and link prediction. Recent year has witnessed the efforts for learning node representations with external information, such as network semantics [16]. However, those embedding techniques are mainly based on network structure, and are incapable for learning representations across heterogeneous information spaces.

3 Preliminaries

3.1 Problem Formulation

We denote a user-item **interactive** matrix as $\mathbf{R} \in \{0,1\}^{M \times N}$, where M and N are the number of users and items respectively. An non-empty entry $r_{ij} = 1$ refers to a positive interaction (e.g. visit, check-in, rate, purchase) between user u_i and item v_j. In our case, we have an additional user **structural** network $G_u = (U, E_u)$, where each edge $(u_i, u_k) \in E_u$ indicates an interrelation between the user pair. Also, users are associated with a **semantic** attributes matrix $\mathbf{X} \in \mathbb{R}^{d_u \times M}$, and each column in the attribute matrix, $\mathbf{x}_i$, is the corresponding attribute vector (e.g. user demographics) for user u_i. Similarly, we have structural network $G_v = (V, E_v)$ and semantic attributes matrix $\mathbf{Y} \in \mathbb{R}^{d_v \times N}$ for items. The task of the proposed recommendation model is to leverage the aforementioned information for predicting the missing values (i.e. $\hat{r}_{ij}$) in $\mathbf{R}$, and recommend to each user the items with high prediction scores.

3.2 Basic Model

We employ probabilistic matrix factorization (PMF) [15] as our base model. Given a user latent vector $\mathbf{u}_i$ and item latent vector $\mathbf{v}_j$, the corresponding rating score $\hat{r}_{ij}$ can estimated as:

$$\begin{aligned} \mathbf{z}_{ij} &= \mathbf{u}_i \oplus \mathbf{v}_j \oplus \mathbf{u}_i^T \mathbf{v}_j \oplus (\mathbf{u}_i \circ \mathbf{v}_j) \\ \hat{r}_{ij} &= f_L(\cdots f_2(f_1(\mathbf{z}_{ij}))\cdots) \end{aligned} \tag{1}$$

where $\oplus, \circ$ are the concatenation and element-wise product operation respectively. $\mathbf{z}_{ij}$ is the concatenation of $\mathbf{u}_i, \mathbf{v}_j, \mathbf{u}_i \circ \mathbf{v}_j$, and $f_L(\cdots f_2(f_1())\cdots)$ is a multiple-layer neural network with sigmoid function at the last layer. In PMF, the conditional probability over observed implicit interactions is as follows:

$$\begin{aligned} P(\mathbf{R}, \mathbf{U}, \mathbf{V}|\sigma, \sigma_u, \sigma_v) = &\prod_{i=1}^{M} \mathcal{N}(\mathbf{u}_i|0, \sigma_u^2\mathbf{I}) \prod_{j=1}^{N} \mathcal{N}(\mathbf{v}_j|0, \sigma_v^2\mathbf{I}) \\ &\prod_{(\mathbf{u}_i,\mathbf{v}_j)\in\mathcal{D}} \mathcal{N}(r_{ij}|\hat{r}_{ij}, \sigma^2) \end{aligned} \tag{2}$$

where $\mathcal{D}$ is the training set, and $\mathcal{N}$ is the normal distribution.

3.3 Attentive Representation Learning

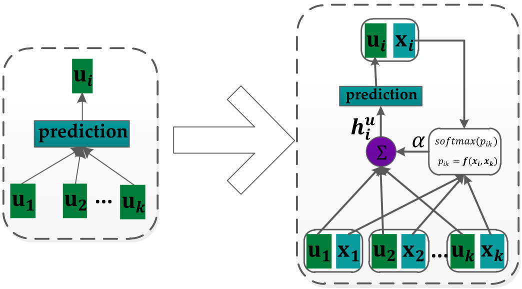


Fig. 2. Comparison between Skip-gram-based representation learning and the proposed attentive representation learning.

The core of the HisRec is the proposed attentive representation learning method. With the attention mechanism, we can identify informative structural neighbors for representation learning based on the attribute data. In this light, we are able to bridge structural and semantic information spaces, so that information from heterogeneous spaces can be complimentary to each other for learning compact representations.

As shown in the left figure of Fig. 2, in a Skip-gram-based model [17], learning the representation of a target user u_i is to predict his probability given his social contexts $C(u_i) = \{u_k|(u_i, u_k) \in E_u\}$. Alternatively, one can predict the social contexts given the user representations [14]. Existing works [17,20] make conditional independence assumption among the social contexts, and the predictive probability of the target user is formulated as

$$P(u_i|C(u_i)) = \prod_{u_k \in C(u_i)} \frac{exp(\mathbf{u}_i^T \mathbf{u}_k)}{\sum_{u_p \in U} exp(\mathbf{u}_p^T \mathbf{u}_k)} \tag{3}$$

On the contrary, we attend differently to the social neighbors for predicting the predictive probability of the target user, and the attentions are as a function of the attributes between the target users and their social neighbors.

$$\begin{aligned} p_{ik} &= \boldsymbol{\beta}_u^T tanh(\mathbf{W}_{u_1}^T \mathbf{x}_i + \mathbf{W}_{u_2}^T \mathbf{x}_k + \mathbf{b}_u); \quad \alpha_{ik} = softmax(p_{ik}) \\ \mathbf{h}_i^u &= \sum_{u_k \in C(u_i)} \alpha_{ik} \mathbf{u}_k; \quad P(u_i|C(u_i)) = \frac{exp(\mathbf{u}_i^T \mathbf{h}_i^u)}{\sum_{u_p \in U} exp(\mathbf{u}_p^T \mathbf{h}_i^u)} \end{aligned} \tag{4}$$

where p_{ik} is an intermediate scalar measuring the interaction between user u_i and his/her social context u_k, and the measurement is based on their respective attribute vectors $\mathbf{x}_i$ and $\mathbf{x}_k$. α_{ik} is the normalized scalar of p_{ik}. $\mathbf{h}_i^u$ is a weighted sum of the representations of u_i's social contexts, and it can be viewed as the contextual representation of u_i. $\mathbf{W}_{u_1}, \mathbf{W}_{u_2} \in \mathbb{R}^{d_u \times k}$ are weight matrices, and $\mathbf{b}_u$ is a bias vector and $\boldsymbol{\beta}_u \in \mathbb{R}^{k \times 1}$ is a parameter vector. As illustrated in the right figure in Fig. 2, we first model the interactions between the target user and each of its social neighbor, which results in an attention score α_{ik}. Then, contextual representation of the target user is calculated as the weighted sum of the representations of his social neighbors. As such, representation of a given user attends differently to his structural neighbors, and the attentions are parameterized by the interaction of the attributes between the target user and his/her counterpart social contexts. Similarly, the predictive probability of an item can be formulated as follows.

$$\begin{aligned} p_{jl} &= \boldsymbol{\beta}_v^T tanh(\mathbf{W}_{v_1}^T \mathbf{y}_j + \mathbf{W}_{v_2}^T \mathbf{y}_l + \mathbf{b}_v); \quad \alpha_{jl} = softmax(p_{jl}) \\ \mathbf{h}_j^v &= \sum_{v_l \in C(v_j)} \alpha_{jl} \mathbf{v}_l; \quad P(v_j|C(v_j)) = \frac{exp(\mathbf{v}_j^T \mathbf{h}_j^v)}{\sum_{v_p \in V} exp(\mathbf{v}_p^T \mathbf{h}_j^v)} \end{aligned} \tag{5}$$

where $C(v_j) = \{v_l|(v_j, v_l) \in E_v\}$ is the collection of structural contexts for item v_j. $\mathbf{W}_{v_1}, \mathbf{W}_{v_2} \in \mathbb{R}^{d_v \times k}$ are weight matrices, and $\mathbf{b}_v$ is a bias vector and $\boldsymbol{\beta}_v \in \mathbb{R}^{k \times 1}$ is a parameter vector. p_{jl} measures the interaction between item v_j and v_l based on their respective attributes $\mathbf{y}_j$ and $\mathbf{y}_l$, and α_{jl} is the normalization of p_{jl} and it can be regarded as the attention score that v_j assigns to v_l for representation learning. $\mathbf{h}_j^u$ is the aggregation of the representations of v_j's structural contexts, and it can be viewed as the context representation for v_j.

3.4 The Unified Model

In this subsection, we propose to seamlessly combine the deep interaction modeling and the attentive representation learning into a unified model. HisRec is a hybrid model, which combines collaborative filtering and attention mechanism for learning compact user/item representations across heterogeneous information spaces. The unified probabilistic can be mathematically described as in Eq. (6).

$$\begin{aligned} & P(\mathbf{R}, \mathbf{U}, \mathbf{V} | \mathbf{X}, \mathbf{Y}, G_u) \\ = & P(\mathbf{R}|\mathbf{U}, \mathbf{V}) P(\mathbf{U}|G_u) P(\mathbf{V}|G_v) P(\mathbf{U}) P(\mathbf{V}) \\ = & \prod_{(\mathbf{u}_i, \mathbf{v}_j) \in \mathcal{D}} \mathcal{N}(r_{ij}|\hat{r}_{ij}, \sigma^2) \prod_{j=1}^{N} \mathcal{N}(\mathbf{v}_j|0, \sigma_v^2 \mathbf{I}) \prod_{i=1}^{M} \mathcal{N}(\mathbf{u}_i|0, \sigma_u^2 \mathbf{I}) \\ & \prod_{i=1}^{M} \frac{exp(\mathbf{u}_i^T \mathbf{h}_i^u)}{\sum_{u_p \in U} exp(\mathbf{u}_p^T \mathbf{h}_i^u)} \prod_{j=1}^{N} \frac{exp(\mathbf{v}_j^T \mathbf{h}_j^v)}{\sum_{v_p \in V} exp(\mathbf{v}_p^T \mathbf{h}_j^v)} \end{aligned} \tag{6}$$

Taking the negative log-likelihood of Eq. (6), we have the objective function as follows

$$\begin{aligned} \mathcal{J} = & \frac{1}{2} \sum_{(\mathbf{u}_i, \mathbf{v}_j) \in \mathcal{D}} (r_{ij} - \hat{r}_{ij})^2 + \frac{\lambda_u}{2} \sum_i ||\mathbf{u}_i||^2 + \frac{\lambda_v}{2} \sum_j ||\mathbf{v}_j||^2 \\ & - \alpha \sum_{i=1}^{M} log \frac{exp(\mathbf{u}_i^T \mathbf{h}_i^u)}{\sum_{u_p \in U} exp(\mathbf{u}_p^T \mathbf{h}_i^u)} - \alpha \sum_{j=1}^{N} log \frac{exp(\mathbf{v}_j^T \mathbf{h}_j^v)}{\sum_{v_p \in V} exp(\mathbf{v}_p^T \mathbf{h}_j^v)} \end{aligned} \tag{7}$$

where $\lambda_u = \sigma^2/\sigma_u^2, \lambda_v = \sigma^2/\sigma_v^2$, and $\alpha = 2\sigma^2$. The graphical representation of the proposed model is shown in Fig. 3.

3.5 Model Learning

Due to the nature of implicit feedback [9] and the large number of items in the recommendation task, we randomly sample unobserved items for every user as negative samples [14]. In Eq. (7), the calculation of $\sum_{u_p \in U} exp(\mathbf{u}_p^T \mathbf{h}_i^u)$ requires summation over all the users and incurs high computational overhead. In practice, we adopt negative sampling [14] to approximate $log \frac{exp(\mathbf{u}_i^T \mathbf{h}_i^u)}{\sum_{u_p \in U} exp(\mathbf{u}_p^T \mathbf{h}_i^u)}$ as:

$$log\sigma(\mathbf{u}_i^T \mathbf{h}_i^u) + \sum_{s=1}^{r} log\sigma(-\mathbf{u}_{is}^T \mathbf{h}_i^u) \tag{8}$$

where $\mathbf{u}_{is}, s = 1, 2, \cdots, r$ are r negative samples for $\mathbf{u}_i$. Similarly, $log \frac{exp(\mathbf{v}_j^T \mathbf{h}_j^v)}{\sum_{v_p \in V} exp(\mathbf{v}_p^T \mathbf{h}_j^v)}$ can be approximated as:

$$log\sigma(\mathbf{v}_j^T \mathbf{h}_j^v) + \sum_{s=1}^{r} log\sigma(-\mathbf{v}_{js}^T \mathbf{h}_j^v) \tag{9}$$

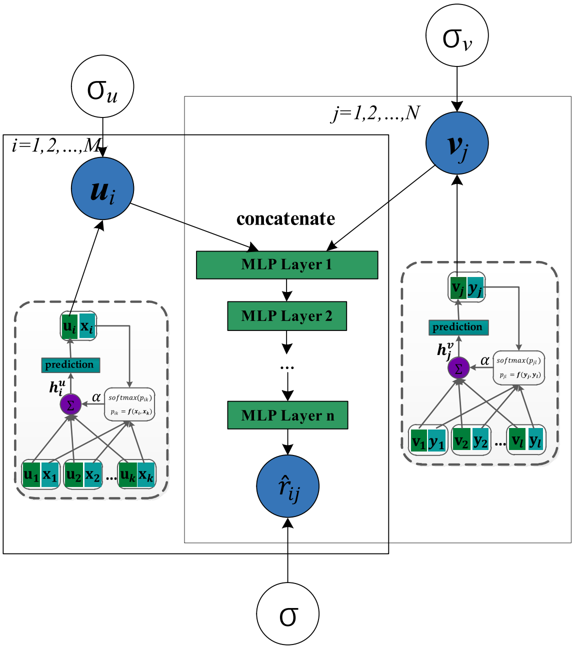


Fig. 3. Graphical representation of the proposed model.

The objective function $\mathcal{J}$ can be optimized with stochastic gradient descent. It updates parameters every step along the gradient direction:

$$\boldsymbol{\theta}^{new} \leftarrow \boldsymbol{\theta}^{old} - lr\frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}} \tag{10}$$

where $\boldsymbol{\theta}$ are the parameters, and lr is the learning rate. $\frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}}$ is the partial derivative of the $\mathcal{J}$ with respect to the model parameters, and it can be automatically computed with typical deep learning libraries.

4 Experiments

4.1 Datasets Collections and Preprocessing

To validate the effectiveness of the proposed model, we focus on texts recommendation, which recommends to each user the texts that may be of interest to each user based on historical activities (for example, questions with answers). With the publicly available crawler in **Github**[1], we collect a month's data from July 2017 to August 2017 from two popular CQA services, *Quora*[2] and *Zhihu*[3].

[1] https://github.com/scku/Quora-Crawler.

[2] http://www.quora.com.

[3] http://www.zhihu.com.

Inspired by the previous work [10], we filter out the users with fewer than 5 interactions. The *Quora* dataset finally consists of 7761 users, 19058 texts and 89442 their interactions. The *Zhihu* dataset includes 10928 users, 29822 texts and 267791 their interactions. In addition, their sparsity rates are 99.0940% and 99.917%, respectively. On average, for the Quora dataset, we have 11.4 interactions for each user and 4.6 users for each text. While for the second dataset, we have 24.05 interactions for each user and 9 users for each text. Moreover, we extract user-user network and user-attribute information from the dataset for texts recommendation with heterogeneous information spaces. To summarize, we crawler user following network as the user-user network for propagating user preferences. As for the semantic information space, user-attribute information contains age, gender, occupation and the topics (i.e. tags) that they follow is encoded into a normalized vector of length 200 for *Quora* and 321 for *Zhihu*. Since item-item structural network is not explicitly available, we calculate the pair-wise cosine similarity between the rating vectors, and select for each item 20 most similar items as its neighbors. As for the item-attribute matrix, we encode the tags associated with each item into a vector of length 152 for *Quora* and 273 for *Zhihu*.

4.2 Experimental Setup

We split the datasets into training set (70%), validation set (10%), and testing set (20%). For each positive user-item pair, we randomly sample 5 unobserved items as negative samples. During the testing process, for each user, we rank each ground truth item along with 99 randomly sampled items, and measure the recommendation performance with widely employed Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). The ranking list is truncated as $k \in [1, 2, \cdots, 10]$.

We set the initial learning rate to 0.0001, and mini-batch size 256. The model is trained for a maximum of 500 epochs, and the model parameters are fine-tuned on the validation set. We employ two-hidden-layer neural network for modeling user-item interactions, and the number of unites are 64 and 16 respectively. The embedding size for users/items is set to 64, while the weight (α in Eq. 7) for attentive representation learning is set to 0.01.

4.2.1 Baselines

- NeuMF [8]. It generalizes matrix factorization and multiple-layer perceptron with an end-to-end neural network.
- PACE [24]. Beside modeling user-item interaction with neural networks, it leverages user/item contexts for bridging gaps between similar users/items.
- HERec [18]. It measures user-user similarities based on meta paths across heterogeneous information networks for recommendation.
- SHINE [21]. It utilizes autoencoder for embedding heterogeneous network (e.g. social, profile network), then combine the embedding vectors for sentiment prediction.

We select those representative methods as our baselines since they cover state-of-the-art recommender models that consider different sources. For example, NeuMF models user and item representations mainly based on the interaction data, while PACE consider user-user network for regularizing similar users and propagating user preferences. Finally, SHINE leverage data sources of different modalities for boosting recommendation performance.

To validate the effectiveness of the proposed attention mechanism, we also study different variants of HisRec:

- HisRec_u is the variant with only the user attentions, and the items attend equally to their structural neighbors for representation learning.
- HisRec_i is the variant with only the item attentions, and the users attend equally to their structural neighbors for representation learning.
- HisRec_o is the variant excluding both user and item attentions.

5 Experiment Results

5.1 Top-K Recommendation

The comparisons between HisRec and the baselines are presented in Fig. 4 and Fig. 5. From the figures, we can draw the following conclusions.

All the other models consistently achieve better performance than NeuMF, as NeuMF mainly relies on user-item interactions to learn user/item representations, and it suffers from data sparseness. The superiority of the other models over NeuMF demonstrates that the incorporation of auxiliary information can alleviate data sparseness and boost recommendation performance.

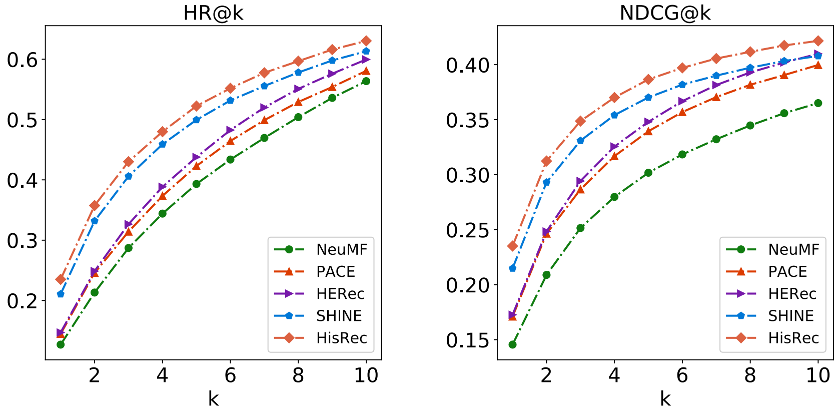


Fig. 4. Evaluation of Top-k item recommendation where k ranges from 1 to 10 on Quora

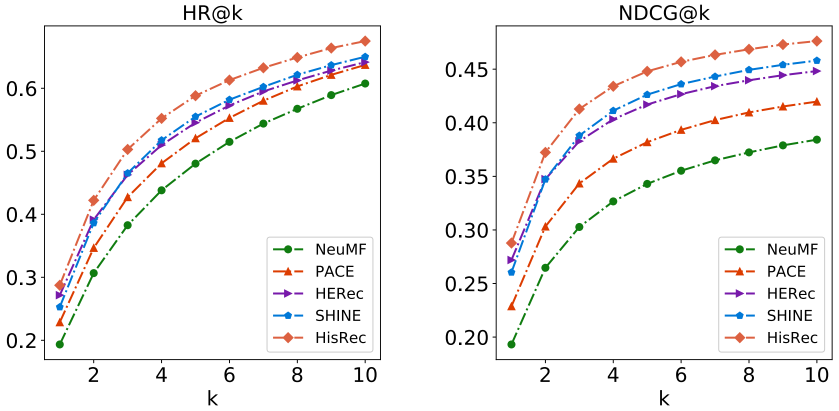


Fig. 5. Evaluation of Top-k item recommendation where k ranges from 1 to 10 on zhihu

Table 1. Comparisons among variants of HisRec.

Models	*Quora*				*Zhihu*			
	HR@5	HR@10	NDCG@5	NDCG@10	HR@5	HR@10	NDCG@5	NDCG@10
HisRec-o	0.42297	0.58072	0.3395	0.39971	0.54538	0.64147	0.41699	0.44824
HisRec-u	0.51909	0.62278	**0.39271**	**0.42632**	0.57017	0.6653	0.41094	0.44165
HisRec-i	0.50844	0.61118	0.38502	0.4183	0.58447	**0.67846**	0.44524	0.47587
HisRec	**0.52239**	**0.63046**	0.38657	0.42164	**0.58819**	0.67491	**0.44799**	**0.47623**

PACE outperforms NeuMF across the datasets and metrics. The behind reason is that PACE jointly predicts contextual information using user/item embeddings, and users/items with similar contexts are supposed to similar contexts. In PACE, contexts act as bridges for mitigating data sparseness and boosting recommendation performance.

HERec achieves better performance than PACE across different datasets and metrics. One possible explanation is that PACE only captures 2nd user proximity, given the fact it predicts user contexts with user representations. By contract, HERec measures user similarities based on meta paths across heterogeneous information space, and the Random Walk it employed to obtain the meta paths is demonstrated to be able to capture proximity of different orders.

SHINE performs slightly better than HERec. This is because SHINE involves multiple information sources for recommendation, and an item can be ranked higher in a user's recommendation list, as long as the user's preference over the item can be appropriately modeled in any one of the data sources.

It can be seen from the figures that the proposed model, HisRec, yields the best recommendation performance in various settings. Specifically, for the top-10 recommendation, HisRec outperforms the best baseline with relative improvements of 5.14% (HR@10) and 2.86% (NDCG@10) on *quora*, and 5.21% (HR@10) and 6.24%(NDCG@10) on *zhihu*. In HisRec, the proposed attention mechanism can filter out noises, and pay more attention to the informative structural neighbors for learning comprehensive representations. This experiment demonstrates that HisRec can effectively bridge information of heterogeneous spaces, and information from different spaces can compensate each other for improving recommendation performance.

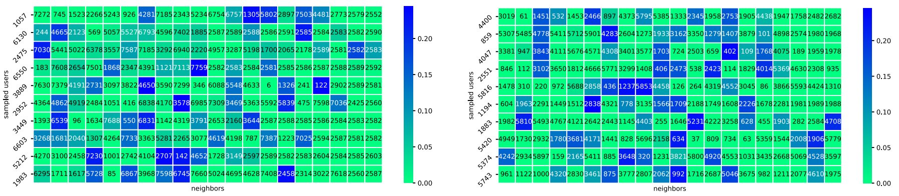
5.2 Efficacy of Attention

In this subsection, we compare three variants with the HisRec. The variants exclude either one or both attention mechanisms. Therefore, the comparison between the proposed model and the variables can verify the effectiveness of the proposed attention mechanism. The results are provided in Table 1. We have observed that in most cases, the recommendation effect of HisRec-i and HisRec-u is better than HisRec-o, which shows that the proposed attention mechanism can actually discriminate the most informative information in heterogeneous spaces, and can improve recommendation.

Even though HisRec-u and HisRec-i performs slight better than HisRec in some cases, the relative improvement is insignificant and negligible(i.e. smaller than 1.5%). On the contrary, HisRec steadily provides the overall best recommendation performance across different datasets and metrics.

5.3 Visualization of Attentions

We visualize attention weights of 10 randomly sampled users on each of the datasets in Fig. 6, where each row starts with a user ID, followed by his/her structural neighbors. We truncate the number of structural neighbors at 20, and for those with fewer than 20 neighbors, we compare their rating vectors with that of all the other users and select the top-k most similar (e.g. cosine similarity) users as their neighbors [12]. We use color depth to visualize the attentions that each user pays to his structural neighbors, with dark color indicating high attention and light color representing low attention. We can observe from figure that most of the users attend to a small number of their structural neighbors for learning representations. The visualization of the attentions demonstrates that



(a) Quora

(b) Zhihu

Fig. 6. Attentions visualization of 10 randomly sampled users on *Quora* and *Zhihu*.

structural information space contains a lot of noises, that is, entities with structural links do not necessarily mean that they share similar representations in the latent space. Therefore, it is necessary to bridge information across heterogeneous spaces and distinguish the most informative data for recommendation.

5.4 Sensitivity Analysis

In this subsection, we investigate the recommendation performance of HisRec with respect to different hyper-parameters. Specifically, we vary the size of latent factor $D \in \{32, 64, 128, 256\}$, and the weight for attentive representation learning (i.e. α in Eq. 7) $\alpha \in \{0.0001, 0.001, 0.01, 0.1\}$. Table 2 summarizes the performance of top-10 recommendation for HisRec across the datasets under different embedding sizes. We can see that HisRec experiences a significant improvement when the embedding size is increased from 32 to 64. However, the performance maintains a relatively stable level with larger embedding size. HisRec yields the best recommendation performance with the embedding size $D = 128$ on *Zhihu*. This may be caused by the larger size and higher density of *Zhihu* dataset, and larger embedding size is required to sufficiently capture user preferences.

Table 2. Recommendation performance as a function of embedding size.

D	*Quora*		*Zhihu*	
	HR@10	NDCG@10	HR@10	NDCG@10
32	0.6068	0.36453	0.6653	0.44165
64	0.63046	0.42164	0.67491	0.47623
128	0.63892	0.41623	0.67935	0.50685
256	0.63315	0.39458	0.6777	0.46771

The performance of HisRec with respect to α is shown in Table 3. The weight determines the similarity between an entity (i.e. user,item) and its attentive structural neighbors. We can observe from the table that, the performance degrades dramatically with large or small α. On the one hand, small α weakens the information flow from structural and semantic spaces, and makes HisRec mainly rely on interactive space for recommendation. On the other hand, large α exerts strong regularization on user/item representations, and this is not concern with realistic scenario. However, HisRec can achieve the optimal performance with a wide range of $\alpha \in [0.01, 0.001]$.

Table 3. Recommendation performance as a function of α (Eq. 7).

α	*Quora*		*Zhihu*	
	HR@10	NDCG@10	HR@10	NDCG@10
1e−4	0.4757	0.27194	0.63441	0.41025
1e−3	0.62844	0.37744	0.67012	0.45803
1e−3	0.63046	0.42164	0.67491	0.47623
1e−1	0.61325	0.4182	0.6074	0.41043

6 Conclusion

In this paper, we propose a recommendation model that is able to seamlessly integrate heterogeneous information spaces for recommendation. Rather than modeling each information independently, we proposed an attention mechanism that is able to bridge information across heterogeneous spaces, so that information from different spaces can compensate each other for learning better representations. In the attention mechanism, users/items attend differently to their structural neighbors (structural information space) for learning their representations, and the attention weights are formulated as functional similarities between the attributes of a user/item and that of its structural neighbors. We also theoretically prove that the proposed attentive representation learning method is generic and can express and generalize traditional social regularization and network embedding methods. Extensive experiments and comparative analysis on two real datasets demonstrate the advantage of the proposed model over the state-of-the-art baselines, and the effectiveness of the proposed attention mechanism.

References

1. Xu, Z., Lukasiewicz, T., Chen, C., Miao, Y., Meng, X.: Tag-aware personalized recommendation using a hybrid deep model. In: Proceedings of the International Conference on IJCAI, pp. 3196–3202. AAAI Press (2017)
2. Chen, J., Zhang, H., He, X., Nie, L., Liu, W., Chua, T.S.: Attentive collaborative filtering: multimedia recommendation with item-and component-level attention. In: Proceedings of International Conference on SIGIR, pp. 335–344. ACM (2017)
3. Cheng, Z., Shen, J., Nie, L., Chua, T.S., Kankanhalli, M.: Exploring user-specific information in music retrieval. In: Proceedings of the International Conference on SIGIR, pp. 655–664 (2017)
4. Cheng, Z., Shen, J., Zhu, L., Kankanhalli, M.S., Nie, L.: Exploiting music play sequence for music recommendation. In: Proceedings of the International Conference on IJCAI, vol. 17, pp. 3654–3660 (2017)
5. Cui, H., Zhu, L., Li, J., Yang, Y., Nie, L.: Scalable deep hashing for large-scale social image retrieval. IEEE Trans. Image Process. **29**, 1271–1284 (2019)
6. Dong, X., Wu, Z., Yuxia, S., Lingfeng, Y., Zhang, F.: A hybrid collaborative filtering model with deep structure for recommender systems. In: Proceedings of the International Conference on AAAI (2017)

7. Guo, G., Zhang, J., Yorke-Smith, N.: TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: Proceedings of the International Conference on AAAI, pp. 123–129. AAAI Press (2015)
8. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the International Conference on WWW, pp. 173–182. ACM (2017)
9. Hu, G., Zhang, Y., Yang, Q.: CoNet: collaborative cross networks for cross-domain recommendation. In: Proceedings of the International Conference on CIKM (2018)
10. Liang, D., Charlin, L., McInerney, J., Blei, D.M.: Modeling user exposure in recommendation. In: Proceedings of the 25th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, pp. 951–961. ACM (2016)
11. Lu, X., Zhu, L., Cheng, Z., Li, J., Nie, X., Zhang, H.: Flexible online multi-modal hashing for large-scale multimedia retrieval. In: Proceedings of the 27th ACM International Conference on Multimedia, pp. 1129–1137 (2019)
12. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: Proceedings of the International Conference on WSDM, pp. 287–296. ACM (2011)
13. Meng, X., Wang, S., Liu, H., Zhang, Y.: Exploiting emotion on reviews for recommender systems. In: Proceedings of the International Conference on AAAI (2018)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
15. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems, pp. 1257–1264 (2008)
16. Ni, J., Chang, S., Liu, X., Cheng, W.: Co-regularized deep multi-network embedding. In: Proceedings of the International Conference on WWW, pp. 469–478. ACM (2018)
17. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representation. In: Proceedings of the International Conference on SIGKDD, pp. 701–710. ACM (2014)
18. Shi, C., Hu, B., Zhao, W.X., Yu, P.S.: Heterogeneous information network embedding for recommendation. IEEE Trans. Knowl. Data Eng. **31**(2), 357–370 (2019)
19. Shi, D., Zhu, L., Cheng, Z., Li, Z., Zhang, H.: Unsupervised multi-view feature extraction with dynamic graph learning. J. Vis. Commun. Image Represent. **56**, 256–264 (2018)
20. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, pp. 1067–1077. ACM (2015)
21. Wang, H., Zhang, F., Hou, M., Xie, X., Guo, M., Qi, L.: SHINE: signed heterogeneous information network embedding for sentiment link prediction. In: Proceedings of the International Conference on Web Search and Data Mining, pp. 592–600. ACM (2018)
22. Wang, X., He, X., Nie, L., Chua, T.S.: Item silk road: recommending items from information domains to social users. In: Proceedings of the International Conference on SIGIR, pp. 185–194 (2017)
23. Xu, Y., Zhu, L., Cheng, Z., Li, J., Sun, J.: Multi-feature discrete collaborative filtering for fast cold-start recommendation. In: Proceedings of the International Conference on AAAI, vol. 34, pp. 270–278 (2020)

24. Yang, C., Bai, L., Zhang, C., Yuan, Q., Han, J.: Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In: Proceedings of the International Conference on SIGKDD, pp. 1245–1254. ACM (2017)
25. Zheng, L., Noroozi, V., Yu, P.S.: Joint deep modeling of users and items using reviews for recommendation. In: Proceedings of the International Conference on WSDM, pp. 425–434. ACM (2017)
26. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.Y.: Collaborative knowledge base embedding for recommender systems. In: Proceedings of the International Conference on SIGKDD, pp. 353–362. ACM (2016)
27. Zhang, Y., Ai, Q., Chen, X., Croft, W.: Joint representation learning for top-n recommendation with heterogeneous information sources. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1449–1458. ACM (2017)
28. Zhang, Z., Liu, L., Shen, F., Shen, H.T., Shao, L.: Binary multi-view clustering. IEEE Trans. Pattern Anal. Mach. Intell. **41**(7), 1774–1782 (2018)
29. Zhang, Z., et al.: Inductive structure consistent hashing via flexible semantic calibration. IEEE Trans. Neural Netw. Learn. Syst. **PP**, 1–15 (2020)
30. Zheng, C., Zhu, L., Lu, X., Li, J., Cheng, Z., Zhang, H.: Fast discrete collaborative multi-modal hashing for large-scale multimedia retrieval. IEEE Trans. Knowl. Data Eng. **32**, 2171–2184 (2019)
31. Zhu, L., Huang, Z., Li, Z., Xie, L., Shen, H.T.: Exploring auxiliary context: discrete semantic transfer hashing for scalable image retrieval. IEEE Trans. Neural Netw. Learn. Syst. **29**(11), 5264–5276 (2018)
32. Zhu, L., Lu, X., Cheng, Z., Li, J., Zhang, H.: Deep collaborative multi-view hashing for large-scale image search. IEEE Trans. Image Process. **29**, 4643–4655 (2020)

A Neighbor-Aware Group Recommendation Algorithm

Rong Pu, Bin Wang(✉), Xiaoxu Song, Xinqiang Xie, and Jing Qin

School of Computer Science and Engineering, Northeastern University,
Shenyang 110004, China
purong0519@126.com, binwang@mail.neu.edu.cn, songxiaoxu112@163.com,
xiexq@foxmail.com, annyproj@126.com

Abstract. With the rapid development of e-commerce, how to recommend the groups that are most likely to buy a certain kind of items to the merchants accurately has become an increasing research of scholars. However, the existing group recommendation technology rarely considers the influence of the close relationship between users on user preferences. Thus we propose a group recommendation model NGRN to generate groups and make recommendations. First we extract k-core groups on the social network, the groups meet the conditions that each user has k neighbors at least. Then we get the recommendable probability of candidate groups under different items. At the same time, the validity of this method is verified on two public datasets. Experiment shows our model has a great improvement for recommendation accuracy compared with other models.

1 Introduction

Group recommendation has attracted more and more attention from scholars [1,12]. Traditional recommendation methods are based on the assumption that user preferences are independent of each other [11,14]. However, the preferences of users are not independent of each other [15], and we want to find the possible group of users who are most likely to buy something from the perspective of the item provider, item providers can then intuitively know which group to recommend to more effectively. In addition, users' preferences are influenced by their neighbors' preferences. Thus the traditional recommendation method is not applicable.

Example. *As shown in Fig. 1, solid lines indicate the existence of neighborhood relationships between users, and dashed lines indicate the existence of observable user-item interactions. If Susan and Bob have similar preferences, Susan is likely*

The work is partially supported by the National Natural Science Foundation of China (Nos. U1736104), and the Fundamental Research Funds for the Central Universities (No. N171602003), Ten Thousand Talent Program (No. ZX20200035), Liaoning Distinguished Professor (No. XLYC1902057).

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 444–452, 2020.
https://doi.org/10.1007/978-3-030-65390-3_34

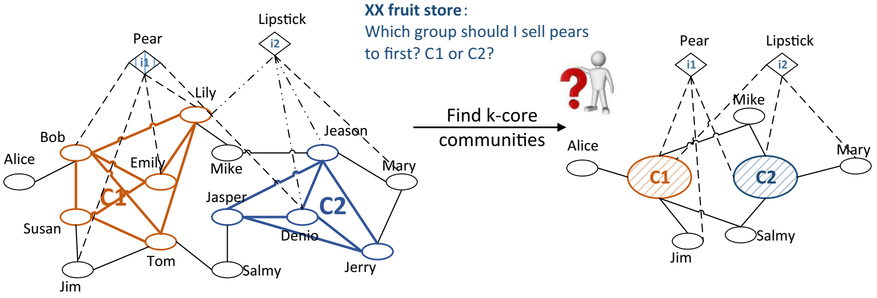


Fig. 1. Illustration for social networks based on group recommendation.

to buy the pears recommended to the candidate group C_1. For item providers, our goal is to find out which group should recommend to i_1 first.

Challenges and Contributions. What makes our work different from others is that we are the first to consider the impact of the close relationship of users within the group on the group's preference and recommend groups to item providers instead of recommending items to groups. In this paper, we think about how to build an appropriate group to make the item more relevant to the preferences of users in the group. On the other hand, we also need to consider how to represent the preferences of a group accurately. To this end, we proposes a new solution, called Neighbor-aware Group Recommendation Network (NGRN). Specifically, we first find the groups that meets the criteria on social network, any user in the group has at least k neighbors. Secondly, we use GCN to model all the groups and generate a characteristic representation of the groups, so as to obtain the optimal groups recommendation sequence. And we verify the validity of this model on two public datasets.

Related Work. Most of the existing group recommendation algorithms are based on the conventional collaborative filtering method [2,5], but with the widespread application of neural network in various fields, deep learning technology has been gradually applied to group recommendation [13]. In [8], Ntoutsi et al. divide all users in the whole database into several groups by hierarchical clustering method, then the collaborative filtering (CF) method is used to recommend items to users with similar preferences in group. In [9], Ntoutsi et al. study a group recommendation method based on subspace partitioning.

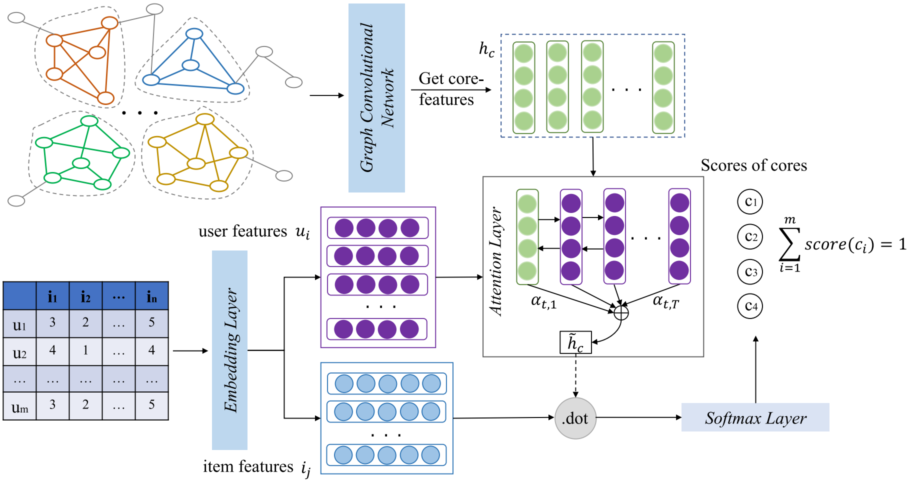
In [1], the authors add the concepts of correlation and divergence to the CF model, believing that differences in preferences among members of the group for each item were inevitable. [15] proposes the probability model COM for the generation process of group activities. The model takes into account the different weights of users in the group to make recommendations.

Because of its ability to capture nonlinear features, neural network has been widely used in recommendation system [4,7]. ATT-AVG is an effective method to combine neural network and attention mechanism on group recommendation

[3]. In addition, others have proposed a method of group recommendation by exploring the connection between group interests and group users [10].

2 Neighbor-Aware Group Recommendation Network

In this section, we elaborate the Neighbor-aware Group Recommendation Network (see Fig. 2) and its optimization process in detail. First, we extract the group and obtain the eigenvectors of the groups. Then the user characteristics are used to enrich the group characteristics. Finally, we predict the probability of the groups that is recommended to each item.



	i1	i2	…	in
u1	3	2	…	5
u2	4	1	…	4
…	…	…	…	…
um	3	2	…	5

Fig. 2. The framework of the proposed NGRN method.

2.1 Problem Definition

Our work aims to predict which group is most likely to buy an item. Let $G = (V, E)$ denote the user-user social interaction network, V is the set of users and $(u, v) \in E$ is the edge between u and v when the users u and v are neighbors. We use k-core algorithm to find the candidate user groups. $C = \{c_1, c_2, \cdots, c_m\}$ is the set of candidate groups, m is the number of groups. Let $I = \{i_1, i_2, \cdots, i_{|I|}\}$ denotes the set of items, we represent a user's rating of an item with a triple $\tau = (u, i, rating)$, an observed interaction represents an rating operation, where $|I|$ is the number of items. In our work, for the item i, the output of the model is the recommendable probabilities $\hat{\mathbf{r}}$ for all possible groups, where an element value of vector $\hat{\mathbf{r}}$ is the recommendation score of the corresponding groups.

2.2 Extracting K-Core Groups

Taking the close relationship between users as an important factor influencing group preferences, we extract k-core groups of social network. *K-core* is the largest group of entities, all of which are connected to at least k other entities in the group and are independent of each other. We extracted all k-core groups from the user's social network as candidate groups for our item recommendation. Each group c_i can be modeled as a undirected graph $G_{c_i} = (V_{c_i}, E_{c_i})$, which is a subgraph of G.

We count the labels of all items that the user u_i has rated as user's label l_{u_i}. According to the labels of each user in the obtained group, we want to get the ground-truth labels of the whole group. We first calculate the users' average-label-number $M = |\frac{1}{N} \Sigma_i Count\,(l_{u_i})\,|$ in the group, then take the top-m labels as the ground-truth feature vector $\mathbf{r_c} \in \mathbf{R}^m$ of the group by completing labels.

2.3 Generate Groups' Embedding

After the candidate groups are extracted, we further generate the eigenvectors of the group. The proposed NGRN algorithm can capture more property information on the basis of preserving the structure information of the graph. For the sake of completeness, we briefly explain how to get the latent vectors for groups via GCN [6]. Given a graph G_{c_i}, the nodes and edges in the graph can be aggregated as follows:

$$\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N \tag{1}$$

$$\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij} \tag{2}$$

$$\mathbf{N} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X} \boldsymbol{\Theta}\right) \tag{3}$$

where $\mathbf{D} \in \mathbf{R}^{S \times S}$ is the degree matrix and $\mathbf{A} \in \mathbf{R}^{S \times S}$ is the adjacency matrix of graph G_{c_i}, for a node $u \in G_{c_i}$, Eq. (2) denotes the updated node degree matrix of u, $\sigma\,(.)$ is ReLU activation function, $\mathbf{X} \in \mathbf{R}^{S \times C}$ is the input graph signal matrix, $\boldsymbol{\Theta} \in \mathbf{R}^{C \times d}$ is the corresponding graph filter with d filters and $\mathbf{N} \in \mathbf{R}^{S \times d}$ is the convolved signal matrix.

$\mathbf{N} \in \mathbf{R}^{S \times d}$ is actually the matrix made up of a single feature of all the nodes in the group. We want to use a global vector to represent the whole group, we simply mean the representation of all the nodes on the graph G_{c_i}. In this way, we get the eigenvector $\mathbf{h}_{c_i} \in \mathbf{R}^d$ for G_{c_i}. Relevant information of the scoring matrix is fed into the neural network, and feature vectors of users and items $\mathbf{u} \in \mathbf{R}^{N \times d}$, $\mathbf{v} \in \mathbf{R}^{I \times d}$ are realized through a embedding layer. If $\mathbf{u}_j \in \mathbf{R}^d$ is a neighbor of G_{c_i}, supposing that we regard a group as a special node and use $\mathbf{h}_{c_i}$ to represent the local embedding of the group. Then NGRN aggregates the local embedding with the feature vectors of all its neighbors to obtain the global vector $\tilde{\mathbf{h}}_{c_i}$ of the group by attention mechanism:

$$z_j = p^T \cdot tanh\left(\mathbf{W_1}\mathbf{h}_{c_i} + \mathbf{W_2}\mathbf{u}_j + \mathbf{b}\right) \tag{4}$$

$$\alpha_j = softmax\left(z_j\right) \tag{5}$$

$$\mathbf{h}_{c_i,u} = \sum \alpha_j \mathbf{u}_j \tag{6}$$

$$\tilde{\mathbf{h}}_{c_i} = \mathbf{W_3}[\mathbf{h}_{c_i,u}||\mathbf{h}_{c_i}] \tag{7}$$

where parameter $p \in \mathbf{R}^d$ and $\mathbf{W_1}, \mathbf{W_2} \in \mathbf{R}^{d\times d}$ denote the weights of node embedding vectors. Finally, we get the global embedding vector $\tilde{\mathbf{h}}_{c_i}$ of G_{c_i} by taking linear transformation over the concatenation of the local embedding vector and its neighbors' aggregated embedding vector, where $\mathbf{W_3} \in \mathbf{R}^{d\times 2d}$ maps the embedding vector to latent space $\mathbf{R}^d$.

The number of users in the group obtained in Sect. 3.2 is determined by the value k of core. In the NGRN model, we uniformly map the group and individual user's characteristics into the d-dimensional implicit space, so as to calculate the group's recommendation score according to each item.

2.4 Making Recommendation and Model Training

Thus, we are able to get all the group embedding vectors. For group c_i in set $C = \{c_1, c_2, \cdots, c_m\}$, this final representation $\tilde{\mathbf{h}}_{c_i}$ can be interpreted as the feature representation of the group of users, which can be easily matched with the item embedding to determine the recommendation score. More specifically, we can get the recommended score of item $\mathbf{v}$ by user group G_{c_i} as follows:

$$\hat{\mathbf{q}}_i = \mathbf{v}^T \tilde{\mathbf{h}}_{c_i} \tag{8}$$

$$\hat{\mathbf{r}} = softmax\left(\hat{\mathbf{q}}\right) \tag{9}$$

where $\hat{\mathbf{q}} \in \mathbf{R}^m$ denotes the recommendation scores over all candidate items, and $\hat{\mathbf{r}_c} \in \mathbf{R}^m$ denotes the groups' recommendable probability for item $\mathbf{v}$. $\sigma(.)$ is the activation function of the output layer, we get the output by softmax function. This function maps the input to [0, 1], with a guaranteed normalized sum of 1.

For each item, in order to make the model parameters more in line with our requirements, the loss function is defined as the cross-entropy of the prediction and the ground truth. It can be written as follows:

$$\mathcal{L}(\hat{\mathbf{r}}) = -\frac{1}{m}\sum_{i=1}^{m} \mathbf{r}_i log\hat{\mathbf{r}_i} + (1 - \mathbf{r}_i)log(1 - \hat{\mathbf{r}_i}) \tag{10}$$

Since each item has more than one label, we treat the multi-label classification problem as the dichotomy problem on each label, and the final loss function value is the average value of the loss value on all labels.

Finally, we use Back Propagation (BP) algorithm to train our proposed NGRN model. In addition, in order to prevent model overfitting, *Dropout* is applied during the training phrase of the model to improve the generalization ability of the proposed NGRN.

3 Experiments and Analysis

In this section, we first explain the datasets, baseline algorithms, and evaluation metrics used in the experiments. Then, we compare the proposed NGRN with other comparative methods. Finally, we make detailed analysis of NGRN under different experimental settings.

Datasets. We conduct experiments on three real datasets, *Flickr*[1] dataset, *MovieLens (1M)* and *MovieLens (10M)*[2] datasets. Since there is no specific user rating on Flickr, we use the items and rating data on the Movielens dataset as the rating data in our model. In our experiment, we only keep the number of users in line with the number of Movielens, regardless of the number of users and neighbors beyond the scope of the experimental dataset. The statistics of datasets are summarized in Table 1.

Table 1. Statistics of datasets used in the experiments

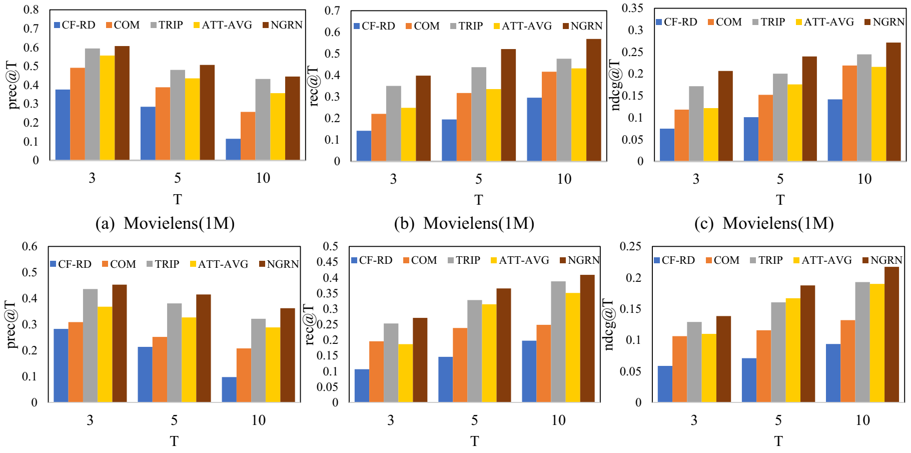
	MovieLens (1M)	MovieLens (10M)
Number of users	6040	71568
Number of movies	3952	10681
Number of user-links	580989	7455842
Rating parsity	4.17%	1.39%

Evaluation Metrics. We evaluate model performance with these three evaluation metrics: Precision ($prec@T$), Recall ($rec@T$), and Normalized Discounted Cumulative Gain (NDCG) ($ndcg@T$). Here T is the number of recommendations. We evaluate recommendation accuracy with T={3, 5, 10}. We take the average of all the test item measures as the final measure.

Baseline Algorithms. We compare our proposed model NGRN with four state-of-the-art baselines: User-based (CF-RD), COM, TRIP, ATT-AVG. User-based (CF-RD) [1] method makes recommendations by integrating user relevance and difference. COM [15] models the generation process of group activities according to probability model and makes group recommendation. TRIP is a hybrid recommendation method that combines user configuration with rating information to make recommendations. ATT-AVG represents group characteristics by the characteristics of all users in the aggregation group.

Parameter Settings. For COM and ATT-AVG, we still make their hyperparameters as default. All parameters in our model are initialized with a Gaussian distribution with mean of 0 and standard deviation of 0.1, we use Adam optimizer to improve the performance of NGRN.

[1] http://networkrepository.com/.
[2] https://grouplens.org/datasets/movielens/.



(a) Movielens(1M) (b) Movielens(1M) (c) Movielens(1M)

(a) Movielens(10M) (b) Movielens(10M) (c) Movielens(10M)

Fig. 3. Performance comparison of group recommendation methods on two datasets.

Table 2. The performance of each model is compared on two datasets when T = 5.

	MovieLens (1M)			MovieLens (10M)		
	prec@5	*rec@5*	*ndcg@5*	*prec@5*	*rec@5*	*ndcg@5*
CF-RD	0.284932	0.194737	0.101053	0.213699	0.146053	0.070790
COM	0.389041	0.317808	0.152105	0.251781	0.238356	0.115788
TRIP	0.481081	0.437838	0.200517	0.380811	0.328379	0.1603878
ATT-AVG	0.435725	0.335944	0.175960	0.326794	0.314458	0.166970
NGRN	**0.507286**	**0.521849**	**0.239932**	**0.415474**	**0.365226**	**0.187656**

3.1 Overall Performance Comparison

This subsection compares the recommendation results from NGRN to those from the baseline models. Figure 3 report the *prec@T*, *rec@T* and *ndcg@T* values for the two datasets with T = {3, 5, 10}. From the comparative data, we can observe that: (i) Compared with all the baseline methods, our proposed NGRN model always has the best performance. (ii) As shown as the result, group recommendations can be made as long as the users are related to each other, which reflects the flexibility of the model proposed in this paper. (iii) TRIP shows better performance across all the baseline algorithms. This is because TRIP put into effective use more information in its recommendations, including ratings and configuration information, than other baseline algorithms.

We observe that the results of TRIP are significantly better than those of the other baseline methods from Table 2. Taking TRIP's performance data as a reference, *prec@5*, *rec@5* and *ndcg@5* on *Movielens* (*1M*) of our model are superior to TRIP 5%, 19% and 20% respectively. On the basis of capturing user configuration information, NGRN uses convolutional network to mine more hidden information, leading to the best effectiveness.

4 Conclusions

In this paper, we present a new group recommendation method NGRN. The model can represent groups effectively, as users in the group have neighborhood relations, it greatly improves the trust of users in the group on the items recommended. In addition, we use the soft attention mechanism to enrich the characteristics of the group. Finally, we carried out extensive experiments on three real datasets to verify the validity of the model.

References

1. Amer-Yahia, S., Roy, S.B., Chawla, A., Das, G., Yu, C.: Group recommendation: semantics and efficiency. Proc. VLDB Endow. **2**(1), 754–765 (2009)
2. Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A., Leonardi, S.: Online team formation in social networks. In: Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, pp. 839–848. ACM (2012)
3. Cao, D., He, X., Miao, L., An, Y., Yang, C., Hong, R.: Attentive group recommendation. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, pp. 645–654. ACM (2018)
4. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805 (2018)
5. Ghazanfar, M.A., Prügel-Bennett, A.: The advantage of careful imputation sources in sparse data-environment of recommender systems: generating improved SVD-based recommendations. Informatica (Slovenia) **37**(1), 61–92 (2013)
6. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017, Conference Track Proceedings. OpenReview.net (2017)
7. LeCun, Y., Bengio, Y., Hinton, G.E.: Deep learning. Nature **521**(7553), 436–444 (2015)
8. Ntoutsi, E., Stefanidis, K., Nørvåg, K., Kriegel, H.-P.: Fast group recommendations by applying user clustering. In: Atzeni, P., Cheung, D., Ram, S. (eds.) ER 2012. LNCS, vol. 7532, pp. 126–140. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34002-4_10
9. Ntoutsi, E., Stefanidis, K., Rausch, K., Kriegel, H.: "Strength lies in differences": diversifying friends for recommendations through subspace clustering. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, pp. 729–738. ACM (2014)
10. Qin, D., Zhou, X., Chen, L., Huang, G., Zhang, Y.: Dynamic connection-based social group recommendation. IEEE Trans. Knowl. Data Eng. **32**(3), 453–467 (2020)
11. Tang, J., Hu, X., Liu, H.: Social recommendation: a review. Soc. Netw. Anal. Min. **3**(4), 1113–1133 (2013). https://doi.org/10.1007/s13278-013-0141-9

12. Yang, X., Wang, B., Yang, K., Liu, C., Zheng, B.: A novel representation and compression for queries on trajectories in road networks. IEEE Trans. Knowl. Data Eng. **30**(4), 613–629 (2018)
13. Yang, X., Wang, Y., Wang, B., Wang, W.: Local filtering: improving the performance of approximate queries on string collections. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data 2015, pp. 377–392. ACM (2015)
14. Yu, F., Zeng, A., Gillard, S., Medo, M.: Network-based recommendation algorithms: a review. CoRR abs/1511.06252 (2015)
15. Yuan, Q., Cong, G., Lin, C.: COM: a generative model for group recommendation. In: The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014, pp. 163–172. ACM (2014)

Cross Product and Attention Based Deep Neural Collaborative Filtering

Zhigao Zhang[1,2], Jing Qin[1], Feng Li[1], and Bin Wang[1](✉)

[1] School of Computer Science and Engineering, Northeastern University,
Shenyang 110004, China
zzgtongxin@163.com, binwang@mail.neu.edu.cn

[2] College of Computer Science and Technology, Inner Mongolia University
for Nationalities, Tongliao 028000, China

Abstract. Matrix factorization and its subsequent models have been widely used in recommendation systems due to their simple and efficient performance. However, this simple and intuitive operation has its natural limitations, which limit performance improvements. The reason is that it assumes that the embedded dimensions of the user and item are independent and identically distributed (IID), and that each dimension contributes equally to the predicted score. To overcome these limitations, we propose a new deep neural network architecture CADNCF to perform ranking recommendation. The idea is to use the cross product to build a two-dimensional correlation matrix, which can not only explicitly model the pairwise correlations between the dimensions of the embedded space, but also be more expressive and semantic. We also designed an attention-mechanism learning module to extract these useful information from the correlation matrix for the final prediction and eliminate noise. Then, we adopt the MLP to learn their interactions function and make predictions. We conducted extensive experiments on three benchmark data sets, and the results show that our proposed CADNCF model is superior to some baseline methods and other sate-of-the-art methods.

Keywords: Recommender system · Collaborative filtering · Deep neural network · Attention mechanism

1 Introduction

In an age of data explosion, recommendation technology has become the key technology of information retrieval and push. Matrix factorization (MF) has become the mainstream of recommendation research due to its superior performance over others [3]. The basic principle of MF is that the user and item

The work is partially supported by the Fundamental Research Funds for the Central Universities (No. N171602003), and Liaoning Distinguished Professor (No. XLYC1902057).

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 453–461, 2020.
https://doi.org/10.1007/978-3-030-65390-3_35

are represented by two low-dimensional vectors(also termed embedding), use the inner product as the interaction function of them.

Despite the effectiveness of MF and many subsequent developments, we argue that MF has its natural limitations due to its use of a data-independent and fixed inner product as an interaction function between the embedding of the users and items. It actually supposes that the embedding dimensions of the users and items are independent and identically distributed (IID) and that the interactions between them are linear. Meanwhile, it assumes that each embedding dimensions has equal effect on the prediction of all data points. We point out that these assumptions did not conform to the real application scenario, since each embedded dimension can be interpreted as an attribute, but in reality the attributes of things are not necessarily independent [9]. Moreover, the inner product was insufficient to capture the complex structure of user interaction data that has rich yet complicated patterns. In addition, the user's focus on some attributes of the item is dynamic, which means that each dimension of the item embedding contributes differently to the prediction of the user's preferences.

In order to solve the above problems, we propose a new method to integrate the pairwise correlations between embedded dimensions into the model. We use the cross product to create a pairwise correlation matrix. This correlation matrix has a high applicability for CF tasks because it also includes the interaction signals used in MF, and also includes all other paired correlations.

Our main contributions are as follows: (1) A new model CADNCF is proposed to implement the ranking recommendation task. (2) We designed an attention-mechanism learning module that can transform the abstract correlation matrix into the most useful information for the final prediction. (3) We conducted a large number of experiments on three real-world data sets. The experimental results show that the model proposed in this paper is superior to the state-of-the-art baseline methods.

2 Related Works

Recently, incorporate deep learning to recommendation system has become the main trend of current research, such as Xue *et al.* proposed a novel model DMF [8], He *et al.* proposed a neural based CF model NCF [3]. Since cold start and sparsity are important factors that restrict the recommendation performance, to solve these problems, many researchers integrate a large number of auxiliary information into the recommendation system, and utilizes deep learning to automatically extract the effective representations.

Attention mechanism is an important technology to pay special attention to and extract important information. Adding attention mechanism module into a deep learning model to improve the overall performance of the model has become a successful paradigm. Now, the attention mechanism is gaining popularity in recommendation tasks as well. For example, He *et al.* proposed a NAIS model [2] to implement item-based collaborative filtering.

In this paper, we extract the complex and rich semantic relationship between users and projects with the help of deep neural network. What is more relevant to our work is literature [1]. However, different from the CNN network used in [1], we effectively extracted the user's preference information through the self-attention model, which eliminated the influence of noise and greatly reduced the complexity of the model.

3 Preliminaries

For a given recommendation question, suppose there are M users and N items. Let $R \in \mathbb{R}^{M \times N}$ represents the rating matrix, in which each entity R_{ij} is a rating of user i on an item j. We construct the user-item interaction matrix $Y \in \mathbb{R}^{M \times N}$ from R as follows:

$$Y_{ij} = \begin{cases} R_{ij}, & \text{if a user } i \text{ rating an item } j \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

here, the explicit rating R_{ij} indicates the user's preference degree for an item. If the evaluation is unknown, it is marked zero, we term it non-preference implicit feedback. We still use Y_{i*} represents the i_{th} user's ratings across all items, and use Y_{*j} represents the j_{th} item's ratings across all users.

In this paper we focus on the recommended tasks for Top-N ranking. We followed the methodes in [6], which use a latent factor model(LFM) to generate all ratings, the approach as follows.

$$\hat{Y}_{ij} = F_{LFM}(u_i, v_j | \Theta) \tag{2}$$

here $\hat{Y}_{ij}$ represents the predicted score of interaction Y_{ij}, Θ representatives all the parameters of model, F_{LFM} representatives a map function, which can map all parameters to the predicted scores.

4 Proposed Methods

4.1 Overall Framework

In this section, we present the four components of our model in detail, the overall framework of the model is shown in Fig. 1.

Input and Embedding Layer. we make Y_{i*},Y_{*j} as the original denotations of the user u_i and the item v_j respectively, then fed them to two fully connected multi-layer neural networks simultaneously, output two low-dimensional vectors p_i and q_j, the intermediate hidden layers by h_{Ui} and $h_{Vi}, i = 1, 2, ...N - 1$, so we have

$$h_{U1} = W_{U1} Y_{i*} + b_{U1}$$
$$h_{V1} = W_{V1} Y_{*j} + b_{V1}$$

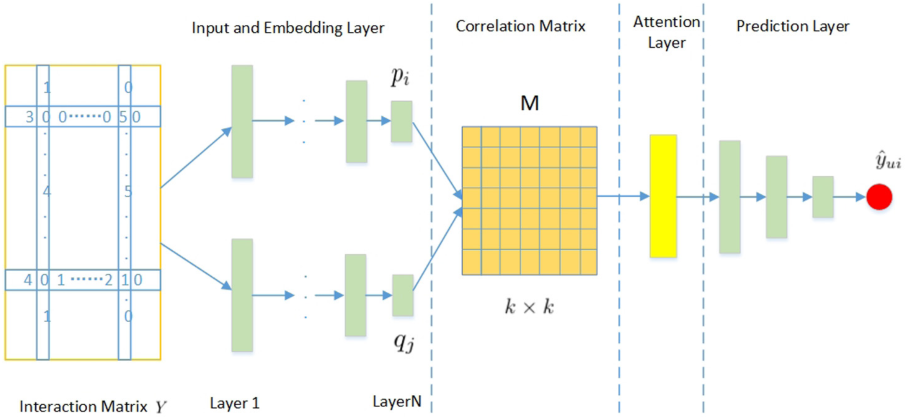


Fig. 1. The overall framework of the model consists of four parts: Input and Embedding Layer, Correlation Matrix, Attention Layer and Prediction Layer.

$$\begin{aligned} p_i &= \delta(W_{Ul-1} h_{Ul-1} + b_{Ul}), \quad l = 2, ...N \\ q_j &= \delta(W_{Vl-1} h_{Vl-1} + b_{Vl}), \quad l = 2, ...N \end{aligned} \tag{3}$$

where, W_{Ui}, W_{Vi} are the ith layer weight matrix for U and V, respectively, then b_{Ui}, b_{Vi} are the ith layer bais term, $p \in \mathbb{R}^K, q \in \mathbb{R}^K$ represent the final embedding of u_i and v_j respectively, K is the final embedding size($K \ll M, K \ll N$), δ denote the activation function. Herein, the ReLU is used as the activation function.

Correlation Matrix. For the latent factors of p_i and q_j obtained by Eq. 3, we use the cross product operation to construct their correlation matrix M.

$$M = p_i \otimes q_j = p_i q_i^T \tag{4}$$

M is a square matrix with size $k \times k$. We believe that using the correlation matrix M in our model has the following advantages: (1) This method contains richer coding information than MF by considering the correlation between different embedding; (2) The correlation matrix is more abundant and reasonable in the expression of interaction relationship and semantics than the simple concatenation; (3) It's an extension of the MF model, which only considers diagonal elements in our correlation matrix.

Attention Layer. we use self-attention [10] mechanisms to encode useful interaction characteristics from the correlation matrix. Here, we regard the correlation matrix M as a matrix composed of multiple word vectors, so the ith row M_i can be viewed as a word vector. We define Q denote the set of all query vectors, make K and V to denote the sets of all key vectors and value vectors respectively, and then all query, key and value vectors are expressed in the form of matrix as follows:

$$Q = M \times W_Q, K = M \times W_K, V = M \times W_V$$

$$Z = softmax(\frac{Q \times K^T}{\sqrt{d_k}})V \tag{5}$$

where, the input consists of three parts: queries $Q \in \mathbb{R}^{d_k}$, keys $K \in \mathbb{R}^{d_k}$ and values $V \in \mathbb{R}^{d_V}$. $W_Q \in \mathbb{R}^{k \times d_k}, W_K \in \mathbb{R}^{k \times d_k}, W_V \in \mathbb{R}^{k \times d_V}$ are the appropriate parameters learned during our model training. The outputs matrix is $Z \in \mathbb{R}^{k \times d_v}$, which can be seen as a new representation of the correlation matrix M transformed by self-attention.

Prediction Layer. We will use a multi-layer perceptron module to learn the nonlinear preference model based on the final representation obtained from the attention layer. The matrix representation will be flattened into a vector, and then fed into a fully connected multilayer MLP to predict the user's rating score as follows:

$$h_0 = flatten(Z)$$

$$\hat{y}_{ui} = ReLU(W_l h_{l-1} + b_l), \;\; l = 1, ...N \tag{6}$$

4.2 Loss Function

In this paper, we follow the loss function in DMF model [8], in which explicit rating is incorporated into cross entropy loss, the formula is shown in Eq. 7.

$$L = - \sum_{(i,j) \in Y^+ \cup Y^-} (\frac{Y_{ij}}{max(Y)} log \hat{Y}_{ij} + (1 - \frac{Y_{ij}}{max(Y)} log(1 - \hat{Y}_{ij})) \tag{7}$$

where, the $max(Y)$ (in our work it is 5) is the max score in all ratings, which is used for normalization.

5 Experiments

5.1 Experiment Settings

Datasets and Evaluation Metrics. We select three benchmark datasets from the recommended tasks as data sources for our model performance tests, which are MovileLens100K(ML_100K), MovileLens1M(ML_1M) and Delicious. The Hit ratio and Normalized Discounted Cumulative Gain, which are most effective and widely used in ranking tasks, are used as the metrics to evaluate the performance of the model [11]. In this article, we set the N for HR@N and NDCg@N to 10.

5.2 Performance Comparison

In order to directly present the effectiveness of our model performance, we compare CADNCF with the following baseline methods. (1) **eALS** This is a MF-based model with the least square method optimize the parameters [4]. (2) **ItemPop** It ranked the items by the number of interactions with users and their popularity degree. [7]. (3) **NCF** It use cross entropy loss optimize the parameters of model [3], which represents the state-the-art neural collaborative filtering methods. (4) **ItemKNN** This is a collaborative filtering model using implicit feedback, which is a item-based model and the earliest recommendation algorithm applied to Amazon business [5]. (5) **DeepMF** It is a MF model with neural network architecture for Top-N recommendation tasks, which use deep neural network to learn latent representation to users and items [8].

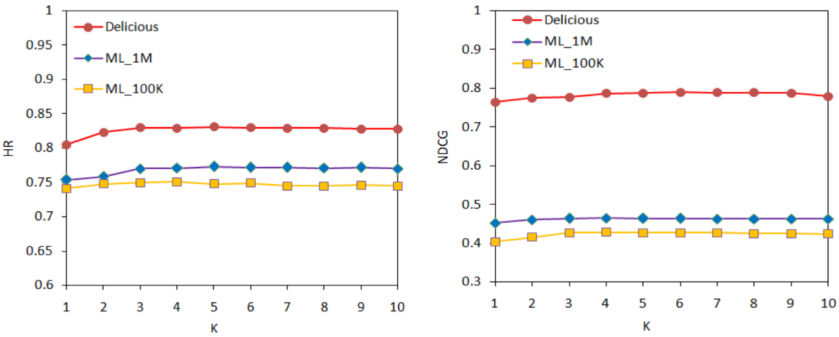
Table 1. Comparison of evaluation results of different methods for NDCG@10 and HR@10

Datasets	Metrics	ItemPop	ItemKNN	eALS	NCF	DeepMF	CADNCF
ML_100K	HR	0.406	0.600	0.621	0.670	0.687	0.751
	NDCG	0.231	0.334	0.356	0.395	0.409	0.428
ML_1M	HR	0.472	0.637	0.709	0.731	0.732	0.773
	NDCG	0.263	0.372	0.425	0.448	0.451	0.464
Delicious	HR	0.104	0.687	0.676	0.788	0.793	0.831
	NDCG	0.048	0.688	0.617	0.781	0.784	0.790

According to the comparison results in Table 1, it can be seen that our model achieved the best performance on NDCG and HR evaluation criteria in the three data sets. This also directly proves the effectiveness of our proposed method. Compared with the three baseline methods of ItemKNN, eALS and ItemPop, the performance of our model was significantly improved. Compared with the two most advanced models based on neural network, NCF and DMF, the performance of HR@10 was improved by 7.33% and 6.0% on average, and that of NDCG@10 improved by 4.0% and 2.5%, respectively.

5.3 Sensitivity of Hyper-parameters

Negative Sampling Ratio. As shown in algorithm 1 in Sect. 4, in order to better train the model, we selected some data for each user as negative instances, which were generated by $neg - ratio$ sampling from data that had never been interacted with. In the experiment, we set the value of $neg - ratio$ of the model from 1 to 10, tested it on three datasets, and observed and recorded the changes of model performance. The result of the recording is shown in Fig. 2. According to the recorded results, it can be found that, in general, the performance tends to increase with the increase of $neg - ratio$. However, it is not that the greater the value of $neg - ratio$ is, the better the performance of the model will be. For HR, the value of $neg - ratio$ with peak performance on the three data sets is 5, 5, 4 respectively. For NDCG, the optimal value of $neg - ratio$ is 4.



(a) HR@K

(b) NDCG@K

Fig. 2. The HR and NDCG results of our model with different neg-ratio in three datasets

Factors of the Final Embeddings Size. The final embeddings denote the representations of the user and item obtained in the Input and Embedding Layer. The dimension size of the final embeddings determines the semantic representation ability of the correlation matrix and is also an important factor affecting the performance of our model. We set the number of the dimension size from 8 to 128, the performance variance are shown in Table 2. For ML_100k and Delicious, when the final embedding size is 64, our model achieves optimal performance, but, for ML_1M, the best size is 32.

Table 2. The effect of the final embedding size on the performance of our model

Datasets	Metrics	8	16	32	64	128
ML_100K	HR	0.726	0.739	0.746	0.751	0.750
	NDCG	0.401	0.418	0.423	0.428	0.426
ML_1M	HR	0.758	0.764	0.773	0.770	0.768
	NDCG	0.424	0.431	0.464	0.458	0.453
Delicious	HR	0.814	0.820	0.824	0.831	0.827
	NDCG	0.747	0.753	0.778	0.790	0.787

6 Conclusion

In this paper, we propose a new neural network model to complete the task of recommending user Top-N items. In the model, we make full use of explicit rating information and implicit non-preference information, and use the neural network to learn the dense low-dimensional embedded representation of users and items. The cross product operation is used to establish their correlation matrix, which has rich semantic expression ability. In order to enhance the information that is useful for the final prediction and eliminate the noise interference, we designed an attention model and finally used the multi-layer perceptron to learn the interaction function to complete the prediction. Experimental results show that our method achieves better performance on three benchmark data sets. In the future work, we will introduce some auxiliary data such as social information and comment information into the model to solve the sparsity and cold start problems of the model data.

References

1. He, X., Du, X., Wang, X., Tian, F., Tang, J., Chua, T.: Outer product-based neural collaborative filtering. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI, pp. 2227–2233. ijcai.org (2018)
2. He, X., He, Z., Song, J., Liu, Z., Jiang, Y., Chua, T.: NAIS: neural attentive item similarity model for recommendation. IEEE Trans. Knowl. Data Eng. **30**(12), 2354–2366 (2018)
3. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.: Neural collaborative filtering. In: Barrett, R., Cummings, R., Agichtein, E., Gabrilovich, E. (eds.) Proceedings of the 26th International Conference on World Wide Web, WWW, pp. 173–182. ACM (2017)
4. He, X., Zhang, H., Kan, M., Chua, T.: Fast matrix factorization for online recommendation with implicit feedback. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR, pp. 549–558. ACM (2016)
5. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), pp. 263–272. IEEE Computer Society (2008)
6. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434. ACM (2008)
7. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 452–461. AUAI Press (2009)
8. Xue, H., Dai, X., Zhang, J., Huang, S., Chen, J.: Deep matrix factorization models for recommender systems. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI, pp. 3203–3209. ijcai.org (2017)
9. Yang, X., Li, C.: Secure XML publishing without information leakage in the presence of data inference. In: (e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, VLDB 2004, pp. 96–107. Morgan Kaufmann (2004)

10. Yang, X., Wang, B., Yang, K., Liu, C., Zheng, B.: A novel representation and compression for queries on trajectories in road networks. IEEE Trans. Knowl. Data Eng. **30**(4), 613–629 (2018)
11. Yang, X., Wang, Y., Wang, B., Wang, W.: Local filtering: improving the performance of approximate queries on string collections. In: Sellis, T.K., Davidson, S.B., Ives, Z.G. (eds.) Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp. 377–392. ACM (2015)

Privacy and Security

Blockchain-Based Privacy Preserving Trust Management Model in VANET

Ruochen Liang(✉), Bohan Li, and Xinyang Song

Nanjing University of Aeronautics and Astronautics, Nanjing, China
531113463@qq.com, liangruochen@sina.com, bhli@nuaa.edu.cn

Abstract. With the development of the intelligent traffic system (ITS), vehicular ad hoc network (VANET) has been widely used as an important part of ITS. Location based service (LBS) is such a significant application of VANET. But how to protect vehicles' data and privacy security has not been well solved. We propose a trust management model based on blockchain. In our scheme, we use blockchain to realize the data security of vehicles, and we give a trust management algorithm to construct anonymous cloaking region to protect the privacy security. We use various datasets to simulate the model. And the experimental results show that our model can resist most common trust attacks and greatly reduce the possibility of vehicle privacy disclosure.

Keywords: Location privacy security · Blockchain · Trust model · VANET

1 Introduction

Based on mobile vehicles and traffic infrastructure, VANET is a vehicle mobile communication network which uses wireless communication technology [1]. The vehicles communicate with traffic infrastructure and other vehicles through on-board equipment. We take the LBS as an example. Through collecting and analyzing the location information, the vehicle can plan the optimal driving route and improve the traveling efficiency. However, VANET is vulnerable to a variety of attacks in LBS. So far, some studies make use of secure communication channels to prevent attackers from launching attacks from the outside. But the problem of trust security within the VANET has not been solved [2, 3]. Malicious nodes may collect and investigate the vehicles' personal privacy during communication, such as driving habits and scope of activities.

VANET privacy-preserving usually guarantees two aspects of security: personal information security and data security. K-anonymous algorithm is one of the most popular personal information protection algorithms, which has small computational overhead and accurate query results. K-anonymous algorithm can be divided into two categories: centralized and distributed [4]. The former uses a trusted central server to store users' personal information. However, single point failures can destroy the whole system and performance bottlenecks also limits the scalability of the model. The latter overcomes above problems, but the trust problem between users occurs. Hence, we introduce trust management to solve trust crisis. Blockchain is a widely used data security technology, which

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 465–479, 2020.
https://doi.org/10.1007/978-3-030-65390-3_36

ensures the consistency and non-tampering of data. In our model, we use k-anonymous trust management and blockchain to protect VANET privacy. The contribution of this paper is as follows:

1) We use digital certificates to replace pseudonyms in k- anonymous algorithm. The certificates ensure the legitimacy of users' identity and requests without revealing the real information of users. At the same time, we adopt the blockchain to record the certificates, which enables RA to trace malicious users.
2) When constructing the anonymous cloaking region in k- anonymous algorithm, we use RSUs to dominate the construction instead of the direct communication between vehicles. This reduces the probability of privacy leakage and computing burden significantly.
3) We combine the trust management and distributed k-anonymous algorithm. Experiments indicate that our model can protect the privacy of users effectively, and our scheme has strong resistance to classical trust attacks.

2 Related Work

One of the first distributed k-anonymous scheme was proposed by Chow et al. [5]. Their scheme recommended request vehicles construct anonymous cloaking regions with at least k-1 nearby users by point-to-point communication. However, it requires request vehicles must wait passively until k-1 cooperator positions are received. So Ghaffari et al. proposed a P^4QS privacy protection query service scheme [6], which allows anonymous agents to generate false locations to replace the cooperator positions. But none of these algorithms take into account the problem of mutual trust between vehicles, so Bin Luo et al. proposed a reputation system which is evaluated by vehicles each other [7]. But the processing of malicious evaluation is limited, so regions cannot protect the privacy of vehicles effectively.

The research on distributed trust management system was first put forward normally by M. Raya et al. in 2008 [8]. Then Z. Li and C. T. Chigan proposed a model which combines privacy-preserving with trust values in 2014 [9]. With the development of blockchain, Ao. Lei et al. introduced the blockchain into the trust management in 2017 [10]. And in 2018, Zhe Yang et al. proposed the first blockchain-based distributed trust management system [11]. It has many problems, such as using the distance from the place of occurrence as a single measure of the credibility of information.

3 Description of Problem and Model

3.1 System Model

The architecture of our model is shown in Fig. 1. Then we will introduce the most important parts of our scheme in detail.

1) Registration Authority

RA main functions include vehicle registration and cancellation, dispute arbitration and blockchain for certificates (CerBC) maintenance [12]. When a vehicle enters the LBS, it should first register with RA. After authentication, RA will send a certificate containing the public key to the vehicle [13]. The certificate will be used as authentication, and the public key plays the role of pseudonym to realize the non-linkability between the user and the transactions. For further implementation of non-linkability between two transactions, vehicles need to change the public key on a regular basis. Besides, when there is a dispute of vehicles or RSUs, RA will make the judgment [14]. RA also maintains the public ledger of CerBC and record the certificates information in an open and transparent manner.

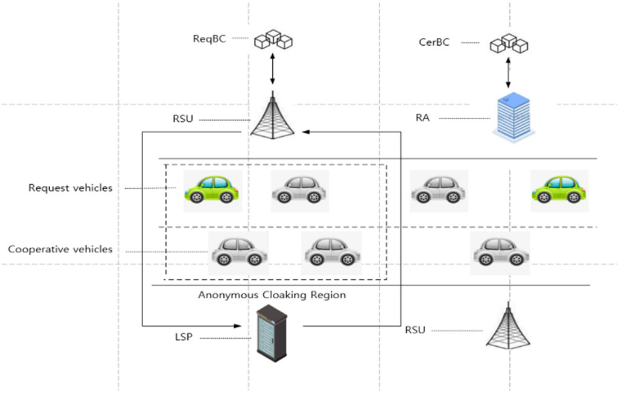


Fig. 1. Architecture of our model

2) Roadside Unit

RSU connect with the vehicle through a wireless channel. First, it accepts the vehicle's query request. Then it looks for the cooperative vehicles and constructs the anonymous cloaking region. Finally it will fetch the query results from the LSP and send results to the request vehicles. Besides, RSUs are also responsible for updating and maintaining the blockchain for requests (InfBC) [15].

3) Vehicles

Vehicles can initiate LBS requests as request vehicles or participate in the construction of anonymous cloaking region as cooperative vehicles. Although the vehicle is not directly involved in the creation and maintenance of the blockchain, it can supervise the transactions in the blockchain and initiate arbitration to the RA against undiscovered malicious behaviors in the system. All behaviors of the vehicles will be reflected in the trust values in the LBS [16].

4) Location-Based Service Provider

After receiving anonymous cloaking region and query requests from RSUs, LSP will retrieve the database and return the query results to RSUs [17]. The query algorithms for anonymous cloaking region already have a variety of solutions [18], we do not discuss them in this paper.

3.2 Blockchain Model

Our model maintains two blockchains: Blockchain for Certificates (CerBC) and Blockchain for Requests information (InfBC). The former is recorder by RSUs and all certificates registration, updating, and revocation transactions are stored in it. The latter records all query requests in the LBS, in which both the request vehicles and cooperate vehicles use pseudonyms [19, 20].

The consensus mechanism is the core mechanism of the blockchain. The traditional blockchain with PoW consensus mechanism is slow to block out, and waste resources. The emerging HotStuff consensus mechanism has the disadvantages of leading communication burden and system inertia. So this paper synthesizes two consensus mechanisms and proposes a hybrid consensus model of HotStuff + PoW.

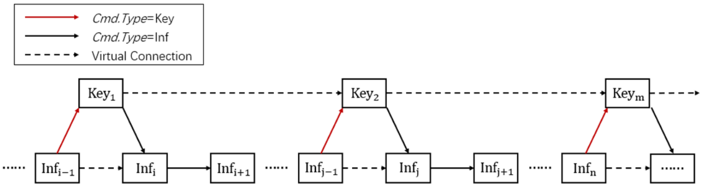


Fig. 2. Advanced structure of blockchain

The structure of the blockchain is shown in Fig. 2. Its physical structure is a complete uninterrupted chain. We divide it into two sub-chains by Cmd. Type, that is, the key chain containing the membership of the consensus committee and the Inf chain containing road traffic information and vehicle trust values.

Consensus committee members do not need to change frequently, but require most members be honest. So the key chain adopts a slower but safer PoW mechanism. The sub-chain structure of the key chain is shown in Fig. 3. PoW requires nodes to solve a mathematical problem based on the same nonce. Who solve the problem first can

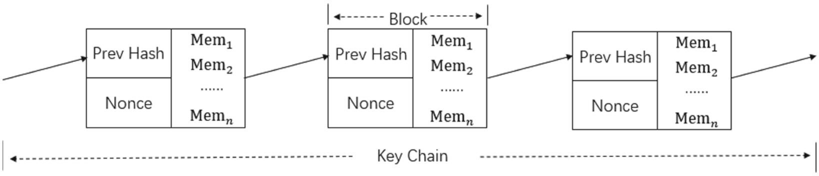


Fig. 3. Structure of key chain

generate a block containing the new consensus committee member list "mem" and link it to the blockchain. As a reward, this node can replace an old member in the committee.

As shown in Fig. 4, we adopt the HotStuff in the Inf chain, which is a Byzantine fault-tolerant consensus protocol [21]. It is a star-based point-to-point communication network, which is divided into two views. The first view uses three-stage point-to-point communication to reach a consensus. The second view changes the leader to reduce communication complexity and improve the consensus speed. The red line presents the message from client, while the blue ones mean the messages from the leader. To some extent, this network meets the requirements of some real-time systems.

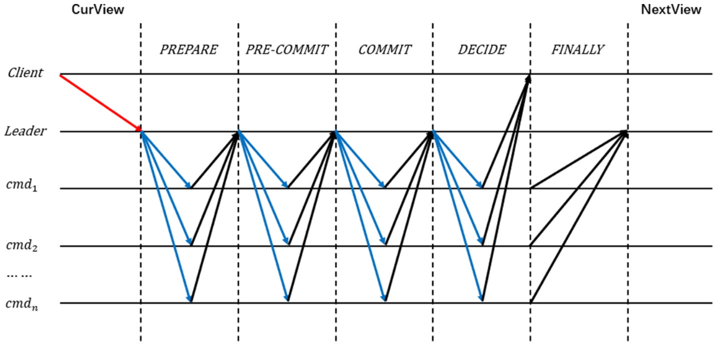


Fig. 4. Structure of basic HotStuff

4 Privacy Preserving and Trust Management

In this section we present the privacy preserving scheme and trust management algorithm.

4.1 Privacy-Preserving Authentication with Certificate Blockchain

Privacy-preserving authentication aims to build trust in V2V and V2I. In our system, RA is responsible for the function of certificate registration and certificate update using CerBC. We will introduce these two functions separately and give the process of privacy-preserving authentication.

1) Certificate Registration

When the vehicle *A* enters the LBS network, it should generate a pair of public-private keys first. And then *A* uses a secure channel to provide the RA with a material containing legal identity and public key. If the RA determines that the identity is legal, it will issue an initial certificate to *A* and add this certificate to the latest block in CerBC:

$$C_A =< ID_{CA}, PU_{CA}, Rpt_{CA}, T_{CA}, Tuple_{Hash} > \tag{1}$$

Where ID_{CA} is the certificate number, PU_{CA} represents the public key generated by *A* itself, and Rpt_{CA} is the initial trust value of the vehicle. The initial trust value can be the same or different according to the social credit, which is not within the scope of this paper. T_{CA} indicates the expiration time of the certificate and $Tuple_{Hash}$ stores the hash values available in authentication.

2) Certificate update

During authentication in LBS network, vehicles use public keys to replace their legal identity. So the public keys need to be updated frequently to avoid attackers destroying the non-linkability between public keys and legal identity through empirical attack. At the expiration of the certificate, vehicle A will regenerate a pair of public-private key pairs and send a certificate update request to the RA through a secure channel.

RA should make a proof-of-presence of certificate C_A in CerBC to prevent certificate forgery attack. RA also needs to check if the difference between the current time and the expiration time is greater than the preset value *Thr_time*. This is to avoid malicious vehicles send many certificate update requests to RA in a short time. After verification, RA sets the expiration time of the old certificate C_A to the current time first, which indicates that the certificate C_A has been revoked. Then RA queries the current trust value of A, the low trust value (less than Rpt_{cur}) will lose the qualification to update certificate. A will get new certificate if trust value meets the requirement and RA will add new certificate to CerBC.

3) Authentication Process

Authentication is divided into two parts, the first is the proof of legitimacy. The verifiers (possibly RSUs or RA) verify the coherence of the PU_{CA} and expiration time T_{CA}. If both two requirements are satisfied, the verifiers should proof the presence of C_A next. Merkle tree provides the proof of presence for verifiers, as shown in Fig. 5. If we want to proof the presence of the C_A, we can calculate root hash using the $Tuple_{Hash}$ (Hash8, Hash56, Hash14) recorded in C_A. If this value is equal to that recorded in CerBC, it indicates C_A is valid.

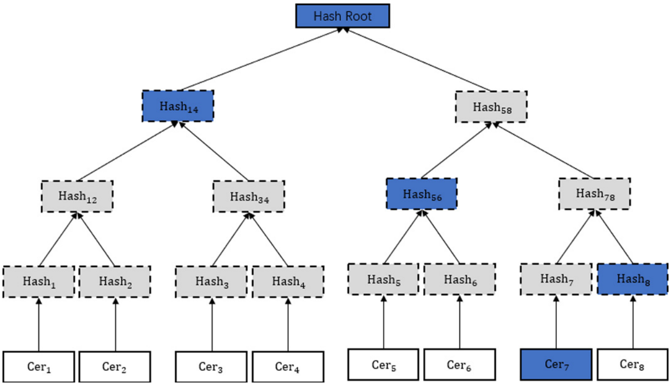


Fig. 5. The proof of presence

4.2 Trust Management Model

In this subsection, we use current behaviors of vehicles to evaluate whether the queries are reasonable or not. Then we give the calculation method of trust level using the historical trust information. Finally we show how to construct anonymous cloaking regions.

1) Evaluation of Vehicle's Behaviors

After verifying the certificate of vehicles, RSUs need to make a trust rating of vehicles according to their behaviors. The former sends the anonymous query requests to RSUs to obtain the target location information, so malicious request vehicles tend to send a large number of invalid queries to waste RSUs' and LSPs' computing resources. As a result, we will evaluate the rationality of the request vehicles' LBS queries from two aspects: query space rationality and query frequency rationality. While the latter provides RSUs with their own location information under the premise of protecting individual privacy. So malicious cooperative vehicles may provide false location information to influence the construction of regions. And we will evaluate them from the rationality and authenticity of location information.

2) Vehicle Trust Ratings Inspection

After evaluating behaviors of the vehicles, the corresponding trust rating will be made according to the evaluation results. And the specific classification criteria can be determined according to the actual scenes. In order to test whether the credit rating is reliable, we use v's historical trust rating to compare with it. The behaviors of vehicles in our scheme is divided into n grades, which are l_i, $i = 1,2,\ldots,$n. While the trust value represents the probability that the behavior is at the corresponding level. With the increase of i, the corresponding reliability of l_i also rises.

$$p(l_i) \in (\frac{i-1}{n}, \frac{i}{n}]\,(1 \le i \le n) \tag{2}$$

Specially, we define that $0 \in l_1$. To facilitate the assessment of vehicle's behaviors, the InfBC records the number of different evaluations received by all vehicles. Taking v_c as an example, assuming that this query is the *p-th* query in which v participated. The recorded historical trust information on the blockchain for v is $\overrightarrow{A_v^{p-1}} = (a_v^{1\_p-1}, a_v^{2\_p-1}, \ldots, a_v^{n\_p-1})$. If v receives a l_i rating from a RSU, then $a_v^{i\_p} = a_v^{i\_p-1} + 1$. The number of times related to other levels remains unchanged.

However, to prevent RSUs from being hijacked and give an unfair evaluation, we use the probability density function of the Dirichlet distribution to calculate. Suppose p (l_i) is the prior probability of v receiving a l_i rating, then there is

$$p(l_i) = E(p_v(l_i)|\overrightarrow{A_v^{p-1}}) \tag{3}$$

Where p_v (l_i) is the probability distribution that v receives a l_i rating, and its value refers to the proportion of l_i in a query, $\overrightarrow{P_v} = (p_v(l_1), p_v(l_2), \ldots, p_v(1_n))$. v's behaviors follow the Dirichlet probability density function distribution, so there is

$$f(\overrightarrow{P_v}|\overrightarrow{A_v^p}) = Dir(\overrightarrow{P_v}|\overrightarrow{A_v^p})$$

$$= \frac{\Gamma(\sum_{i=1}^{n} a_v^{i\_p})}{\prod_{i=1}^{n} \Gamma(a_v^{i\_p})} \prod_{i=1}^{n} p_v(l_i)^{a_v^{i\_p}-1} \tag{4}$$

The $E(p_v(l_i)|\overrightarrow{A_v^p})$ is

$$E(p_v(l_i)|\overrightarrow{A_v^p}) = \frac{a_v^{i\_p}}{\sum_{i=1}^{n} a_v^{i\_p}} \tag{5}$$

So, we can get

$$p(l_i) = E(p_v(l_i)|\overrightarrow{A_v^{p-1}}) = \frac{a_v^{i\_p-1}}{\sum_{i=1}^{n} a_v^{i\_p-1}} \tag{6}$$

if $p\ (l_i) \geq p\_thre$, then this rating is an accurate rating and RSU will record it into the blockchain.

3) Anonymous Cloaking Region Construction

Building an anonymous cloaking region is an important way to protect the privacy of vehicles. The details of it as follows.

Request vehicles v_r send LBS query request to nearby RSU as below

$$Req =< ID_{Cv_r},\ t_0,\ Sig_{SK-v_r}(C_{v_r}||I_{req}) > \tag{7}$$

Where ID_{v_r} is the v_r certificate ID, and RSU can find the public key according to it. t_0 is the timestamp of the request, $Sig_{SK-v_r}(C_{v_r}||I_{req})$ is the information signed with the v_r's private key, C_{v_r} is the v_r's certificate, I_{req} is the query content initiated by the v_r. "||" represents the union operation.

RSU will check the signature and see if the public key is consistent with the certificate first. And then it will verify the certificate validity, which includes certificate existence proof in the CerBC. If any of the above is not satisfied, a warning message is returned to the v_r.

RSU will evaluate the reasonability of the v_r query request if all the above conditions are satisfied. If the query request is unreasonable, RSU will return the error message to the v_r, and if the request is reasonable, go to the next step.

The higher the trust value of the v_r, the higher the trust value of the cooperative vehicle selected by the RSU when constructing the anonymous cloaking region. This is because vehicles with high trust value construct anonymous cloaking regions. And the privacy and credibility should be higher. When screening cooperative vehicle v_c, v_c's trust value $R_{v_c} \geq \sigma\ R_{v_r}$. where σ is the preset proportional coefficient.

If the RSU region is sparse and the number of cooperative vehicles satisfying the conditions is less than k, RSU will generate a virtual location complement k-anonymous cloaking region. Too many virtual locations usually means that LSP will consume a lot of computing resources to generate useless location information. but we only generate a small number of virtual locations in the case of sparse environment to ensure the credibility and privacy of anonymous cloaking regions, while also motivating vehicles to show honest behavior. Experiments show that the extra cost of generating a small number of virtual locations is negligible.

RSU send the anonymous cloaking region along with the query request to the LSP, and send the result of the query received from the LSP to the RSU, response format closest to the current location of the requested vehicle as shown below

$$Res =< ID_{Cv_r},\ t_s,\ Sig_{PU-v_r}(C_{v_r}||I_{res}) > \tag{8}$$

Where ID_{v_r} is the v_r certificate ID, t_s is the timestamp of the reply sent. $Sig_{PU-v_r}(C_{v_r}||I_{res})$ is the information signed with the v_r public key, I_{res} is the query result set to reply to the v_r.

4.3 Update the Trust Information on the Information Blockchain

When a RSU give the evaluation of vehicles, it will broadcast the results of the evaluation to the other RSUs by generating a transaction containing trust information. When the transactions in the network reach a certain amount, the current leader RSU will generate a new block containing these transactions. After all RSUs make a HotStuff consensus, the leader will update the InfBC blockchain. The details is shown below.

The leader RSU will collect the same evaluation for the vehicles. Assuming that v received a l_i evaluation s_i, then $a_v^{i\_p} = a_v^{i\_p-1} + s_i$. After v trust rating is updated, the v trust value will be recalculated. Suppose R_v is v's trust value.

$$R_v = \frac{\sum_{i=1}^{n} (a_v^{i\_p} * i)}{n} \tag{9}$$

if $R_v < Thr\_rpt$, then v will be identified as a malicious vehicle and will be expelled from the system when the certificate is updated.

In the traditional PoW, all nodes share the same threshold. All nodes get a hash value below the threshold by constantly modifying the nonce to calculate the hash value of the block, and the person who first computed this value becomes the miner.

The threshold of each RSU is different, we define S_i as the threshold for nodes with ID_{RSU} is i, and different RSUs have different thresholds, as follows:

$$Hash(ID_{RSU}, time, PreHash, nonce) \leq S_i \tag{10}$$

Where time represents the block generation time, and nonce is a random value that can make the upper expression hold. Finding nonce at the earliest is worthy of the RSU will win the election of the validation node and enter the consensus committee.S_i is a binary number beginning with multiple consecutive zero bites, and we assume that the number of digits of this binary number is N_m (the specific value of N_m varies according to the hash method adopted. For example, when the hash method of SHA-256 is adopted, the value of N_m is 256). The number of consecutive zero bits. At the beginning of this binary number is N_z.

$$S_i = 2^{N_m-N_z} - 1 \tag{11}$$

From the above formula, N_m is a fixed value, the smaller the N_z, the larger the S_i, the easier it is to find the qualified nonce value. So we can control the size of S_i by controlling the size of N_z, we define N_z as the number of times RA have judged an evaluation error.

Whenever there is a RSU to complete the PoW validation, the validation node that participates most in the consensus committee will leave the consensus committee and the RSU to complete the PoW validation becomes a new validation node. This process cannot be predicted in advance, and the permanent dynamic rotation is realized under the premise of ensuring security.

5 Experiments

5.1 Experiment Environment

We deployed our experiments in Hyperledger Fabric. As a permissioned blockchain infrastructure, Hyperledger Fabric is an open source technology platform. We have made the necessary changes to each layer according to our scheme. In the data layer, we modify the data structure of the transactions. We adopted the PoW + HotStuff consensus mechanism in the consensus layer.

Elliptic Curve Cipher (ECC) algorithm is one of the most popular encryption algorithm at present. It is simple and has high security. Our scheme uses ECC-secp256k1 to sign messages and use ECDSA-secp256k1 to verify them. The parameters used in the program and their values are shown in Table 1. In particular, the given values do not have essential effects on performance. The data samples used in the program are all from [22]. The experimental environment is: Intel (R) Core (TM) i5-8300HCPU @2.30 GHz, 8 GB, and the Operating System is Windows10.

Table 1. Parameters used in the program

Notation	Definition	Values
Thr_time	Time determination threshold	-60
Thr_rpt	Reputation determination threshold	10
m	The number of cells	$m = 25, n = 5$
ρ	Time delay rate	$\rho = 0.8, n = 10$
η_{c}	Rationality determination threshold	0.2
p_thre	Rating determination threshold	*p_thre* $= 0.1, n = 10$

5.2 Performance Analysis

Since we only store the hash value in block header, so the size of each block is about 80 bytes and the certificate size is about 100 bytes. We assume that the number of vehicles is n, then the overhead for proof of existence of certificates is 32 * $\log_2^{\mathrm{n}}$. If a block is generated every 10 min on average, the storage overhead as shown in Table 2 and the time consumption as shown in Table 3.

Table 2. Storage Overhead

Storage overhead (byte)/Number of vehicles	1000	10000	100000	1000000
Our scheme	485	605	712	818
Bin Luo et al.	851	1091	1330	1569
Lu et al.	873	1283	1670	2034

Table 3. Time consumption

Time consumption (ms)/Number of vehicles	1000	10000	100000	1000000
Our scheme	0.099	0.133	0.167	0.199
Bin Luo et al.	0.174	0.245	0.313	0.378
Lu et al.	0.182	0.268	0.408	0.485

5.3 Trends in Trust Values

It's obvious that the difference of the changing trend of vehicles' trust value compared with Bin Luo et al. scheme. We conducted 10 rounds of experiments, the first four-wheel vehicles of the experiment will show honest behaviors, the next three-wheel vehicles will show malicious behaviors, and the last three-wheel vehicles will show honest behaviors again. As shown in the Fig. 6, the red is the Bin Luo et al. scheme, after the continuous honest behaviors of the vehicle, the upward trend of trust value is still not slowed down, and after showing malicious behaviors, several honest behaviors of the vehicle makes the trust value rise rapidly again. If the scheme is under the on/off attack, it can quickly detect malicious behavior, but cannot immediately remove the malicious vehicle out of the system. Black is the Zhaojun Lu's scheme [23], the value of trust, whether rising or falling, its amplitude is smaller than the other two schemes. Especially after the vehicle continuously shows malicious behavior, the decline of trust value is slowed down instead, which is not consistent with our expectations.

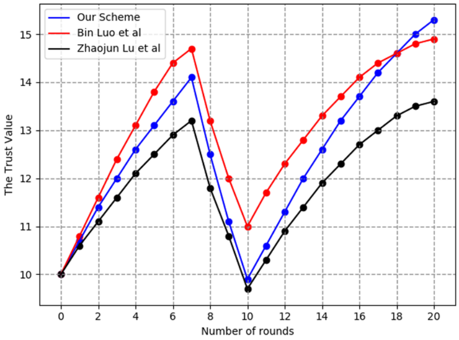


Fig. 6. Trust value tendency under on/off attack (Color figure online)

Our scheme is shown in the blue line in the Fig. 6. After several honest behaviors, the upward trend of trust values slows down. If the vehicle has a continuous malicious behavior, the trust value drops rapidly. And if we assume that malicious vehicles with a reputation value less than 10, it will be cleared out of the network. Then the malicious vehicles represented by the blue line will be removed. Even if the malicious vehicle does not meet the standard of being removed, the rising speed of the vehicle trust value is obviously slowdown, which can effectively counter the on/off attack.

Then we discuss the trend of trust value variation of honest vehicles when they are subjected to bad-mouthing attacks, as shown in Fig. 7. Honest vehicles suffered badmouthing attacks during rounds 5 and 8. red line in Fig. 7 is the Bin Luo et al. scheme, and the vehicle trust value that is attacked is briefly reduced, but subsequent honest behavior will restore the vehicle trust value. The black line in the picture is the Zhaojun Lu's scheme, and the bad-mouthing attack in the fifth round is exposed, so the attack is invalid. and the 8th round of bad-mouthing attack was not found, and the trust value of the vehicle dropped significantly. This scheme of exposing malicious acts by vehicles itself is not stable. the blue line in Fig. 7 is our scheme. if the RSU is hijacked to make a bad-mouthing attack, then the abnormal state of the RSU will be detected before the block out, thus the transaction containing the bad-mouthing attack is invalid,

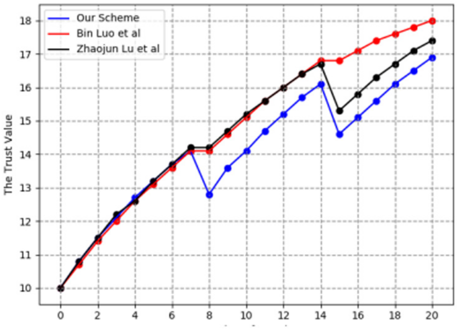


Fig. 7. Trust value tendency under bad-mouthing attack (Color figure online)

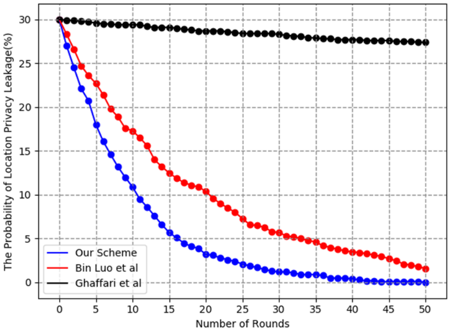


Fig. 8. The probability of location privacy leakage (Color figure online)

so our scheme is generally not affected by the bad-mouthing attack. Figure 7. Trust value tendency under bad-mouthing attack.

5.4 Location Privacy Protection

We will present the probability of location privacy leakage. We assume that when constructing the anonymous cloaking regions, if there is a malicious vehicle, the security of the regions will decrease. We assume that half of the 30% malicious vehicles are ordinary malicious vehicles and half are malicious vehicles that take on/off attacks. We compare the results with the other two articles, as shown in the Fig. 8.

The Blue Line represents our scheme, and the trend is basically consistent with the previous experiment, which can guarantee the security of location privacy after 37 rounds. The red line represents the Bin Luo et al. scheme. The malicious vehicles in the system cannot be eliminated in time, so the probability of privacy leakage of the scheme drops slowly. But it can basically achieve its purpose after 50 rounds. The black line represents the scheme of Ghaffari. It is the traditional trust scheme and it does not use blockchain to maintain the trust values of each vehicle separately. Therefore, it is difficult to prevent malicious vehicles from participating in the construction of anonymous cloaking regions, and the probability of requesting the location privacy disclosure of vehicles remains high.

6 Conclusion

This paper proposes a blockchain-based location privacy protection trust model. In this model, the request vehicles send location query requests to a nearby RSU, RSU is responsible for collecting cooperative vehicles to build an anonymous cloaking region. Then it returns a query result to the request vehicles. In this paper, the certificate is used as the false name, and the direct communication between the vehicles is avoided, which reduces the possibility of privacy leakage. And the anonymous cloaking region also protects the vehicles' privacy from the LSP. Our scheme employs the PoW + HotStuff consensus mechanism to maintain the blockchain. It reduces the resource consumption and improves the computational efficiency compared with PoX. Finally, we propose a trust management algorithm and verify the performance of it through experiments.

Acknowledgement. This work was supported partly by National Natural Science Foundation of China (61728204), Key Laboratory of Safety-Critical Software Ministry of Industry and Information Technolog (NJ2018014), Fundamental Research Funds for the Central Universities (NS2019001, NS2018057), Ford Motor URP Funding, Guangzhou Science and Technology Program (202002030246).

References

1. Zhang, K., Ni, J., Yang, K., Liang, X., Ren, J., Shen, X.S.: Security and privacy in smart city applications: Challenges and solutions. IEEE Commun. Mag. **55**(1), 122–129 (2017)

2. Mahmoud, M.E., Shen, X.: An integrated stimulation and punishment mechanism for thwarting packet dropping attack in multihop wireless networks. IEEE Trans. Veh. Technol. **60**(8), 3947–3962 (2011)
3. Lai, C., Zhang, K., Cheng, N., Li, H., Shen, X.: SIRC: a secure incentive scheme for reliable cooperative downloading in highway VANETs. IEEE Trans. Intell. Transp. Syst. **18**(6), 1559–1574 (2016)
4. Singh, M., Kim, S.: Blockchain based intelligent vehicle data sharing framework. arXiv preprint arXiv:1708.09721 (2017)
5. Chow, C.-Y., Mokbel, M.F., Liu, X.: A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In: Proceedings of 14th Annual ACM International Symposium Advance Geographic Information System, pp. 171–178 (2006)
6. Ghaffari, M., Ghadiri, N., Manshaei, M.H., Lahijani, M.S.: P4QS: a peer-to-peer privacy preserving query service for location-based mobile applications. IEEE Trans. Veh. Technol. **66**(10), 9458–9469 (2017)
7. Luo, B., Li, X., Weng, J., et al.: Blockchain enabled trust-based location privacy protection scheme in VANET. IEEE Trans. Veh. Technol. **69**(2), 2034–2048 (2020)
8. Raya, M., Papadimitratos, P., Gligor, V.D., Hubaux, J.-P.: On data-centric trust establishment in ephemeral ad hoc networks. Paper presented at the IEEE INFOCOM 2008-The 27th Conference on Computer Communications (2008)
9. Li, Z., Chigan, C.T.: On joint privacy and reputation assurance for vehicular ad hoc networks. IEEE Trans. Mob. Comput. **13**(10), 2334–2344 (2014)
10. Lei, A., Cruickshank, H., Cao, Y., Asuquo, P., Ogah, C.P.A., Sun, Z.: Blockchain-based dynamic key management for heterogeneous intelligent transportation systems. IEEE Internet Things J. **4**(6), 1832–1843 (2017)
11. Yang, Z., Yang, K., Lei, L., Zheng, K., Leung, V.C.: Blockchain-based decentralized trust management in vehicular networks. IEEE Internet Things J. **6**(2), 1495–1505 (2018)
12. Wang, E.K., Liang, Z., Chen, C., et al.: PoRX: a reputation incentive scheme for blockchain consensus of IIoT. Future Gener. Comput. Syst. **102**, 140–151 (2020)
13. Nakamoto, S., Bitcoin, A.: A peer-to-peer electronic cash system. Bitcoin (2008). https://bitcoin.org/bitcoin.pdf
14. Yang, Z., Zheng, K., Yang, K., Leung, V.C.: A blockchain-based reputation system for data credibility assessment in vehicular networks. Paper presented at the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC) (2017)
15. Daian, P., Pass, R., Shi, E.: *Snow White*: robustly reconfigurable consensus and applications to provably secure proof of stake. In: Goldberg, I., Moore, T. (eds.) FC 2019. LNCS, vol. 11598, pp. 23–41. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32101-7_2
16. Yin, M., Malkhi, D., Reiter, M.K., Gueta, G.G., Abraham, I.: HotStuff: BFT consensus in the lens of blockchain. arXiv preprint arXiv:1803.05069 (2018)
17. Belotti, M., Božić, N., Pujolle, G.: A vademecum on blockchain technologies: when, which, and how. IEEE Commun. Surv. Tutor. **21**(4), 3796–3838 (2019)
18. Zivic, N., Ruland, C., Ur-Rehman, O.: Addressing Byzantine fault tolerance in blockchain technology. Paper presented at the 2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO) (2019)
19. Xhafa, F., Barolli, L., Amato, F.: Advances on P2P, Parallel, Grid, Cloud and Internet Computing: Proceedings of the 11th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC–2016) November 5–7, 2016, Soonchunhyang University, Asian, Korea, vol. 1. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-49109-7
20. Cai, C., Yuan, X., Wang, C.: Towards trustworthy and private keyword search in encrypted decentralized storage. Paper presented at the 2017 IEEE International Conference on Communications (ICC) (2017)

21. Cai, C., Yuan, X., Wang, C.: Hardening distributed and encrypted keyword search via blockchain. Paper presented at the 2017 IEEE Symposium on Privacy-Aware Computing (PAC) (2017)
22. Ni, L.M., et al.: Smart City Research Group, Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. https://www.cse.ust.hk/scrg/
23. Zhaojun, L., Wenchao, L., Qian, W., et al.: A privacy-preserving trust model based on blockchain for VANETs. IEEE Access **6**, 45655–45664 (2018)

SecureRec: Privacy-Preserving Recommendation with Distributed Matrix Factorization

Wenyan Liu, Junhong Cheng, Xiangfeng Wang, and Xiaoling Wang(✉)

East China Normal University, Shanghai, China
{wyliu,jhcheng}@stu.ecnu.edu.cn, xfwang@sei.ecnu.edu.cn,
xlwang@cs.ecnu.edu.cn

Abstract. Recommender systems have received much attention recently because of their abilities to capture the interests of users. A standard solution is to collect and analyze users' historical behavior data, which might raise privacy concerns, e.g., Facebook-Cambridge Analytica data scandal. Collaborative filtering has been widely used in recommender systems for its simplicity. However, it suffers from an efficiency issue owing to a large amount of data and time-consuming operations. Therefore, an interesting question arises: how to provide recommendation services and protect users' privacy at the same time based on distributed matrix factorization? The paradox is that sharing inaccurate information about user data makes it difficult for the recommender to infer personal preference. In this paper, we propose an item recommender system named SecureRec. We formulate the notion of (α, β)-accuracy. We prove that SecureRec is (α, β)-accurate and ϵ-differentially private. Experimental results on three real-world datasets show that SecureRec achieves comparable precision to non-private item recommendation methods while offering privacy guarantees to users.

Keywords: Differential privacy · Item recommendation · Matrix factorization · Probabilistic analysis · Optimization

1 Introduction

Recent years have witnessed the pervasiveness of customer-oriented applications, such as online shopping websites, social media platforms, and location-based services (e.g., eBay, Youtube, and Gowalla). With these applications, producers sell their items or services to customers (for example, a buyer purchases a computer, a customer goes to a restaurant). In this context, personalized recommender systems attempt to assist customers in discovering their potential interests by collecting and analyzing their historical behavior data. Typically, producers (i.e.,

This work was supported by National Key R&D Program of China (No. 2017YFC0803700), NSFC grants (No. 61532021 and 61972155), Open Project Fund from Shenzhen Institute of Artificial Intelligence and Robotics for Society.

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 480–495, 2020.
https://doi.org/10.1007/978-3-030-65390-3_37

item providers) and customers (i.e., users) register with an application broker (i.e., model server) who predicts user preference for items.

The model server keeps collecting massive behavior data from users, aiming at digging the users' preference out and making a profit from their applications. However, the model server may not be trustworthy, and disclosing historical data may severely increase the risk of the privacy leakage [15]. According to the user profile, an adversary can launch a broad spectrum of attacks, such as spying or snooping, harassment or stalking, hacking or identity theft, and political advertising. Hence, users are inclined to protect themselves from potentially malicious actors. The objects to be protected are 1) privacy of input (for example, a user clicks an item several times), 2) privacy of existence (whether a user clicks an item), 3) user profile privacy, and 4) privacy of recommendation results.

The research community tries to investigate privacy-preserving techniques and recommendation tasks. Recent work [1,6] assumes that the recommender is trustworthy and only preserves the privacy of recommendation results. However, the recommender with user data gathered is likely to become the vulnerability or adversary, causing the leakage of three other privacy objectives. Several studies [10,13,14,17,18] focus on the untrustworthy recommender systems for *explicit feedback* (i.e., rating prediction), which requires the users to spend time on rating the items as preferred or not. Besides, they preserve most of the privacy objectives but the privacy of existence, which is important in *implicit feedback* (i.e., item recommendation). Unfortunately, applying the existing proposals directly for item recommendation leads to low accuracy and severe privacy breach.

We propose SecureRec for the untrusted recommender to protect implicit feedback from users. Our goal is two-fold. First, SecureRec preserves four kinds of privacy. Second, it infers user-preferred items with high accuracy and efficiency. The challenges and contributions are:

- We study privacy in item recommendation for implicit feedback. To ensure that the untrusted recommender cannot infer any private information, we adopt the distributed Matrix Factorization (MF) based on bootstrap sampling to perturb the interaction's existence. A user's input, profile, and recommendation results remain in his device while the Laplace mechanism masks the intermediate results. We prove that SecureRec provides a ϵ-differential privacy guarantee.
- Due to the lack of ratings in the implicit feedback, predicting preference as accurate as the results derived from explicit feedback is difficult. We solve it as a probabilistic ranking problem, formulate the notion of (α, β)-accuracy that measures the probability of error bound, and prove that SecureRec is (α, β)-accurate.
- To alleviate the efficiency issue caused by transferring the computational burden from server to client devices, SecureRec enables parallel computing by dividing the matrix into sub-matrics which retain the low-rank property.

We conduct experiments on three real-world datasets. The results demonstrate that SecureRec protects the privacy of user information without significantly affecting the accuracy of the recommendation results in an efficient way.

Table 1. Notations

Symbol	Specification	Symbol	Specification
i	i-th user	j	j-th item
h	h-th sub-matrix	d	Dimension of latent factors ($\ll \lvert m\rvert, \lvert n\rvert$)
m	Number of users	n	Number of items
λ	Regularization parameter w.r.t. U or V	η	Noise matrix w.r.t. R ($\in \mathbb{R}^{n\times d}$)
H	Number of sub-matrices	R	User-item interaction matrix ($\in \mathbb{R}^{m\times n}$)
P	Binary matrix of R ($\in \mathbb{R}^{m\times n}$)	C	Confidence matrix of P ($\in \mathbb{R}^{m\times n}$)
U	User latent matrix w.r.t. R ($\in \mathbb{R}^{m\times d}$)	V	Item latent matrix w.r.t. R ($\in \mathbb{R}^{m\times d}$)
W	Weight matrix of P ($\in \mathbb{R}^{m\times n}$)		

2 Related Work

The researchers propose a variety of techniques to perform tasks in a privacy-preserving way. We divide these techniques into obfuscation, anonymization, cryptography, and differential privacy (DP) categories. The first three techniques either pay intolerable time overhead or fail to preserve strict privacy. Thus, the focus of research has shifted from the preceding three categories to the lightweight DP technique with a formal guarantee recently.

In the untrustworthy scenario, DPRS [13] and DPMF [10] are the first methods to introduce DP into rating prediction. DP-UnP^{3}R [17] and EpicRec [18] inject instance-based noise to protect items within the pre-assigned categories. They require additional information on items, which may not always exist. PrivSR [14] further divides ratings into sensitive and public ones, which trades partial information for the quality of service. Although protection for the exact values of historical behavior data is similar in rating prediction and item recommendation, our objective is to provide extra protection for the existence of interaction.

Item recommendation grabs increasing attention. DMF [3] probes into Point-of-Interest (POI) recommendation and protecting users' privacy. One of DMF's main contributions is to apply random walk theory to the trusted relationship between nearby users in the location-based network. However, side-information provides more opportunities for attackers. GD-DR [19] adopts random projection-based dimensionality reduction technique to tackle utility issues. However, the size of the conversion matrices is proportional to the number of items, which is too big for mobile users whose computational resources are limited.

3 Preliminaries

3.1 Matrix Factorization

Collaborative Filtering (CF) is regarded as one of the most popular and successful recommendation techniques because of its simplicity. MF [16] is among the best CF algorithms driven by the reasonable hypothesis that users with similar past behavior tend to be interested in similar items. The basic idea of MF is to embed users and items into low-dimensional dense vector space and factorize the sparse user-item interaction matrix into the product of user and item latent matrix. MF simulates the observed patterns and completes the missing entries.

Probabilistic Matrix Factorization (PMF) [16] is applied to rating prediction on explicit feedback datasets while Weighted Matrix Factorization (WMF) [9] is used to item recommendation on implicit feedback datasets. Table 1 summarizes the symbols. The objective function of WMF [9] is:

$$\min_{U,V} \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij}(P_{ij} - U_i^\top V_j)^2 + \lambda(||U||_F^2 + ||V||_F^2), \tag{1}$$

where decomposes R into P and C. P is the target matrix that we aim to recover. Let the elements in R higher than 0 replaced by 1 in P, and the others are zero. C describes the interest level based on R. The second addend is the regularization term for the prevention of over-fitting. Our approach can be easily extended to the variants of WMF. We choose to discuss one of them. LWMF [20] advocates modeling the local property of implicit feedback. We adopt the item-based variant of LWMF.

3.2 Differential Privacy

Intuitively, given two datasets that only differ in at most one record, the output distribution of a randomized function on these two datasets shall be close. Formally, ϵ-indistinguishable DP [5] is defined as follows: Given two neighboring datasets D and D'. A randomized function F is ϵ-distinguishable if and only if for any output O, $\frac{\Pr[F(D) \in O]}{\Pr[F(D') \in O]} \leq e^\epsilon$, where ϵ is the privacy budget.

DP is available to assist in aggregate-level information, which leads to analytical useful, and privacy-dependable work. SecureRec satisfies DP if and only if the recommendation result is distinguishable, withholding the information about whether a particular user interacts with a single item.

4 Private-Preserving Recommendation

4.1 Problem Statement

Let R_{ij} be the observed feedback from the i-th user on the j-th item. The task of privacy-preserving item recommendation is 1) to provide a list of top-k items that i might like depending on R and 2) under the constraint that the recommender is untrustworthy, each user intends to keep his preference secret, that is, the user shares the input and the existence of his historical behavior data R_i, latent factors U_i, and recommendation results with nobody else.

4.2 System Architecture

SecureRec decouples the computation into the update processes of U and V. Each user i builds his profile locally, which keeps U_i secret. Since V is closely associated with U, SecureRec publishes V under the requirement of DP. However, the subtraction of noisy gradients at consecutive iterations may eliminate the influence of global noise. Thus, SecureRec also adds temporary noise to the item gradients at each iteration. Moreover, if the users directly request to update V_j, they breach the existence of the entries in R. Thus, they extract likes and dislikes in SecureRec. Lastly, i infers his recommendation results locally with U_i and V. So far, we achieve all of the four privacy objects.

Through the analysis above, we propose the framework of SecureRec. The model server, item provider, and users are in the situation of tripartite confrontation. Users store the global noise at their side. Due to the nature of the additive noise, it is a challenging problem that will be solved later. When a user requests a batch of item profiles, the item provider responds with latent vectors and temporary noise. Next, every user i calculates the gradients of U_i and V in conjunction with R_i. He then updates U_i locally and sends item gradients perturbed with global and temporary noise to the model server. Upon receiving the noisy item gradients, the model server adds them up according to identical items. It sends the aggregation to the item provider. After that, the item provider removes the summation of the temporary noise and updates V. In the model training phase, users acquire personal recommendation results without sharing R. It is a meaningful scheme and will be discussed later.

4.3 Privacy Model and Assumptions

The model server is assumed to follow the protocol specification but watches for his chance of exploiting the user information, such as intermediate gradients or deviation of the gradients from adjacent iterations. The item provider is assumed to be honest-but-curious about the user preference from which he might commercially benefit. Moreover, users do not have to trust each other.

Our goal is to protect the input and existence of R, U, and results during recommendation. Once a user decides to interact with an item, the user himself may directly disclose information to the item provider. However, such additional disclosure is out of our scope, as each user has the right to disclose his individual information. We focus on what happens before the actual interaction behavior.

In practice, the item provider is a retailer who wants to sell goods to the users. The model server is like a shopping guide who holds a recommendation model. It allows the users to find the products they need quickly and charges the item provider a fee for promoting turnover.

Algorithm 1: Initialization

```
Model server calculates item similarity $K(V_j, V_{j'})$;
Model server chooses $A$ based on DCGASC;
Model server calculates $W$ based on $A$;
Model server publishes $A$;
Users calculate $P$ based on Equation (3);
Users calculate $C$ based on Equation (4);
Model server calculates $\theta$ based on Equation (7);
Model server sends $2\sqrt{\theta}$ to user;
Users receive $2\sqrt{\theta}$ from model server;
Users calculate $\phi \sim \mathcal{N}(0,1)$;
Users calculate $\zeta$ based on Equation (8);
return $\eta$ to users;
```

4.4 Implementation

SecureRec consists of the Initialization and Model Training phase. To be specific, the objective function of SecureRec is perturbed as follows:

$$\min_{U^h, V^h} \sum_{h=1}^{H} \sum_{i=1}^{m} \sum_{j=1}^{n} W_{ij}^h C_{ij}^h \left(P_{ij}^h - {U_i^h}^\top V_j^h \right)^2 + \left(\eta_j^h\right)^\top V_j^h + \lambda^h \left(||U||_F^2 + ||V||_F^2 \right), \tag{2}$$

where η^h is composed of global noise ζ^h and temporary noise ξ^h. P is approximated by the set of sub-matrices $\{P^1, P^2, \cdots, P^H\}$:

$$P_{ij}^h \approx \frac{1}{Z_{ij}} \sum_{h=1}^{H} W_{ij}^h {U_i^h}^\top V_j^h, \tag{3}$$

where $Z_{ij} = \sum_{h=1}^{H} W_{ij}^h$ is the normalizer. Each entry C_{ij}^h in C^h is derived:

$$C_{ij}^h = 1 + \log(1 + R_{ij}^h \times 10^{\omega}), \tag{4}$$

where the constant ω is used to control the rate of increment.

Let $E_{(t-1)}$ substitute for $2W_{ij}^h C_{ij}^h (P_{ij}^h - {U_i^h}_{(t-1)}^\top {V_j^h}_{(t-1)})$. The corresponding update rules in the t-th iteration are:

$${U_i^h}_{(t)} = {U_i^h}_{(t-1)} + \gamma \left[\textstyle\sum_{j=1}^{n} {V_j^h}_{(t-1)} E_{(t-1)} + 2\lambda^h {U_i^h}_{(t-1)} \right], \tag{5}$$

$${V_j^h}_{(t)} = {V_j^h}_{(t-1)} + \gamma \left[\textstyle\sum_{i=1}^{m} {U_i^h}_{(t-1)} E_{(t-1)} + 2\lambda^h {V_j^h}_{(t-1)} + \eta_j^h \right]. \tag{6}$$

Initialization. Initialization is an essential phase which aims to determine the training manner and the constant global noise. The pseudo-code is depicted in Algorithm 1. In this phase, SecureRec completes two primary tasks.

Algorithm 2: Model Training

```
   Input  : $R$, $H$, DCGASC, sets $A^h$ covered by each anchor point $a^h$, $C$, $\lambda$, $\epsilon$
   Output: $U$, $S$
1  for $h \leftarrow 1$ to $H$ do
2      for $j \leftarrow 1$ to $n$ do
3          if $K(a^h, j) > 0$ then
4              Server calculates $A^h \leftarrow A^h \bigcup \{j\}$;
5          end
6      end
7      for $t \leftarrow 1$ to $T$ do
8          User requests $V_j^h(t-1)$ and $V_{j'}^h(t-1)$ from item database;
9          Item provider calculates random noise vector $\xi_j^h(t)$;
10         Item provider sends $\xi_j^h(t)$ to user;
11         User receives $\xi_j^h(t)$ from item provider;
12         User updates $U_u(t)$ locally based on Equation (5);
13         User predicts $P(t)$;
14         User sends $V_j^h(t) + \zeta_j + \xi_j^h(t)$ and $V_{j'}^h(t) + \zeta_{j'} + \zeta_{j'}^h(t)$ to model server;
15         Model server calculates summation of $\sum_j V_j^h(t) + \zeta_j + \xi_j^h(t)$;
16         Model server sends $\sum_j V_j^h(t) + \zeta_j + \xi_j^h(t)$ to item provider;
17         Item provider receives $\sum_j V_j^h(t) + \zeta_j + \xi_j^h(t)$ from model server;
18         Item provider calculates $V_j^h(t) = \sum_j [V_j^h(t) + \zeta_j + \xi_j^h(t)] - \xi$;
19         Item provider updates $V$ based on Equation (6);
20     end
21 end
22 User predicts the missing entries in $R$;
23 return $U$ to users, $V$ to item database;
```

In the first task, the model server identifies the similarity between items using prior knowledge (Line 1), where $K(V_j, V_{j'})$ denotes the relevance of V_j to $V_{j'}$. Next, the model server divides the original matrix into sub-matrices based on DCGASC [20] (Lines 2–4), where A is the anchor point set. In the case of a cold start where the model server has not gathered sufficient item profiles, SecureRec supports the global WMF without modeling the local property and set $H = 1$. After the model server obtains enough V or several rounds of model training, it restarts the initialization protocol and sets an appropriate H. Then, the users calculate P and C in a distributed way (Lines 5–6).

In the second task, the model server and users negotiate ζ_{ij}^h (Lines 7–11). By the central limit theorem, the summation of η tends to converge to a normal distribution. Fortunately, a Laplace random variable ζ has the representation:

$$\zeta = \sqrt{2\theta}\phi, \tag{7}$$

where the random variables θ and ϕ follow the standard exponential and normal distribution. V is ϵ-differentially private if each η_{ij}^h of Eq. (2) is independently and identically drawn from $Lap(0, \frac{2\Delta\sqrt{d}}{\epsilon})$, where $\Delta = \max(P) - \min(P)$. Since $\zeta_j^h \propto e^{-\frac{\epsilon||\zeta_j^h||}{2}}$, users compute the noise as:

$$\zeta_{ij}^h = \frac{2\sqrt{d}}{\epsilon} \cdot \sqrt{2\theta}\phi. \tag{8}$$

Model Training. The dominant optimization process in WMF [9] is Alternating Least Square (ALS) [9]. Nevertheless, ALS caches all the entries in U, which is contrary to the privacy-preserving scenario. SecureRec minimizes the objective function with the sample-by-sample computational method. As shown in Eq. (5), we observe that U_i is related to the records of i and is independent of the records of the other users. It suggests that users can update their latent factors locally. SecureRec is different from the previous work [10]. In their proposal, at the beginning of each iteration t, i requests V_j from the recommender directly. In our scenario, it makes C_{ij} disclose to the model server entirely, which makes privacy policies be of no use.

The pseudo-code is depicted in Algorithm 2. For each sub-matrix (Line 1–5), SecureRec obtains the anchor points. At each iteration (Line 7), the user requests two latest item profiles, including a like and a dislike (Line 8). The item provider brings ξ_{ij} to the user (Lines 9–10). The user updates U_i locally (Lines 11–13). He sends the summation of the gradients of the items, the pre-defined ζ_{ij} and ξ_{ij} to the model server (Line 14). The model server aggregates the update values' fragments and forwards them to the item provider (Lines 15–16). The item provider eliminates the impact of ξ and updates the item profiles (Lines 17-19). Finally, the user ranks his preference locally (Line 22).

5 Accuracy, Privacy and Overhead Analysis

5.1 Accuracy Analysis

To discuss accuracy, we define what we mean by the accuracy of an algorithm that outputs x' in a metric space Y in response to input x in a metric space X. We formulate the notion of (α, β)-accuracy as follows:

Definition 1 ((α, β)-Accuracy). *An algorithm $f : X \to Y$ is (α, β)-accurate if with probability at least β, it outputs Y such that the error is under α, i.e.,*

$$|f(x_t) - f(x^*)| \leq \mathcal{O}(\frac{1}{t}) + \alpha.$$

Difficulties of optimizing a non-convex function come from two aspects [7]: 1) the objective function may have many local minima, while it is hard to find the global minimum among them; 2) the saddle points might hinder optimization to the local minima. To clarify the matters, we perform the following conversion.

Consider the function $f_{\textsc{SecureRec}} : \mathbb{R}^{n\times d} \rightarrow \mathbb{R}$, i.e.,

$$f_{\textsc{SecureRec}}(U,V) := \|P - U^\top V\|^2 + \lambda\|U\|^2 + \lambda\|V\|^2,$$

which is a non-convex function denoted as $g \circ \mathcal{S}$, while

$$\mathcal{S}(U,V) = \big((P - U^\top V), U, V\big)\,, g(x,y,z) = \|x\|^2 + \|y\|^2 + \|z\|^2. \tag{9}$$

We obtain that $\mathcal{S}$ is a homeomorphism and g is convex. Based on [2] Theorem 2.1, $f_{\textsc{SecureRec}}$ is a lower semicontinuous lower-level-set function. We have that every local minimizer of $f_{\textsc{SecureRec}}$ is a global minimizer.

U and V have identical dimension d. Denote $X = (U_1, U_2, \cdots, U_m, V_1, V_2, \cdots, V_n)$. Convert $P - U^\top V$ to

$$\textstyle\sum_{i,j}[B_{ij} - (XE_i)^\top(XE_j)]^2 = \sum_{i,j}[B_{ij} - (X^\top X)_{ij}]^2,$$

where E_i is a $(m+n)$-dimensional vector with the i-th entry value 1 and the others 0, $i \in [1,n]$, $j \in [n+1, n+m]$. Notice that B is a block matrix with a valid $m \times n$ group and the others will never be observed. Therefore, it is obvious that $B = Q^\top Q$. With high probability [8], Stochastic Gradient Descent (SGD) on the regularized objective converges to a solution X such that $X^\top X = Z^\top Z = B$ in polynomial time from any starting point. Gradient descent converges to such a point with probability 1 from a random starting point.

In summary, our goal is to analyze the convergence accuracy of f, rather than the convergence rate, where $f(x')$ restricted to 2ρ neighborhood of a local minimum x^* ($||x' - x^*|| \leq 2\rho$) is convex. Loss fluctuates as the Stochastic Gradient (SG) steps are taken. We combine the randomness in SG and artificial noise, and rewrite the update rule $x_t = x_{t-1} - \gamma(\nabla f(x_{t-1}) + \mu_{t-1})$ [7], where $\mu = SG(x_t) - \nabla f(x_{t-1}) + \eta$. In the objective perturbation [11], the loss of accuracy is associated with the gradient.

SecureRec works with the noisy version of the actual gradient. To this end, we take δ to rephrase the error. Consider f is a convex function whose gradient is L-Lipschitz continuous. The noisy gradient $g_{\delta,L}(y)$ satisfies

$$-\delta \leq f(x) - [f(y) + g_{\delta,L}^\top(y)(x-y)] \leq \frac{L}{2}||x-y||^2 + \delta. \tag{10}$$

Note that if δ is zero, then $g_{0,L}(y)$ is the true gradient.

Lemma 1. *For all $t \geq 1$, we have*

$$\textstyle\sum_{i=0}^{t-1} \frac{1}{L}[f(x_{i+1}) - f(x^*)] \leq \frac{1}{2}||x_0 - x^*||^2 + \sum_{i=0}^{t-1} \frac{2\delta_i}{L}. \tag{11}$$

Proof. Denote $r_t = ||x_t - x^*||^2$, $g_t = g_{\delta_t,L}(x_t)$.

$$\begin{aligned} r_{t+1}^2 &\leq r_t^2 + \frac{2}{L}g_t^\top(x^* - x_{t+1}) - ||x_{t+1} - x_t||^2 [4] \\ &= r_t^2 + \frac{2}{L}g_t^\top(x^* - x_{t+1}) - \frac{2}{L}[g_t^\top(x_{t+1} - x_t) + \frac{L}{2}||x_{t+1} - x_t||^2]. \end{aligned}$$

By Eq. (10),

$$r_{t+1}^2 \leq r_t^2 + \frac{2}{L}[f(x^*) - f(x_t) + \delta_t] - \frac{2}{L}[f(x_{t+1}) - f(x_t) - \delta_t],$$
$$\vdots$$
$$r_1^2 \leq r_0^2 + \frac{2}{L}[f(x^*) - f(x_0) + \delta_0] - \frac{2}{L}[f(x_1) - f(x_0) - \delta_0].$$

Summing up these inequalities for $i = 0, \cdots, t-1$, we obtain Eq. (11).

Theorem 1. SecureRec *is* $\left(2\delta, \beta = \Pr\{\chi^2(d) \leq \frac{(\delta-2\kappa)^2}{8b^2}\}\right)$*-accurate.*

Proof. Denote $\bar{x}_t = \frac{\sum_{i=0}^{t-1} L^{-1} x_{i+1}}{\sum_{i=0}^{t-1} L^{-1}}$. By Lemma 1, we have

$$f(\bar{x}_t) - f(x^*) \leq \frac{\sum_{i=0}^{t-1} L^{-1}[f(x_{i+1}) - f(x^*)]}{\sum_{i=0}^{t-1} L^{-1}} \leq \frac{\frac{1}{2}||x_0 - x^*||^2 + \sum_{i=0}^{t-1} L^{-1} 2\delta_i}{\sum_{i=0}^{t-1} L^{-1}}.$$

When $\delta_i = \delta$, $f(\bar{x}_t) - f(x^*) \leq \frac{L{r_0}^2}{2t} + 2\delta$. Thus, SecureRec asymptotically tends to 2δ when t is large enough.

We separate noise μ from the noisy gradient $g_{\delta,L}(x)$, that is, $g_{\delta,L}(x) = g_{0,L}(x) + \mu = g_{0,L} + \mu_{SG} + \eta$, where $\mu_{SG} = SG(x) - \nabla f(x)$. By Eq. (10), we have $-\delta \leq f(x) - \{f(y) + [g_{0,L}(y) + \mu]^\top (x-y)\} \leq \frac{L}{2}||x-y||^2 + \delta$, and thus $|\mu^\top (x-y)| \leq \delta$, then $||\mu|| \cdot ||x-y|| \leq \delta$, and $||\mu||^2 \leq (\frac{\delta}{2} - ||\mu_{SG}||)^2$. We assume that $||\mu_{SG}||^2 \leq \kappa^2$, where $\kappa > 0$. In our scenario, SGD batch size is $2m$, which leads to a small variance. Then, $||\mu||^2 \leq (\frac{\delta}{2} - \kappa)^2$. η in SecureRec is drawn from Laplace distribution with scale b. According to 3-σ rule, the percentage of values that lie within one standard deviation from the mean in a Laplace distribution is higher than in a normal distribution with the same variance.

Let ϕ denote the random variable drawn from Normal distribution with mean 0 and the same variance as $Lap(b)$, which $\phi \sim \mathcal{N}(0, \sqrt{2}b)$. The Euclidean norm is captured by the square root of the summation of the variable squares, which is distributed according to the Chi-squared distribution with d degrees of freedom. Thus, $\chi^2(d) = \sum_{i=0}^{d-1} (\frac{\phi_i}{\sqrt{2}b})^2$. Since $\Pr\{\chi^2(d) \leq \frac{(\delta-2\kappa)^2}{8b^2}\} < \Pr\{\sum_{i=0}^{d-1} Lap^2(\lambda) \leq \frac{\delta^2}{4}\}$, SecureRec is (α, β)-accurate for $\alpha = 2\delta$ and $\beta = \Pr\{\chi^2(d) \leq \frac{(\delta-2\kappa)^2}{8b^2}\}$.

Note that optimizing a non-convex function might be trapped in the saddle points, where $\nabla f(x_t) = \nabla f(x^*) = 0$, but $f(x_t) > f(x^*)$. Under some circumstances, moderate noise in the stochastic gradient helps the algorithm to escape from the saddle points (as shown in the experiments).

5.2 Privacy Analysis

We analyze the privacy of sub-matrix V^h.

Theorem 2. *If noise entries η_j^h are i.i.d. random variables drawn from $Lap(\frac{4}{\epsilon})$, each sub-matrix V^h preserves ϵ-DP.*

Proof. Consider two neighboring datasets D and D' with only one different entry R_{pq} and $R'_{pq}(1 \leq p \leq n, 1 \leq q \leq m)$. Let V^* be the optimal solution. Then $\frac{\partial \mathcal{L}(V^*,D)}{\partial V} = \frac{\partial \mathcal{L}(V^*,D')}{\partial V} = 0$. That is, $2W_{ij}^h C_{ij}^h U_i^h (P_{ij}^h - {U_i^h}^\top V_s^h) + 2\lambda^h V_s^h + \eta_j^h = 2W_{ij}^h C_{ij}'^h U_i^h (P_{ij}'^h - {U_i^h}^\top V_s^h) + 2\lambda^h V_s^h + \eta_j'^h$.

As only R_{pq} and R'_{pq} are different in D and D', $\eta_j^h - \eta_j'^h = 2W_{ij}^h U_i^h [(C_{ij}'^h P_{pq}'^h - C_{ij}^h P_{pq}^h) + U_j^h V_j^h (C_{ij}'^h - C_{ij}^h)]$.

Since $R'_{pq} - R_{pq} = 1$, $C_{ij}^h = 1 + log(1 + R \cdot 10^w), C_{ij}'^h \leq 1 + log(1 + R \cdot 10^w + 10^w)$. Thus, $C_j'^h - C_j^h \leq log\frac{1+R\cdot 10^w + 10^w}{1+R\cdot 10^w} \leq log2$, so $||\eta_j^h - \eta_j'^h|| \leq 4$.

With the neighbouring datasets D and D', the probability of outputting the same j^* is $\frac{\Pr[j=j^*|D]}{\Pr[j=j^*|D']} = \frac{\prod_{j\in n} P(\eta_j^h)}{\prod_{j\in n} P(\eta_j'^h)} = e^{-\frac{\epsilon(\sum_j ||\eta_j^h|| - \sum_j ||\eta_j'^h||)}{4}} = e^{\frac{\epsilon(||\eta_q^h - \eta_q'^h||)}{4}} \leq e^\epsilon$. Therefore, each sub-matrix V^h satisfies ϵ-DP.

Further, we analyze the privacy of the whole matrix V.

Theorem 3. *If each V^h preserves ϵ-DP, V is ϵ-differentially private.*

Proof. Let V be divided into H mutually exclusive sub-matrices $V = V^1, V^2, ..., V^h$, and each of the sub-matrices be ϵ^h differentially private. According to parallel composition theorem, V is $Max_{1\leq h\leq H}\epsilon_h = \epsilon$-differentially private.

5.3 Computation and Communication Overhead

Computation Overhead. At each iteration, every user updates U_i and the gradients of U_j, whose computation complexity is $O(d)$. The model server aggregates the updates, whose computation complexity is $O(|U| \cdot d)$. The item provider removes temporary noise, whose computation complexity is $O(d|V|)$.

Communication Overhead. At each iteration, every user downloads $2 \cdot d$ item latent factors and sends d item latent vectors together with noise to the model server. Then, the model server forwards $O(d|V|)$ noisy latent factors to the item provider after aggregation. After that, the item provider uploads $d|V|$. In the worst case for the larger Gowalla dataset with $d = 40$, if an item is covered in at most three sub-matrices, each user downloads at most 1.6 KB data and uploads 0.8 KB data to the model server. The model server downloads at most 8 MB and uploads 9.6 MB of all the 24,238 items to the item provider. In the real world, the number of users and items in each sub-matrix is far smaller than the total amount. It indicates that even for the larger dataset under the worst case, SecureRec has low communication cost.

6 Performance Evaluation

We conduct a series of experiments to answer the following research questions.

- **RQ1:** To what extent can SecureRec build user profiles?
- **RQ2:** How much noise can SecureRec tolerate?
- **RQ3:** How does bootstrap sampling help in both accuracy and privacy?

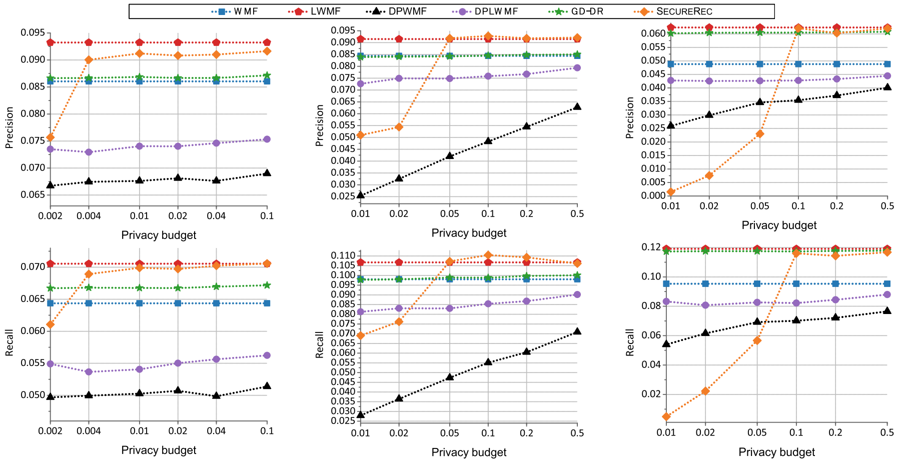
Table 2. Comparison of precision and recall

Method		Dianping				Foursquare				Gowalla			
		Pre@5	Rec@5	Pre@10	Rec@10	Pre@5	Rec@5	Pre@10	Rec@10	Pre@5	Rec@5	Pre@10	Rec@10
WMF	d=10	0.0946	0.0353	0.0793	0.0590	0.1098	0.0645	0.0847	0.0993	0.0523	0.0561	0.0385	0.0779
LWMF		0.1048	0.0403	0.0879	0.0660	0.1166	0.0693	0.0898	0.1047	0.0704	0.0731	0.0528	0.0990
DPWMF		0.0829	0.0297	0.0701	0.0502	0.0600	0.0342	0.0507	0.0562	0.0426	0.0468	0.0316	0.0662
DPLWMF		0.0882	0.0327	0.0747	0.0558	0.1010	0.0575	0.0780	0.0885	0.0549	0.0570	0.0407	0.0795
DG-DR		0.0984	0.0382	0.0815	0.0622	0.1020	0.0619	0.0793	0.0929	0.0678	0.0715	0.0510	0.1009
SecureRec		**0.1001**	**0.0398**	**0.0833**	**0.0646**	**0.1128**	**0.0688**	**0.0875**	**0.1048**	**0.0692**	**0.0747**	**0.0524**	**0.1088**
		-4.48%	-1.24%	-5.23%	-2.12%	-3.26%	-0.72%	-2.56%	+0.10%	-1.70%	+2.19%	-0.76%	+9.90%
WMF	d=20	0.1020	0.0390	0.0860	0.0644	0.1077	0.0632	0.0844	0.0980	0.0595	0.0636	0.0442	0.0871
LWMF		0.1105	0.0423	0.0932	0.0705	0.1192	0.0712	0.0915	0.1067	0.0777	0.0786	0.0581	0.1110
DPWMF		0.0808	0.0298	0.0690	0.0514	0.0593	0.0347	0.0482	0.0551	0.0442	0.0479	0.0340	0.0694
DPLWMF		0.0907	0.0341	0.0753	0.0562	0.0963	0.0530	0.0758	0.0854	0.0595	0.0612	0.0434	0.0841
GD-DR		0.1039	0.0406	0.0867	0.0668	0.1081	0.0656	0.0846	0.0988	0.0758	**0.0774**	0.0567	0.1084
SecureRec		**0.1097**	**0.0436**	**0.0916**	**0.0706**	**0.1213**	**0.0722**	**0.0928**	**0.1105**	**0.0775**	0.0767	**0.0602**	**0.1155**
		-0.72%	+3.07%	-1.72%	+0.14%	+1.76%	+1.40%	+1.42%	+3.56%	-0.26%	-2.42%	+3.61%	+4.05%
WMF	d=40	0.0997	0.0388	0.0844	0.0650	0.1004	0.0610	0.0741	0.0922	0.0650	0.0680	0.0485	0.0953
LWMF		0.1094	0.0431	0.0918	0.0714	0.1178	0.0737	0.0907	0.1054	0.0823	0.0833	0.0623	0.1191
DPWMF		0.0750	0.0286	0.0646	0.0491	0.0578	0.0341	0.0471	0.0541	0.0462	0.0491	0.0355	0.0701
DPLWMF		0.0827	0.0315	0.0698	0.0527	0.0921	0.0507	0.0727	0.0808	0.0580	0.0603	0.0428	0.0822
GD-DR		0.1020	0.0402	0.0861	**0.0668**	0.1077	0.0686	0.0840	0.1027	0.0800	**0.0824**	0.0605	**0.1173**
SecureRec		**0.1079**	**0.0413**	**0.0886**	0.0662	**0.1108**	**0.0702**	**0.0910**	**0.1072**	**0.0802**	0.0777	**0.0621**	0.1160
		-1.37%	-4.18%	-3.49%	-6.44%	-5.94%	-4.75%	+0.33%	+1.71%	-2.55%	-1.08%	-0.32%	-1.51%

6.1 Experimental Setup

The implementation is executed on machines with Intel Core i5-4460S processors at the clock speed of 2.90 GHz, with 16.0 GB of RAM. The experiment is written in the Java Programming language. We use 5-fold cross-validation.

Datasets. We evaluate SecureRec on three real-world datasets. The check-ins in the Dianping dataset are made in Shanghai, China, between January 2003 and August 2016. The check-ins in the Foursquare dataset [21] are made in Singapore between August 2010 and July 2011. The check-ins in the Gowalla dataset [21] are made in California and Nevada between February 2009 and October 2010. Dianping contains 169, 940 check-ins made by 2, 031 users at 5, 649 POIs. Foursquare contains 194, 108 check-ins made by 2, 321 users at 5, 596 POIs. Gowalla contains 456, 988 check-ins made by 10, 162 users at 24, 238 POIs.



(a) Dianping (b) Fourquare (c) Gowalla

Fig. 1. Precision and Recall@10, change ϵ.

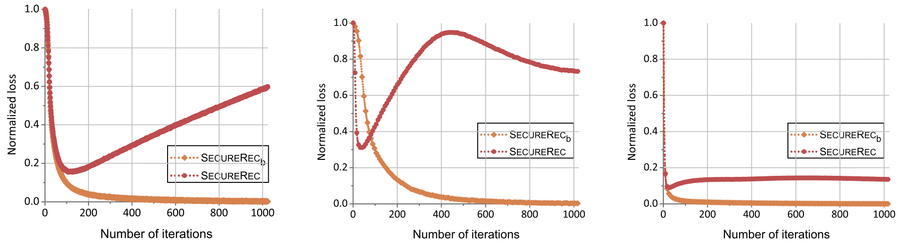
Comparison Methods. We compare SecureRec with the generic method **WMF** [9], which trains the whole matrix with a uniform weight to missing entries; the POI recommendation method **LWMF** [20], which trains the local item-based matrices; **DPWMF** and **DPLWMF**, which are the private implementation of WMF and LWMF with input perturbation respectively; **GD-DR** [19], which is under Local DP with dimensionality reduction.

Metrics. *Precision* and *Recall@k*, where $k = 10$ unless otherwise specified.

Parameter Settings. We follow the parameter settings in counterpart methods [20]. The weight parameter is 4 for Dianping, 2 for Foursquare, and 3 for Gowalla. The learning rate is 0.1 for Foursquare and 0.05 for Dianping and Gowalla. λ is set to 0.01 for Dianping and 0.005 for Foursquare and Gowalla. The bandwidth and discount for DCGASC [20] are set to 0.8 and 0.5 for Dianping and 0.4 for Foursquare and Gowalla. The number of anchor points is 100. The number of iterations is 64 unless otherwise specified.

6.2 Recommendation Accuracy (RQ1)

Table 2 provides the results of Precision and Recall. SecureRec outperforms the counterpart methods regarding different dimensions in most cases. It is suggested that the error range is allowed to fluctuate 3%, and the refinement of a recommendation model is effective if it lifts more than 5% [12]. The results on Foursquare improve those of LWMF by 1.42% and 3.56% at optimal dimension $d = 20$. Thus, SecureRec achieves gratifying ranking results.



(a) Dianping (b) Foursquare (c) Gowalla

Fig. 2. Loss, extending T to 1024 for illustration

6.3 Privacy Protection Level (RQ2)

We traverse ϵ ranging between 0.002 and 0.5. Since check-ins in the Dianping dataset are more scattered than the other two datasets, distinct gradient information tolerates more noise. Figure 1 reports the Precision and Recall@10 influenced by the mechanism. It conforms with the regularity that the less ϵ, the stronger privacy protection, the lower recommendation accuracy, and vice versa. With moderate ϵ, SecureRec brings privacy and accuracy into balance.

Remark 1. Adding extra noise to the updates helps the algorithm escape from saddle points [7], and randomized perturbation sometimes improves the accuracy of recommendations. The non-private methods might trap in a flat region where the gradient is low, and the solution is not a global minimum. The noise guarantees that there is noise in every direction, which allows the algorithm to explore the local neighborhood around saddle points effectively. When ϵ is around 0.1, the calibration of noise contributes to skipping the local minimum. SecureRec with ϵ less than 0.1 decreases in accuracy, and that with ϵ exceeding 0.1 increase in Precision and Recall. When $\epsilon < 0.1$, the noise degrades gradually as the privacy budget increases. Thus, the gradient prevails over the noise. When $\epsilon > 0.1$, the drop in noise mirrors the decline in influence on the gradient. Thus, the accuracy is asymptotically stable at the non-private one.

6.4 Training Efficiency (RQ3)

Recommendation achieves higher accuracy with a smaller loss. We test SecureRec in the ablation experiments: updating with and without bootstrap sampling. For illustration only, we extend the number of iterations to 1024. Figure 2 visualizes the loss along with iterations. The loss based on bootstrap sampling (SecureRec$_b$) monotonously declines while the other fluctuates and slowly approaches local stability because the bootstrap sampling gives missing entries in R practical significance that is negative items the user dislikes.

7 Conclusion

We propose a privacy-preserving recommender system SecureRec for implicit feedback, which enables users' participation without compromising their historical behavior data. Owing to the interaction matrix's local property, SecureRec divides the original matrix into low-rank sub-matrices and processes the training in a distributed environment. We identified bootstrap sampling as a necessary step to accelerate the convergence of the perturbed objective function. SecureRec is not limited to LWMF but extensible to various learning algorithms in recommender systems. We propose the notion of (α, β)-accuracy and prove that SecureRec is (α, β)-accurate and ϵ-differentially private. The experimental results on real-world datasets show that SecureRec is effective, and the cost of recommendation accuracy is practical.

References

1. Berlioz, A., Friedman, A., Kâafar, M.A., Boreli, R., Berkovsky, S.: Applying differential privacy to matrix factorization. In: RecSys, pp. 107–114 (2015)
2. Burai, P.: Local-global minimum property in unconstrained minimization problems. J. Optim. Theory Appl. **162**(1), 34–46 (2014). https://doi.org/10.1007/s10957-013-0432-3
3. Chen, C., Liu, Z., Zhao, P., Zhou, J., Li, X.: Privacy preserving point-of-interest recommendation using decentralized matrix factorization. In: AAAI, pp. 257–264 (2018)
4. Devolder, O., Glineur, F., Nesterov, Y.: First-order methods of smooth convex optimization with inexact oracle. Math. Program. **146**, 37–75 (2013). https://doi.org/10.1007/s10107-013-0677-5
5. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theoret. Comput. Sci. **9**(3–4), 211–407 (2014)
6. Friedman, A., Berkovsky, S., Kaafar, M.A.: A differential privacy framework for matrix factorization recommender systems. User Model. User-Adap. Inter. **26**(5), 425–458 (2016). https://doi.org/10.1007/s11257-016-9177-7
7. Ge, R., Huang, F., Jin, C., Yuan, Y.: Escaping from saddle points. In: COLT, pp. 797–842 (2015)
8. Ge, R., Lee, J.D., Ma, T.: Matrix completion has no spurious local minimum. In: NIPS, pp. 2973–2981 (2016)
9. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: ICDM, pp. 263–272 (2008)
10. Hua, J., Xia, C., Zhong, S.: Differentially private matrix factorization. In: IJCAI, pp. 1763–1770 (2015)
11. Lee, J., Kifer, D.: Concentrated differentially private gradient descent with adaptive per-iteration privacy budget. In: SIGKDD, pp. 1656–1665 (2018)
12. Liu, Y., Pham, T., Cong, G., Yuan, Q.: An experimental evaluation of point-of-interest recommendation in location-based social networks. PVLDB **10**(10), 1010–1021 (2017)
13. McSherry, F., Mironov, I.: Differentially private recommender systems. In: SIGKDD, pp. 627–636 (2009)

14. Meng, X., et al.: Towards privacy preserving social recommendation under personalized privacy settings. World Wide Web **22**(6), 2853–2881 (2018). https://doi.org/10.1007/s11280-018-0620-z
15. Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: S&P, pp. 111–125 (2008)
16. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: NIPS, pp. 1257–1264 (2007)
17. Shen, Y., Jin, H.: Privacy-preserving personalized recommendation. In: ICDM, pp. 540–549 (2014)
18. Shen, Y., Jin, H.: EpicRec: towards practical differentially private framework for personalized recommendation. In: CCS, pp. 180–191 (2016)
19. Shin, H., Kim, S., Shin, J., Xiao, X.: Privacy enhanced matrix factorization for recommendation with local differential privacy. IEEE Trans. Knowl. Data Eng. **30**(9), 1770–1782 (2018)
20. Wang, K., Peng, H., Jin, Y., Sha, C., Wang, X.: Local weighted matrix factorization for top-n recommendation with implicit feedback. Data Sci. Eng. **1**(4), 252–264 (2016). https://doi.org/10.1007/s41019-017-0032-6
21. Yuan, Q., Cong, G., Ma, Z., Sun, A., Magnenat-Thalmann, N.: Time-aware point-of-interest recommendation. In: SIGIR, pp. 363–372 (2013)

Query Processing

Optimizing Scoring and Sorting Operations for Faster WAND Processing

Kun Jiang(✉), Lei Zhu, and Qindong Sun

School of Computer Science and Engineering, Xi'an University of Technology,
Xi'an, China
jk_365@126.com, {leizhu,sqd}@xaut.edu.cn

Abstract. Recent years, a lot of research has focused on how to improve query processing efficiency of large-scale search engines. In this paper, we focus on top-k query processing on document-sorted indexes and the well-known rank-safe dynamic pruning technique called WAND, which can efficiently reduce the hardware computational resources required for the first phase top-k processing in cascade ranking model. Firstly, we carefully analyze the difference of the intrinsic optimization ideas between WAND method and another well-known dynamic pruning method called MaxScore, and provide an updated immediately skipping-over description of WAND (WAND_IS) for faster query processing, which can highly reduce short distance skippings on posting lists. We then propose two key improvements: partial scoring candidates (P.WAND) and less sortings in AND mode (L.WAND) that can leverage the query efficiency of WAND processing. Finally, we perform detailed experiments on TREC GOV2 dataset with self-indexing and Block-Max techniques, which show that our proposals can reduce the query latency by almost 15% on average over the WAND baseline, with a best improvement of about 20%.

Keywords: Inverted index · Dynamic pruning · Weak-AND · Partial scoring · AND mode optimization

1 Introduction

Inverted index traversal techniques which directly operate on inverted index and compute a preliminary top-k score documents as the first phase ranking result, can efficiently quicken the following query processing phases [13]. In this paper, we focus on this first phase top-k query processing and its dynamic pruning optimizations called WAND and MaxScore [2,11,12]. The current WAND method takes advantage of per-term upper-bound value (i.e., maxscore) and the discarding threshold to first select promising document and then evaluate candidate that has a chance to enter the top-k results heap. The upper-bound value can be used to estimate the maximum achievable score of the candidate. After necessary posting skipping and current document alignment, the promising candidates are fully evaluated and their final exact scores are computed to decide whether it has a chance to enter the result heap [7].

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 499–514, 2020.
https://doi.org/10.1007/978-3-030-65390-3_38

However, there exist different descriptions of WAND method on tradition index and on additive index(Block-Max index or Treap index [1,4]), but in essence all perform fully skipping to the candidate document without considering the intrinsic skipping distance of posting lists [2,9]. In this paper, we firstly analyzed the inferiority of current descriptions of WAND method and propose a immediately skipping description that can reduce short distance skipping. As some of the terms contribute smaller real score than its corresponding per-term upper bound, the final score of a candidate has a very high possibility of being smaller than the discarding threshold. This leads to large number of useless scoring functions called. To tackle this problem, we then applied fine-grained partial scoring checking on WAND method, which is a common optimization technique adopted in MaxScore [11,12]. With the increasing of the discarding threshold, all the query terms may appear in the candidate document that make the estimated maximum score large enough as to enter the top-k result heap. Thus, we can further detect the AND mode in WAND processing, leading to less sortings of posting lists that can further reduce the query processing latency.

The remainder of this paper is organized as follows. Section 2 provides background and related works. Section 3 presents our partial scoring improvement on WAND method and the less sorting optimization for AND mode processing. The experimental results and analysis are given in Sect. 4. Finally, Sect. 5 concludes this paper and presents a prospective of future work.

2 Background and Related Work

2.1 Inverted Index and Traversal

An inverted index can be seen as an array of lists or postings, where each entry of the array corresponds to a different term or word in the collection [8]. The set of terms is called the lexicon. For each query term t_i, the index contains a posting list I_{t_i} consisting of a number of postings describing all places where term t_i occurs in the collection. More precisely, each posting in I_{t_i} contains the unique document identifier (docID) d and the number of occurrences of t_i in the document (called term frequency, denoted it as $f_{t_i,d}$). In this paper, we assume postings have docIDs and frequencies that interleaved stored by document-sorted index form.

The posting lists I_{t_i} may consist of many millions of postings $<d, f_{t_i,d}>$, which could be approximately linear with the size of the collection. To allow faster access and limit the amount of memory needed, search engines use various compression techniques that significantly reduce the size of the posting lists within document-sorted indexes (see [8] for some recent work). Researchers have presented self-indexes, which recursively samples block boundary posting and its address offset as the additional synchronization point into blocks of to the compressed posting lists and accelerated the top-k query processing by skipping over compressed blocks [6]. The Block-Max index can also be added to inverted index for fine achievable score of the entries in the block-based inverted lists.

Each block contains information about the maximum achievable impact score among the entries found in it [4].

Given an query $q = <t_1, t_2, \cdots, t_m>$, scalable web search systems typically employ multi-stage retrieval architectures, where an initial stage generates a set of k high scored candidate documents that are then pruned and re-ranked by a complex ranking function with hundreds of features [13]. Much of the research aim to reduce the number of documents considered when creating the initial top-k candidates. For large-scale indexes, the document-at-a-time (DAAT) scheme and its dynamic pruning techniques are commonly used to traverse posting lists within document-sorted index in parallel, leading to faster online top-k query processing performance [12].

2.2 Dynamic Pruning

There are two series of rank safe dynamic pruning techniques, namely, MaxScore and WAND. Both use the maxscore u_{t_i} of each term stored in lexicon to estimate the maximum achievable score $UB(q, d)$ for a given candidate d. By summing the upper bounds of all query terms appearing in a document, we can initially determine an upper bound on the document's query-dependent score with $UB(d, q) = \sum_{t_i \in q \cap d} u_{t_i} \geq S(d, q)$ ($S(d, q)$ is the bag-of-word score of a document d for query q). And the $UB(d, q)$ can be online updated with real per-term score, and used to dynamic pruning candidate d with a lower estimated score than the threshold θ of the result heap.

More precisely, the original MaxScore method [5,12] takes advantage of the partial scoring scheme for optimization. When a document d is scored, we add the partial score of the document to the sum of the accumulated maxscore of the remaining terms. If this sum is less than the threshold θ, then no further evaluation on document d is needed. The MaxScore description by Strohman et al. [11] adopted candidate selection technique, in which the posting lists are sorted by their maxscores before traversing. With the increase of the threshold, the candidate can only be selected in important terms with larger maxscores. More recently, Jonassen et al. [6] present a unified description of the above two versions with efficient skipping operations. In the latest MaxScore, posting lists are sorted from top to bottom by their maxscores and divided into essential lists $q_+ = \{t_1, t_2, ..., t_p\}$ and non-essential lists $q_- = \{t_{p+1}, t_{p+2}, ..., t_m\}$, where p is the smallest value that makes $\sum_{t \in q_-} u_{t_i} < \theta$. The candidate d is selected in essential lists $t_i \in q_+$ and scored in the following posting lists $t_j \in \{t_{i+1}, ..., t_p\} \bigcup q_-$ with partial scoring and skipping techniques.

Broder et al. [2] proposed a Weak-AND (WAND) operator for query processing optimization. The posting lists are sorted in ascending order of their current docIDs ($I_{t_i}.docid < I_{t_j}.docid$ if $i < j$). The pivoting is implemented by finding the first term t_p that its cumulative maxscore is larger than the discarding threshold θ, where p is the lowest value such that $\sum_{i=1}^{p} u_{t_i} \geq \theta$. The aligning is done by checking the current document d equality of the term t_1 and the pivot term t_p. The scoring is fully conducted on all the terms if the alignment is true. The candidate scoring can be avoided if it is not aligned, and the skipping of all posting lists before pivot term t_p to the candidate document d is done thereafter. One drawback of the full skipping and full scoring description of WAND method (WAND_FS) is that there exists large number of short distance skips due to the lazy skipping over detection strategy. And most of the posting lists need to move forward again for an aligned candidate. Recently, Crane et al. [9,10] proposed a WAND version with partial scoring strategy. However, the strategy is applied to the commonly used WAND_FS version unsafely, and the score of the candidate document omits the possible contribution of the terms after pivot term [3].

3 The Proposed Methods

In this section, we first present an updated description of WAND method with immediately skipping, and further propose the partial scoring and AND mode query optimizations.

3.1 Immediately Skipping WAND

As mentioned above, the skipping in WAND_FS is done if the candidate documents are not aligned, and all the posting lists before pivot term advance their current document to the pivot document. This leads to large number of short distance skippings if the pivot is finally discarded, especially for posting lists with a high document frequency. As shown in Fig. 1, documents 99 and 110 are two consecutive pivot documents in WAND_FS method, there will be two times of skippings for posting list of term t_1 to advance its iterator from current position 70 to final pivot position 110. The skipping to position 99 is useless for posting list of term t_1. The full skipping strategy in WAND_FS method makes the posting list advance its iterators to every pivot document if it exists. Thus, there are 4 posting visitings when advance necessary posting lists from pivot 99 to pivot 110. One possible improvement is picking one of the preceding terms and advances its posting list iterator over pivot document location. However, beside the time-consuming selection method, we also need to check if the current document equals to the pivot document after skipping of every posting list, as it may cause a useless resorting. If we check all the equality of the current document and pivot document, the former alignment checking of WAND_FS is completely redundant.

Actually, the WAND method can be reimplemented in an updated description for less sorting and longer skipping distance. We propose a new efficient WAND description by sequentially conducting sorting, pivoting, skipping, aligning and scoring. The skipping is done immediately after the pivot term is determined, and the alignment is done by checking the current document equality of the current term and the pivot term once after a posting list skipping (thus the name WAND_IS). The resorting is triggered once one skip-over of posting list is found to optimize the second scenario mentioned above, which means the pivot document has a high possibility of not surpassing the discarding threshold. In this case, the following posting lists skipping to the pivot document is only for

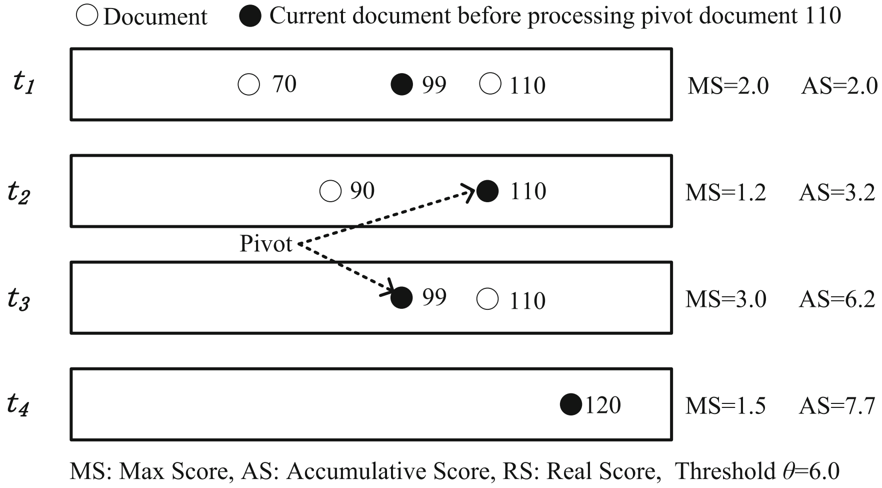

○ Document
● Current document before processing pivot document 110
t_1
○ 70
● 99
○ 110
MS=2.0 AS=2.0
t_2
○ 90
● 110
MS=1.2 AS=3.2
Pivot
t_3
● 99
○ 110
MS=3.0 AS=6.2
t_4
●120
MS=1.5 AS=7.7
MS: Max Score, AS: Accumulative Score, RS: Real Score, Threshold θ=6.0

Fig. 1. Short distance skipping in WAND_FS.

posting list forwarding regardless of the skipping distance. Our improvement can avoid further posting lists skippings if the alignment is false after one posting list skipping, and a resorting on only one skip-over posting list can be done efficiently. A new promising pivot document can provide a much longer skipping distance that can reduce block decompression and postings visiting.

Further, the longer posting list with a higher document frequency is more prone to contain the pivot docid than shorter posting list. Thus, the skip-over case may happen in shorter posting list and the skip-to case may happen in longer posting list, which makes a larger current docid in shorter posting list than that in longer posting list. On one hand, in our updated version of WAND (WAND_IS), the longest posting list has a high possibility of appearing first after the posting lists sorting phase. Thus, to detect the skip-over case timely, the one skipping schema is done on posting lists with descending current docid order, which can detect the skip-over case in time. On the other hand, the longer posting list has a lower contribution to the final document score than the shorter posting list. Thus, to discard the lower scored pivot document, the pivot document scoring is also done on posting lists with descending current docid order. As the skip-over detection is first done on shortest posting list, the there is a small possibility that the new pivot may be equal to the last one. In WAND_IS method, after evaluating pivot 99, term t_2 advance its posting list from position 90 to position 110, and document 110 is selected as a new pivot. Now the skipping of term t_1 can directly advance its posting list from position 70 to the pivot position 110. Thus, there are 3 posting visitings when advance necessary posting lists from pivot 99 to pivot 110 with WAND_IS method.

3.2 P.WAND Algorithm

As defined in [2], the $WAND(x_1, u_{t_1}, x_2, u_{t_2}, ..., x_m, u_{t_m}, \theta)$ operator is true iff $UB(d, q) = u_{t_1}x_1 + u_{t_2}x_2 + ... + u_{t_m}x_m \geq \theta$, where u_{t_i} is maximal contribution of term $t_i(t_i \in q)$ to any document score, x_i (equals 1 or 0) is an indicator variable for the presence of query term t_i in document d and the threshold θ is varied during the algorithm. At some point, when the candidate document d partially finished evaluating the jth contribution score of a term, the $UB(d, q) = s(t_1, d)x_1 + s(t_2, d)x_2 + ... + s(t_j, d)x_j + u_{t_{j+1}}x_{j+1} + ... + u_{t_m}x_m < \theta$, where $s(t_i, d)$ is real score of term t_i to document d. Thus, we can early terminate the scoring phase and select another candidate.

According to the above analysis, we present the pseudocode for the updated version of the WAND heuristic with complete partial scoring strategy (P.WAND) in Algorithm 1. Given the inverted index and the result heap, P.WAND returns the k documents that have the highest scores according to the scoring function $S(q, d)$ $(S(q, d) = \sum_{t_i \in q \cap d} s(t_i, d))$. P.WAND iterates the posting list of every query term by first sorting the posting lists by ascending current docIDs. The maximum achievable score of the candidate is initialized as the cumulative upper bound score of the pivot term and its precedings in pivoting phase (line 8). Then, the skipping is conducted by default on the terms before pivot term (line 11–16), and, if the candidate does not exist in the list, it will point to the first document larger than d. The pivot is discarded if the current document after skipping is not equal to the pivot document (line 13–14). In our updated version of WAND, once one of the posting lists has skipped over the candidate docID, which means the accumulated upper bound score of the pivot term has a high possibility of lower than the discarding threshold, and the current pivot is aggressively discarded and a resorting should be immediately triggered (line 26). Compared to the original WAND that the pivot can only be discarded when the pivot will surely obtain a score that is lower than the threshold. That means in our updated version of WAND, the current pivot term can be discarded to exclude its contribution, but with no further posting lists skippings afterwards. However, in the resorting and pivoting phases it will generate another pivot docid that may be equal to the last one for a low possibility of misestimation. The skipping distance can be enhanced by reducing useless skipping of possible longer posting lists, which also leading to less the sorting as only on one posting list. Thus, the sorting can be partially done on just one skipped over posting list by one insertion sort.

The performance of our updated version of WAND can be further improved by adding an additional online maximum achievable score check during document scoring where the current upper bound score is lowered by subtracting the per-term upper bound from the current estimated document score, and adding the real document score once it is computed. As shown in Algorithm 1, P.WAND applies an revised upper pivot selection strategy with almost the same that used in the original WAND, except considering the equality of the original pivot docID and the current docID of its following term if it exists (line 4–10), which is very helpful in partial scoring phase for considering all the term occurrence within a given candidate document. This metric can avoid the current document equality of the pivot term and its followings, without which will cause a possible scoring flaw that omits the contribution of the query term after pivot if its current document equals to the pivot document [3]. If a posting list points to the candidate, just add its contribution to the real score using a simple ranking function and updated the estimated maximum score for partial scoring checking (line 18, 22). As soon as the current score drops below the discarding threshold, scoring can early terminate and all posting pointers can be updated accordingly (line 19–20). When all of the posting lists are considered for d, we insert the candidate d into the result heap (line 25), and some necessary update operation should be conducted and all posting lists should be fully sorted (line 26). Finally, we return the k top scored documents in the result heap (line 28).

Algorithm 1. Partial Scoring WAND (P.WAND)

```
Input: inverted index iterators $I = \{I_{t_1}, ..., I_{t_m}\}(m = |q|)$ sorted by ascending current
    docID, maximum score $u_{t_i} = \widehat{s}(I_{t_i})(1 \le i \le m)$.
Output: top-$k$ query results sorted by descending score.
 1: $heap \leftarrow \emptyset$
 2: while $m > 0$ do
 3:     $score \leftarrow 0$; $pivot \leftarrow 1$; cumulative maximum score: $cmscore \leftarrow 0$
 4:     for $pivot \le m$; $pivot{+}{+}$ do
 5:         if $cmscore + \widehat{s}(I_{t_{pivot}}) \ge heap.\theta$ and $I_{t_{pivot}}.docid < I_{t_{pivot+1}}.docid$ then
 6:             break;
 7:         else
 8:             $cmscore \leftarrow cmscore + \widehat{s}(I_{t_{pivot}})$
 9:         end if
10:     end for
11:     for $i \leftarrow pivot - 1$; $i > 0$ ; $i{-}{-}$ do
12:         $I_{t_i}.skipTo(I_{t_{pivot}}.docid)$
13:         if $I_{t_i}.docid > I_{t_{pivot}}.docid$ then
14:             proceed to line 26
15:         end if
16:     end for
17:     for $i \leftarrow pivot - 1$; $i > 0$ ; $i{-}{-}$ do
18:         $score \leftarrow score + s(I_{t_i})$
19:         if $score + cmscore < heap.\theta$ then
20:             proceed to line 26
21:         else
22:             $cmscore \leftarrow cmscore - \widehat{s}(I_{t_i})$
23:         end if
24:     end for
25:     $heap.insert(I_{t_{pivot}}.docid, score)$
26:     increment used and remove empty iterators $I_{t_{i \le pivot}}$, update $m$ and $sort(I)$
27: end while
28: return the results in $heap.descOrder()$.
```

3.3 Less Sortings in and Mode

For static definition, the AND operator is defined as $AND(x_1, x_2, ...x_m) \equiv WAND(x_1, 1, x_2, 1, ...x_m, 1, m)$ (threshold $\theta = m$ is fixed for a given query q). With a predefined AND query filter, we can conduct an faster query processing. However, web search engines calculate all term contribution as the weight of indicator with ranking model. Thus, the discarding threshold of the top-k result heap is increasing with the query processing continues. At some point, the discarding threshold is large enough, and a candidate document should include all the query terms such that has a chance to enter the top-k result heap. This will happen only when the discarding threshold is larger than the accumulated upper bound score of all the query terms minus the smallest upper bound. That is to say, the WAND operator enters the AND processing mode only if $\theta > u_{t_2}x_2 + ... + u_{t_m}x_m$ (assume u_{t_1} is least value of all u_{t_i}). For example, in Fig. 2, when the discarding threshold is larger than 6.5 $(2.0 + 3.0 + 1.5 = 6.5)$, the document that does not appear in t_2 with the smallest upper bound score (1.2) will never enter the top-k result heap. That is to say, only candidate document appears in all the four query terms will have a chance to enter the result heap. Then, the WAND algorithm can be seen as a AND model query processing with different candidate selection and partial scoring optimization heuristics, which will lead to less sortings for WAND(L.WAND).

The AND mode query processing procedure can also be separated into three phases: candidate selection, postings skipping and candidate scoring. The explicit implementation of the candidate selection and partial scoring phases is quite different from that in WAND_IS. The costly sorting can be reduced by finding the maximum current docID of all posting lists in candidate generation. The shortest posting list can be used to efficiently skip through the longer posting lists and thus reduce the amount of data to be processed in AND mode. Further, the partial scoring strategy can also be used in scoring the candidate document, which avoids scoring the rest of the postings for a document if the current score plus the accumulative maximum score of the remaining terms falls below the discarding threshold. We only need to recalculate a fixed accumulated upper bound score array as the sorting for all query terms is reduced, and the relative ordering of all terms remains unchanged.

The AND Mode optimization heuristic that combines less sorting and partial scoring further lead to performance improvement for WAND processing. In Fig. 2, prior to further query processing, we order the iterators by ascending upper bounds and calculate their accumulative upper bounds array from first to last (3.0, 5.0, 6.5, 7.7) for the rest query processing procedure. The current discarding threshold (6.6) is larger than the total cumulative upper bound score minus the smallest one (2.0 + 3.0 + 1.5 = 6.5). The AND mode optimization should be applied, and the maximum current docID (180) is directly selected as the next candidate. All skipping is conducted and checked if all the current docID are aligned. After that, partial scoring strategy is applied to the candidate (180) for sorted posting lists (from term t_3, t_1, t_4 to t_2). When the partial score of term t_1 is computed, the current maximum achievable score of the candidate

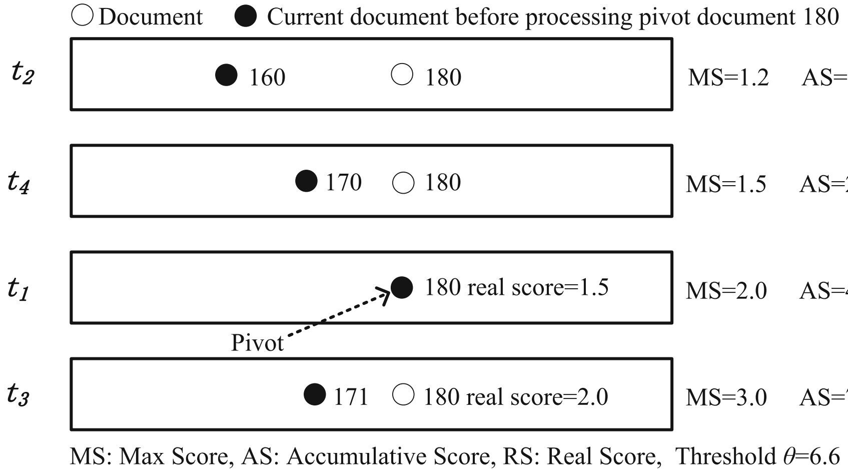

○ Document ● Current document before processing pivot document 180
t_2
160
180
MS=1.2 AS=1.2
t_4
170
180
MS=1.5 AS=2.7
t_1
180 real score=1.5
MS=2.0 AS=4.7
Pivot
t_3
171
180 real score=2.0
MS=3.0 AS=7.7
MS: Max Score, AS: Accumulative Score, RS: Real Score, Threshold θ=6.6

Fig. 2. AND mode optimization in L.WAND.

is 6.2 ($2.0 + 1.5 + 2.7 = 6.2$), which is smaller than the discarding threshold, and the scoring of the rest two query terms (t_4 and t_2) for the candidate can be eliminated.

4 Experiments

In our experiments, we use the standard TREC GOV2 collection containing approximately 25.2 million documents and approximately 32.8 million terms in its lexicon. We build posting lists and self-index structure with the PForDelta compression [8], removing standard English stopwords, and applying Porters English stemmer. The final compressed index size is 6.72 GB. We also build another index without skipping and find that the hierarchical self-index only adds about 82.1 MB to the index size, that is a 1.19% increase. We separate TREC2005 Efficiency Track Queries into those with length from 2 to 5 and those with length larger than 5, and random select 1000 queries for each query length. We use standard Okapi BM25 as the ranking function [12] with parameters: $k_1 = 1.2$ and $b = 0.75$, as default. The Okapi BM25 bag-of-words similarity scoring function as a reference point for $S(q, d)$ is based on their predicted relevance to the query. In addition we compute the upper bound of each term in lexicon on its maximal contribution to any document score, such that $u_{t_i} = max(s(t_i, d_1), s(t_i, d_2), ...)$, and store them along with the lexicon file.

Our experiments were performed on an Intel(r) Xeon(r) E5620 processor running at 2.40 GHz with 8 GB of RAM and 12,288 KB of cache. The total index is located on-disk, and random accessed and loaded into memory when performing querying. In every experiment the results are averaged over 10 independent runs. We measure the performance by average time per query in milliseconds(ms), decoded blocks, posting lists sortings (only in WAND), docID evaluated and calls to the scoring function. The default number of documents retrieved for each query equals 10 that a result page of the common search engine contains. As we focus on rank safe query optimization techniques, the effectiveness measurement is ignored in the following experiments. But we carefully checked the effectiveness of all the top-k processing algorithms to ensure the rank safe premises for all experimental results.

4.1 Different WAND Descriptions

We present an explicit experimental comparison of WAND and MaxScore heuristics, including the original WAND_FS and our updated description of WAND_IS. Table 1 shows the average query processing time of the three dynamic pruning techniques with different query lengths. As can be seen, the two WAND heuristics both outperform MaxScore in query latency for two query terms. But MaxScore achieves better performance than WAND_FS but worse performance than WAND_IS with other query lengths. This verifies our analytical conclusion that the partial scoring strategy plays very important role in MaxScore. Our description of WAND, i.e., WAND_IS achieves better query processing performance than WAND_FS and MaxScore with all query length. With the increase of the query length, all pruning techniques consume more query time without question, but MaxScore outperforms WAND_FS when the query length is larger than 2. That is because the large number of full posting lists sortings in WAND_FS slow down the total evaluation process for longer query length. All above verify the effectiveness of our one skipping over description which can reduce the short distance skippings to unpromising candidates.

Table 1. Average query latency of dynamic pruning techniques for different numbers of query terms.

Algorithm	Average query latency (ms)					
	Avg	2	3	4	5	>5
MaxScore	30.7	21.6	29.8	37.0	43.3	60.2
WAND_FS	30.9	18.8	30.0	38.7	46.2	69.0
WAND_IS	28.0	18.5	27.4	34.8	39.8	56.8

Table 2 shows other criteria for the three pruning techniques. The scoring functions called for WAND series of pruning methods are half of that for MaxScore. This is because the optimizations on MaxScore relies heavily on partial scoring strategy as the analytical results above. Our improvement lies in the posting skipping, not in valid scoring candidates, thus the scoring functions called is unchanged. The docIDs evaluated in MaxScore only comes from the processing of the candidate documents, but in WAND it also contains document-level candidate filtering in pivoting. Thus, the docIDs evaluated in WAND are more than that in MaxScore. Our proposal reduces this criteria by avoiding further postings visitings if one skipping over is detected. The relative long distance skipping in WAND_IS can result in significant less posting lists sortings, and thus less docID visitings for candidate document generation. The block decoded number is slightly increased for both WAND methods than MaxScore. This is because the partial scoring strategy is twisted combined with partial skipping strategy, which eliminate further skipping if the candidate gets a lower score of less than the threshold. But the WAND_FS method conduct full skipping, and our proposal only detect skipping over scenario but not lower scored candidate, which leads to very slight improvement compared to WAND_FS method. In the following paragraphs we take WAND_IS description as the default WAND implementation.

Table 2. Average number of processed elements of different dynamic pruning techniques.

Algorithm	N_{sc} $(\times 10^3)$	N_{de} $(\times 10^3)$	N_{ls} $(\times 10^3)$	N_{bd}
MaxScore	226.8	251.4	–	41.2
WAND_FS	120.4	436.5	748.9	42.7
WAND_IS	120.4	363.2	496.9	42.7

N_{sc}: number of scorings called; N_{de}: number of docIDs evaluated; N_{ls}: number of lists sortings; N_{bd}: number of blocks decoded.

4.2 Optimizations on WAND

Table 3 shows the average query processing time of the four dynamic pruning techniques with different query lengths. The results demonstrate the effectiveness of our further improvements on WAND, and PL.WAND achieves best performance compared against other three methods. This is mainly due to the superiority of partial scoring and AND mode optimizations for WAND method. The query latencies of all pruning techniques increase with the query length. The performance gap between PL.WAND and other techniques reduces but the relative improvements remain unchanged with the increase of query length. This is because for long query length, one small real per-term contribution has quite less impact on the maximum achievable score of a given candidate document, and partial scoring checking is prone to be false that may slow down the performance gains. Also, the absence of one or more query terms is quite common for result document with long query length than short query length. Thus, the AND mode optimization might not take effect for long query length. Overall, our PL.WAND algorithm achieves much faster query processing than our basic description of WAND method by 6% and the original WAND method reported above by 15% on average.

Table 4 shows other criteria for the four pruning techniques that provide an in-depth demonstration of the query performance. From Table 4, we can see that

Table 3. Average query latency of dynamic pruning techniques for different numbers of query terms.

Algorithm	Average query latency (ms)					
	Avg	2	3	4	5	>5
WAND	28.0	18.5	27.4	34.8	39.8	56.8
P.WAND	27.8	18.2	26.9	34.1	39.5	57.6
L.WAND	26.6	16.9	25.5	33.6	39.4	57.5
PL.WAND	26.2	16.3	25.1	33.3	39.4	57.5

all techniques improve significantly over our WAND method. All three methods reduce the scoring functions called by almost 30%, and evaluated less docIDs compared against WAND. The indicates that the partial scoring strategy can lead to less scoring functions called for our WAND description and less docID evaluated when the candidates are discarded with a partial score. The posting lists sortings are also reduced for all three methods, and especially for L.WAND and PL.WAND the reduction can be almost half of our description of WAND (WAND_IS) and one third of the original description of WAND (WAND_FS). This demonstrates the effectiveness of AND mode optimization for WAND. In our P.WAND method, the partial scoring strategy is not twisted implemented with one skipping over detection, a full skipping and aligning is done after the skipping over detection, which has no effect on the following scoring procedure. That is to say, all the posing lists are prepared the same for both partial scoring and full scoring strategy. Thus, the blocks decoded are the same for all dynamic pruning techniques. That means optimizations on candidate documents cannot avoid block decodings, as the skipping distance may be no longer than a block length, and the decompression reduction cannot be avoided.

Table 4. Average number of processed elements of different dynamic pruning techniques.

Algorithm	N_{sc} ($\times 10^3$)	N_{de} ($\times 10^3$)	N_{ls} ($\times 10^3$)	N_{bd}
WAND	120.4	363.2	496.9	42.7
P.WAND	79.2	336.9	473.9	42.7
L.WAND	89.0	340.9	258.5	42.7
PL.WAND	79.0	333.6	247.0	42.7

N_{sc}: number of scorings called; N_{de}: number of docIDs evaluated; N_{ls}: number of lists sortings; N_{bd}: number of blocks decoded.

Real search engines use ranking functions based on hundreds of features. However, such functions are quite expensive to evaluate. To achieve efficiency, search engines commonly separate the ranking process into two or more phases.

In the first phase, a very simple and fast ranking function is used to score the full postings that match the query and then return the top-k scored documents. In the following phases, increasingly more complicated ranking functions with more and more features are applied to documents that pass through the earlier phases. Thus, the later phases only examine a fairly small number of result candidates, and a significant amount of computation is still spent in the first phase. A simple and fast ranking function is adopted to obtain the top 100 or 1000 rough documents in the first phase. However, different dynamic pruning techniques have their own performance superiority with different number of returned results. Thus, it is very important to evaluate the performance of our proposed methods with different numbers of returned results to adapt different search engine needs.

Table 5. Average query latency of dynamic pruning techniques with different numbers of results.

Algorithm	Average query latency (ms)				
	$k = 10$	$k = 50$	$k = 100$	$k = 500$	$k = 1000$
MaxScore	30.7	35.2	38.0	47.3	52.1
WAND_FS	30.9	35.4	36.7	44.9	48.6
WAND	28.0	31.8	33.7	43.1	47.0
P.WAND	27.8	31.5	33.4	42.5	46.8
L.WAND	26.6	31.0	33.1	42.5	46.7
PL.WAND	26.2	30.8	32.8	41.0	46.7

Table 5 shows the experimental results of the 6 dynamic pruning techniques as we increase the number of results for a first phase ranking. We find that the query latency of the six techniques increases with k and that PL.WAND achieves better stable performance than the other techniques with different k. As can be seen, the two WAND methods achieve better average query latency than MaxScore with all different returned results k on the randomly selected queries. And the performance gap between MaxScore and WAND_IS becomes larger with the increase of the number of results (from 8% when $k = 10$ to 20% when $k = 1000$). That is because when the threshold of the result heap keeps a lower value for a larger number of results, the absence of one query term or small per-term contribution has quite less impact on the following evaluation of candidate document, thus the effect of partial scoring strategy adopted in MaxScore is highly reduced. This further indicates that WAND is more suitable than MaxScore for a large number of returned results. With the increase of the number of results, we can find that the performance gap between PL.WAND and other techniques is reduced, especially for the last three optimization method on WAND. The reason is also that advantage of partial scoring strategy without AND mode may not work when the threshold remains a small value with large number of returned results. Thus, the effect of partial scoring adopted in P.WAND and L.WAND is both reduced. Furthermore, with a lower discarding threshold, the AND mode optimization might come late for posting lists traversal. The slight improvement of L.WAND arises from too less candidates containing all query terms.

4.3 On Block-Max Index

As fine-granularity local block max score provide more accurate score estimation for measurable partial scoring improvements, we further extend our work to the Block-Max WAND processing with fixed-sized blocks for online estimating the maximum achievable score of the candidate document more precisely. The Block-Max score is computed for every 128 postings and the extra Block-Max file added about 200 MB to the index size. We adopt the block-max technique proposed in [4] to our optimization metrics, i.e., filtering candidate generated in pivoting phase and early terminating scoring phase. The experimental in Table 6 with top-10 query results shows the superiority of the proposed metrics. All the Block-Max based methods are named by adding BM_ for the methods mentioned in previous experiments. As can be seen, the Block-Max technique added at least 2× performance improvement for all the WAND-based methods. The improvement becomes more significant for larger query length, as the partial scoring and less sorting heuristics can get more precise local achievable score estimation with Block-Max maxscore. The BM_PL.WAND achieves the best performance among all methods, which proves that our optimization heuristics can be applied to Block-Max based method for further improvement.

Table 6. Average query latency of the WAND optimizations on block-max indexes for different numbers of query terms.

Algorithm	Average query latency (ms)					
	Avg	2	3	4	5	> 5
BM_WAND_FS	15.2	10.1	13.9	18.2	20.4	25.3
BM_WAND	19.0	9.5	13.9	17.9	19.9	23.8
BM_P.WAND	18.3	8.3	12.4	16.3	18.5	21.9
BM_L.WAND	18.1	8.7	12.6	16.8	18.9	22.3
BM_PL.WAND	14.2	8.0	12.1	16.3	18.2	20.1

5 Conclusions

In this paper, we focus on the widely used dynamic pruning method WAND for faster query processing reported in previous works. We present an efficient immediately skipping over description of WAND implementation which can highly reduce short distance skippings to unpromising candidates. We further propose partial scoring optimization and AND mode optimization on our description of WAND which can reduce costly scoring functions called and posting lists sortings. Empirical study on TREC GOV2 datasets with self-index and Block-Max index indicates that our description of WAND significantly improve performance compared against the original common used WAND version, and the partial scoring and AND mode optimizations yield further performance gains, with a best total improvement of about 20%.

The dynamic pruning techniques relies heavily on longer skipping distance for random accessing indexes, but the block-based decompression may cut down the optimization gained from skipping. Thus, how to build a proper accessing predicted self-indexing structure for each block-compressed posting is an open challenge for faster query processing.

References

1. Bortnikov, E., Carmel, D., Golan-Gueta, G.: Top-k query processing with conditional skips. In: Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, 3–7 April 2017, pp. 653–661 (2017)
2. Broder, A.Z., Carmel, D., Herscovici, M., Soffer, A., Zien, J.Y.: Efficient query evaluation using a two-level retrieval process. In: Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management, pp. 426–434 (2003)
3. Crane, M., Culpepper, J.S., Lin, J.J., Mackenzie, J., Trotman, A.: A comparison of document-at-a-time and score-at-a-time query evaluation. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, pp. 201–210 (2017)
4. Ding, S., Suel, T.: Faster top-k document retrieval using block-max indexes. In: Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 993–1002 (2011)
5. Fontoura, M., Josifovski, V., Liu, J., Venkatesan, S., Zhu, X., Zien, J.Y.: Evaluation strategies for top-k queries over memory-resident inverted indexes. PVLDB **4**(12), 1213–1224 (2011)
6. Jonassen, S., Bratsberg, S.E.: Efficient compressed inverted index skipping for disjunctive text-queries. In: Clough, P., et al. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 530–542. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20161-5_53
7. Macdonald, C., Ounis, I., Tonellotto, N.: Upper-bound approximations for dynamic pruning. ACM Trans. Inf. Syst. **29**(4), 17:1–17:28 (2011)
8. Moffat, A., Petri, M.: Index compression using byte-aligned ANS coding and two-dimensional contexts. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 405–413 (2018)
9. Petri, M., Culpepper, J.S., Moffat, A.: Exploring the magic of WAND. In: The Australasian Document Computing Symposium, ADCS, pp. 58–65 (2013)
10. Petri, M., Moffat, A., Culpepper, J.S.: Score-safe term-dependency processing with hybrid indexes. In: The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 899–902 (2014)

11. Strohman, T., Turtle, H.R., Croft, W.B.: Optimization strategies for complex queries. In: SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 219–225 (2005)
12. Turtle, H.R., Flood, J.: Query evaluation: strategies and optimizations. Inf. Process. Manage. **31**(6), 831–850 (1995)
13. Wang, Q., Dimopoulos, C., Suel, T.: Fast first-phase candidate generation for cascading rankers. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 295–304 (2016)

Query-Based Recommendation by HIN Embedding with PRE-LSTM

Zhuo-Ming Liu[1,2,3], Yu-Miao Hui[1,2,3], and Ling Huang[1,2,3](✉)

[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
huanglinghl@hotmail.com

[2] Guangdong Province Key Laboratory of Computational Science, Guangzhou, China

[3] Key Laboratory of Machine Intelligence and Advanced Computing,
Ministry of Education, Beijing, China

Abstract. In recent years, as more and more big data platforms have been applied to the government network systems, it's essential to adopt an effective query-based recommendation algorithm to help officers find out the needed table with a keyword. The key challenge lies in the complex relationship among data from different departments, which cannot be easily solved by the existing database query methods. The Heterogeneous Information Network (HIN) is a specific type of networks developed for modeling complex data relations. However, these existing query-based recommendation algorithms could not make use of HIN. Besides, many query-based recommendation algorithms could not make recommendations with the keyword that is not in the query records. In addition, most of the existing recommendation algorithms do not make full use of the semantic meanings. Therefore, in this paper, by making use of the real dataset provided by the local government, the proposed method is the first to use the pretrained word embeddings and LSTM (PRE-LSTM) to train the network embeddings and to learn the relationship among the tables, departments and the keywords, so that it can make use of the data in HIN and enable the network embeddings to obtain the most precise semantic meaning. Additionally, our algorithm uses word embeddings to represent query keywords so as to let our algorithm make the query-based recommendation for nearly any query. Using the trained embeddings and the PRE-LSTM model, the proposed algorithm is able to show user-specific recommendation results sorted in a reasonable order. Experimental results on the real data application tasks confirm the effectiveness of the proposed method.

Keywords: Recommender systems · Query-based recommendation · Heterogeneous information network · LSTM

1 Introduction

With the rapid data accumulation on the Internet, the big data platforms, which are aiming for sharing information between different departments, have

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 515–529, 2020.
https://doi.org/10.1007/978-3-030-65390-3_39

Table 1. Example of raw data.

Table name	Attributes	Subscription department
Charitable organization Name	Location, Advisor department, Liaison, Contact number	Meteorological bureau, Court, Administration data bureau, District government 1, District government 3
Taxpayer credit rating info	Name, Registration number, Registration type, Legal representative, Address, Credit rating, Assessment date and filling date	Administrative service center, Bureau of natural resources, Discipline inspection commission, Municipal supervision bureau, Finance office, Development and reform bureau, Court, Administration bureau, Procuratorate, Police
Natural person info	Citizen number, Name, Certificate number, Certificate type, Position, Operation type, Creation time, Updating time	Administration bureau, Court, Discipline inspection commission, Administrative service center, Talent office, Tax bureau, Urban management bureau, Police, Procuratorate

been widely used in government for enhancing the working efficiency. Since the establishment of the platform, thousands of tables have been collected from all departments, and numerous cross-department data utilizations have been realized. When a department needs to find a table, a keyword would be given for a query on the platform, and some related tables would be recommended. For instance, when the department **Police** uses the keyword **Name** to search for a table, the tables that are closely related to the population registration should be recommended, e.g. the **Table for Transient Population** and the **Table for Basic Information of Population**. To help the officers find out the tables that fit their needs effectively, we need a powerful query-based recommendation algorithm to accomplish the task.

The most naive method for query-based recommendation is the PageRank [3]. But it could not provide personalized recommendation results. In recent years, various methods have been proposed to give out personalized recommendation results for the query-based recommendation. One of the typical methods is the record-query-based algorithm [4,7], which either finds some similar queries as the results for the recommendation, or uses the pretrained query texts as an item for the recommendation. Another typical type of approaches uses the personalized PageRank for recommendation [10]. Some make recommendations by solving the optimization problems in query flow graphs [17,22]. Others consider the relationships of different networks and obtain the optimal recommendations by using matrix factorization [23]. With the wide-range application of the neural networks in recent years, some choose GRU and other neural networks to concatenate the features of different information for personalized recommendation results [25]. However, for most of these methods, they either need interactions with users or make recommendations only when the query has existed in the training data. This indicates that if the query keyword never appears in the query records, the algorithm could not give out a recommendation, even though the keyword is meaningful.

Besides, to improve the recommendation performance, many efforts have been made to utilize any available information. Especially in the era of big data, it is important to integrate different types of data, and the data structures in the database that become more and more complex. Therefore the concept of Heterogeneous Information Network (HIN) appears [16,21], after which, many algorithms have been proposed to train the network embeddings for the recommendation. Some use the auto decoder [9,19,24] to train the embeddings. Some use the matrix factorization [13] to obtain the vector representations of the items. Some use the RSGD to optimize the objective function of embeddings [20]. Others use GRU to train the embeddings [5]. But none of them tries to use HIN in a scenario of query-based recommendation, and few use neural networks to train the embeddings and represent the items in HIN with semantic meanings.

Furthermore, although network embeddings have been widely used for the recommendation, the embeddings could not represent the accurate semantic meanings in the real world. This is because the semantic meanings of network embeddings obtained from meta-paths are sparse compared with the relationship of words in natural language.

Therefore, we aim to solve the query-based recommendation problem by fitting in with the data structure we have obtained. In particular, an algorithm is developed for the query-based recommendation in HIN. By proposing a new method to train the network embeddings and adopting a new representation for the query, our algorithm could respond to nearly any query and give out meaningful recommendation results. Besides, our proposed method tries to make use of the mechanism in NLP, so as to make the network embeddings contain more semantic meanings and the structural characteristics by representing the items in HIN as words in the real world. Thus, it enhances the accuracy of recommendation.

The key contributions of this paper can be summarized as follows:

1. We propose the PRE-LSTM model that uses the pretrained word embeddings and LSTM to train the network embeddings and learn the relationships among the tables, departments and the keywords, so as to make the network embeddings obtain more precise semantic meanings which generates satisfactory recommendation results.
2. Our proposed method successfully converts a query and its source department to a meaningful sequence by making use of the trained network embeddings, solving the HIN query-based recommendation problem which was merely mentioned before.
3. Our algorithm represents the query keyword as a word embedding. By making use of the linear relationship of the word embeddings, the proposed method can represent any query keyword. Thus, it can give out recommendation results for nearly any query.

2 Problem Description

2.1 Data Description

In this work, we adopt the real dataset provided by one local government in China, Guangdong, Foshan city. After the data masking process, the dataset contains 41 departments, 97 data tables, and 1123 distinctive attributes in the tables. Besides, we can gain information about the subscription departments of each data table. Some examples of the raw data are shown in Table 1. All the names of tables, departments, and attributes are Chinese words. We translate them into English in this paper for better understanding.

2.2 Challenges and Problem Analysis

The current recommendation method used in the real big data platform selects all the tables containing the query keyword as their attribute or as the subword of their attribute through keyword matching, returning the result to the platform front-end without sorting. This method is suitable for dealing with a small number of tables. However, with the increasing data collected by the big data platform, the number of tables will increase continuously. This not only costs more time for matching query keyword with the table attributes but also makes the recommendation accuracy drop sharply. At the same time, because the platform does not display the recommended tables in a reasonable order, users need to look for more than ten pages to find the needed tables. To overcome all these shortcomings, some challenges should be taken into consideration as follows:

1. With the growing number of tables in the big data platform, it's not possible for users to traverse all the tables to compare the keywords in query with the attributes in the tables.
2. The currently used method could not respond to the keyword which is not an attribute or a subword of the tables' attribute.
3. We need to give out a meaningful sorting for all the possible recommendation results.

In our query-based recommendation usage scenarios, we need to recommend a series of T`ables` to a specific D`epartment` based on a given K`eyword`. Therefore, the problem can be simplified as follows: We recommend the third element T in the triple (D, K, T) when the first two elements (D, K) are given. This is similar to a common scenario in NLP. When the first few words in the sentence are given, the NLP task generates next words in a sentence with the model. So we need to train the model to learn the relationships among the tables, departments, and the keywords. Since HIN is a good choice for modeling complex relationships, we need to use HIN to describe our data.

Thus, our task could be divided into two parts: (1) training the network embeddings of HIN for representing the tables and departments, and training the model that could learn the relationships among the tables, departments and the keywords, and (2) using the model and the embeddings for query-based recommendations so that we can overcome the mentioned challenges above.

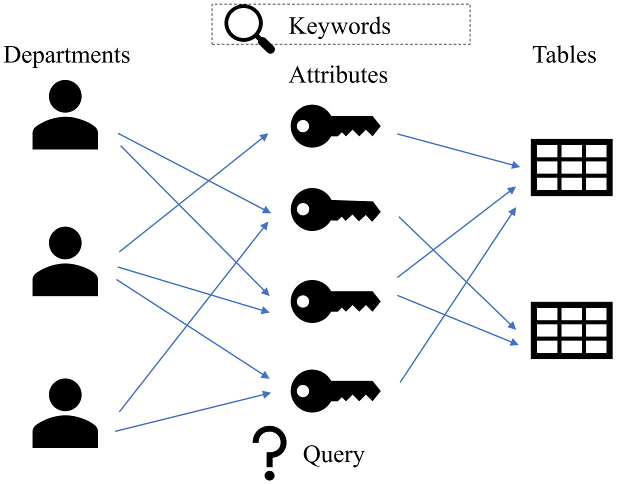


Fig. 1. The HIN constructed from the dataset.

3 Model and Embedding Training

3.1 Transferring Raw Data into HIN

The heterogeneous information network (HIN) [14–16] is a special kind of information network, which contains either multiple types of objects or multiple types of edges. As shown in Fig. 1, we have represented the mechanism of query-based table recommendation systems for departments by HINs. The HIN consists of multiple types of objects, including Department (D), Keyword (K), Attribute (A) and Table (T). There are different types of edges between objects to represent different relations. A department - keyword (attribute) edge indicates the department uses the keyword or attribute to search the most relative items, and it exists only when the keyword is an attribute or a synonym of an attribute that belongs to the table which is subscribed by the department. A keyword (attribute) - table edge indicates a relation that the keyword is an attribute belongs to the table or the keyword is closely relative to the table, and it exists only when the keyword is an attribute or a synonym of an attribute that belongs to the table.

3.2 Meta-paths Generation

In the literature of HIN, a meta-path is an important concept to characterize the semantic patterns for HIN. To generate meaningful node sequences for the procedure of model training, it's necessary to adopt an effective walking strategy that can capture the complex relationships in HIN. Hence, inspired by [6,8,13,18], we use the meta-paths based on random walk method to generate node sequences.

Table 2. Example for generated meta-path.

D	$K(A)$	T
Statistics bureau	Industry categories	Bussiness registration info
Court	ID number	Transient population info
Police	Name	Tax credit rating info
Tax bureau	Name	Natural person info

In our recommendation scenarios, every table in HIN has attributes, and is subscribed by some departments. These relationships indicate that departments have preferences to check the attributes in the table. Therefore, the departments which have subscribed to a table would have a higher possibility than other departments to use the attribute or the similar words of the attribute as the keywords to look for tables on the big data platform. So we generate the meta-paths as follows:

$$(D, K(A), T) \tag{1}$$

In the Heterogeneous Information Network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ used in our work, $\mathcal{V}$ in $\mathcal{G}$ is divided into three types, namely, Department Node N_D, Table Node N_T and Keyword (Attribute) Node $N_{K(A)}$. Every meta-path starts from N_D, passing through $N_{K(A)}$, and reaches the last element of the meta-path in N_T. The meta-path can be described as follows:

$$N_D \xrightarrow{R_{DK(A)}} N_{K(A)} \xrightarrow{R_{K(A)T}} N_T \tag{2}$$

where $R_{DK(A)}$ means the edges between the departments and the keywords (attributes), and same for $R_{K(A)T}$. The meta-paths generation follows the following probability distribution:

$$P(n_{t+1}=x|n_t=v)=\begin{cases}\frac{1}{\mathcal{N}^{N_{K(A)}}(v)}, (v,x)\in R_{DK(A)} \& v\in N_D\\ \frac{1}{\mathcal{N}^{N_T}(v)}, (v,x)\in R_{K(A)T} \& v\in N_{K(A)}\\ 0, otherwise\end{cases} \tag{3}$$

where n_t is the tth node in the walk, $t = 1, 2, 3$. $\mathcal{N}^{N_T}(v)$ is the first-order neighbor set for node v with the type of N_T, and similar for $\mathcal{N}^{N_{K(A)}}(v)$. The example of meta-paths is shown in Table 2.

3.3 Training Model

We choose the LSTM as our training model. The dimension of input vectors and the hidden states are 200 and 400 respectively in our model. We choose LSTM rather than RNN as the neural network model because the forget gate or the input gate of LSTM could help the model ignore some specific elements in the meta-paths, and thus we could enhance the relationship between the specific two elements in the triple. For example, we can enhance the relationship between a specific table and a specific department but weaken the role of a keyword in the recommendation. If the **Water Supply Bureau** searches for a table on the platform, the resulting recommended tables should be closely related to water but weaken the influence of the keyword used in the query. In contrast, if the department is the **Court**, which plays an important role in different sectors of the society, it is better to strengthen the role of the keyword in the recommendation to determine which table should be recommended. We propose two models to train the embeddings as below.

By making use of the meta-paths generated from a random walk, we first train all the embeddings which are randomly initialized. When back-propagating loss in the neural network, we update all the embeddings in the neural network. We call this model as Ordinary LSTM Model (ORD-LSTM). However, Since the training set only makes use of some words in our daily life and the meta-paths in HIN are sparse compared with the relationship of words in natural language, the embeddings trained by the first model do not contain the precise semantic meaning of the word. In other words, the ORD-LSTM could not make use of the potential relationships, stored in the word itself, among the tables, departments and the keywords to make recommendations.

We use the pretrained embeddings as the initialized values of all embeddings in the neural network and add structural characteristics of HIN into embeddings by training the model with meta-paths. However, we find the pretrained vector space S would be changed, and the semantic meanings in the embeddings would be modified when back-propagating the loss to all the embeddings, degrading the performance of the model. Also, after vector space S being changed, the linear relationship of the embeddings loses. Thus, we are not able to provide recommendations for the keywords that have not been trained.

To maintain the vector space which contains rich semantic meanings and train the embeddings of the departments and tables, we propose the Pretrained LSTM Model (PRE-LSTM) and divide the embeddings in the neural network into two independent modules, E_{word} and E_{item}. Both embeddings belong to the same vector space S. The E_{word} is constituted by the pretrained embeddings of the keywords in the generated meta-paths. The E_{item} represents the embeddings for the department names and the table names. The algorithm maps E_{item} into S, which is constituted by E_{word}, through training LSTM model with meta-paths. When back-propagating loss in the neural network, we only update E_{item} and keep E_{word} unchanged. The training pseudocode is shown in Algorithm 1. As shown in experiments, by using PRE-LSTM, the word embeddings can better represent the semantic meanings compared with the word embeddings trained by ORD-LSTM.

Algorithm 1. The Algorithm for Training PRE-LSTM

```
$E_{item}$.initialize()
$E_{word}$.load($pretrained\_embedding$)
$E_{word}$.requiresgrad = False
for $i$ in range(epoch) do
   for $batch$ in $TrainingSet$ do
      $data$ = set of (D,K) in each meta-path in $batch$
      $target$ = set of T in each meta-path in $batch$
      $embed$ = embedding($data$)
      $output$ = lstm.Forward($embed$)
      $result$ = linear($output$)
      $loss$ = criterion($result$, $target$)
      Back propagate $loss$
      if para.requiresgrad == Ture then
         Update the parameter
      end if
   end for
end for
```

4 Recommendation Framework

Different scenarios need the algorithm to response to different kinds of keywords. The keywords could be divided into three types:

1. The keyword embedding in E_{word}.
2. The keyword embedding not in E_{word}, but in pretrained word embeddings.
3. The keyword embedding not in E_{word} and the pretrained word embeddings.

4.1 Preliminary for Recommendation

We deal with different scenarios with diverse methods. For the word not in E_{word} and the pretrained word embeddings, it is usually long and can be divided into a few subwords.

To find out the embedding of a long word, we make use of the linear relationship of the word embeddings. The linear relationship was found by Tomas Mikolov [11,12] and proved by Carl Allen [1] and Sanjeev Arora [2]. Thus, the embedding of a long word $embed_{word}$ can be represented by the sum of the embeddings of the subwords $embed_{subword_i}$ as follows:

$$embed_{word} = \sum_{i=1}^{N} embed_{subword_i} \tag{4}$$

where N denotes the number of subwords that the original word can be divided into.

Algorithm 2. The Algorithm for Recommendation

```
$department\_embed = E_{item}[department.\text{index}]$
$word\_embed = \mathbf{0}$
if $word.\text{index}$ in $E_{word}.\text{key}()$ then
    $word\_embed = E_{word}[word.\text{index}]$
else
    if $word.\text{index}$ in $PretrainEmbed.\text{key}()$ then
        $word\_embed = PretrainEmbed[word.\text{index}]$
    else
        $word\_embed = \text{CalculateEmbeddingForLongWord}()$
    end if
end if
$output, hidden = \text{lstm.Forward}(department\_embed, \mathbf{0})$
$output, hidden = \text{lstm.Forward}(word\_embed, hidden)$
$output = \text{linear}(output)$
$result = output.\text{topk}(10)$
return $result$
```

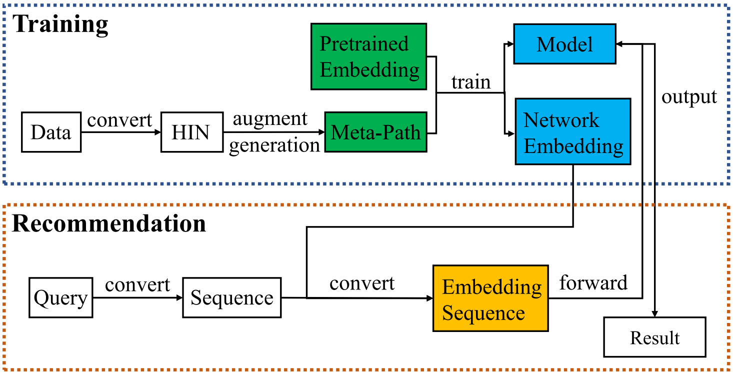


Fig. 2. Procedure for training and recommendation.

By back-propagating loss to some of the parameters and keeping the pretrained word embeddings unchanged through the training, we maintain the vector space S unchanged, so that the word semantic meaning and the linear relationship between the word embeddings are retained. Thus, we can make use of embeddings' characteristics mentioned above to find embeddings for long words. In the following experiments, we will show an example of the recommendation with the third type of keywords.

4.2 Recommendation Algorithm

For the first type of keywords, we find the keywords' embeddings from the E_{word} and pass the embeddings of pair (D, K) through LSTM. Then we can get the recommended tables. For the second type of keywords, by maintaining the vector space S, the pretrained word embeddings could still be used to represent the word not in E_{word}. Thus, we can find out the unique vector representation for a word in the space S by searching the word embedding in the pretrained embeddings set. Algorithm 2 shows the pseudocode for a recommendation when given a keyword and a department name.

In the following experiments, we will show an example of the recommendation with the first and the second type of keywords. The procedure of training and recommendation, as well as its relationship is shown in Fig. 2.

5 Experiment

5.1 Experimental Setting

We train the model with 200 dimensions pretrained word embeddings by minimizing the cross-entropy loss. The embeddings of the departments and the tables are also of 200 dimensions. The model converges in 20 epochs, with a batch size of 256 on an Nvidia GTX-1080 GPU. The top 10 tables are reported as the recommendation results.

5.2 Recommendation Comparison

The whole section is divided into three parts: the evaluation metrics, the recommendations with the query keywords in E_{word} (WIE), the recommendations with the query keywords not in E_{word} (WNIE).

Evaluation Metrics. We compare and report the performance by using the following metrics:

VAL: Validity. We calculate the percent of the recommendation results which are the names of the table instead of the names of departments or keywords in the resulting top 10 recommended tables. This metric represents the quality of the recommendation.

NT: The average Number of Tables in recommendation results. For the method that can give out any number of tables as recommendation results, we note this situation as **10+**. This criterion judges whether the algorithm can provide enough available choices for the users.

TE: The average percent of Target tables Exist in the top 10 recommendation results. This metric indicates whether the algorithm can give out accurate recommendation results.

SR: Sum of Ranking of the target tables. We calculate the sum of the ranking of target tables in the recommendation results. This metric indicates whether the order of the recommended tables is reasonable.

TSK: The average number of Tables with Synonym of Keyword in the recommendation. The table with an attribute that is the synonym of the keyword should be regarded as a meaningful recommendation result. This metric represents whether the recommendation results have generalization ability.

AMD: The Average Minimum Distance between the query embeddings and the recommended tables' embeddings. The query embedding is the sum of embeddings of the query keywords and the embeddings of the departments which have sent the query. The tables' embeddings are the sum of its attributes' embeddings. This metric judges whether the algorithm can provide a recommendation that the semantic meaning is similar to the query keyword and the query department.

All the distances are judged in the same vector space and with the same conversion from item to embedding, in order to ensure the fairness of comparison. For all the metrics except the SR and AMD, higher values indicate better results.

Table 3. Performance comparison for WIE.

Method	VAL	NT	TE	SR	TSK	AMD
PRE_aug	**99.89**	**10+**	**98.35**	**2.67**	**3.85**	**3.01**
PRE	98.89	10+	90.70	3.56	3.79	3.35
PRE_u_aug	99.89	10+	97.89	2.68	3.46	3.19
PRE_u	97.92	10+	91.42	3.27	3.37	3.21
ORD_aug	99.86	10+	97.50	2.67	3.05	3.50
ORD	97.92	10+	91.19	3.31	3.22	3.72
CF	100	10+	16.55	8.86	1.43	4.22
AR	100	3.33	22.89	4.22	3.33	2.61

Table 4. Performance comparison for WNIE.

Method	VAL	NT	TSK	AMD
PRE_aug	**100**	**10+**	**0.77**	**4.17**
PRE	98.41	10+	0.65	4.90
CF	100	10+	0.37	4.39

Recommendation with WIE. In this section, we use 19 thousand triples as testing data. Each triple has a target table, a department and a keyword. We conduct six experiments with two baselines. These six experiments include using the ORD-LSTM model without data augmentation(*ORD*), using the ORD-LSTM model with data augmentation (*ORD_aug*), using the PRE-LSTM model without data augmentation (*PRE*) and using the PRE-LSTM model with data augmentation (*PRE_aug*), using the PRE-LSTM model with unfixed embedding E_{word} without data augmentation (*PRE_u*) and with data augmentation (*PRE _u_aug*). The two baselines for comparison are the recommendation results from Association Rule (*AR*) and Collaborative Filtering (*CF*).

Table 3 shows the performance comparison of different methods. The *PRE_aug* model outperforms two baselines and the other five models which

Table 5. Recommendation for query from BNR with keyword enterprises establishment date.

Method	Table1	Table2	Table3
PRE_aug	Bussiness registration info	Organization code info	Domestic enterprises info
PRE	Bussiness registration info	Domestic enterprises info	Organization code info
PRE_u_aug	Bussiness registration info	Organization code info	Key construction project info
PRE_u	Bussiness registration info	Domestic enterprises info	Organization code info
ORD_aug	Bussiness registration info	Organization code info	Tax credit rating info
ORD	Bussiness registration info	Domestic enterprises info	Organization code info
AR	Bussiness registration info	NA	NA
CF	Enterprise HR info	Real estate enterprises info	Housing construction info

Table 6. Recommendation for query from Talent Office.

Method	Keyword	Table1	Table2	Table3
PRE_aug	Enterprise	Business registration info	Hi-tech enterprises info	High-level enterprises info
PRE	Enterprise	Hi-tech enterprises info	High-level enterprises info	Awarded enterprises info
CF	Enterprise	Enterprise HR info	Real estate enterprises info	Organization code info
PRE_aug	High-tech enterprise	Hi-tech enterprises info	High-level enterprises info	Foreign experts info
PRE	High-tech enterprise	Hi-tech enterprises info	High-level enterprises info	Water fee collection info
CF	High-tech enterprise	Enterprise HR info	Real estate enterprises info	Organization code info
PRE_aug	Talent	Professional personnel info	Transient population info	Population registration info
PRE	Talent	Professional personnel info	Lawyer's basic info	Natural person info
CF	Talent	Enterprise HR info	Real estate enterprises info	Organization code info

do not have the data augmentation or do not use the fixed pretrained word embeddings in terms of NT, TE, SR, and TSK. This indicates PRE_aug can give out more available choices for a recommendation than other methods and the most accurate recommendation according to the given keywords and departments. Besides, it's also able to make recommendations with the highest number of tables that have a synonym of query keyword as an attribute, which enriches the recommendation results and provides more reliable choices for users. In addition, it provides the most reasonable ranking for all the recommended tables.

PRE_aug also has the highest VAL and AMD value in all neural-network-based methods, ensuring the validity and semantic accuracy of the recommendation results. The limited recommendation results make the AR method have the highest score in terms of VAL and AMD, but the user would seriously suffer from a lack of spared recommendation choices.

Recommendation with WNIE. In this section, we choose 1000 queries whose keywords are not in E_{word} as testing data. Table 4 shows the performance comparison. There are only three methods listed in the table since other methods could not be used when the query keywords are not in E_{word}. Besides, since the query with WNIE does not be conducted before, there is no qualified target table for the recommendation. Thus, the metrics TE and SR could not be used in this experiment.

The *PRE_aug* model outperforms the baseline *CF* and method *PRE* in all four used metrics, further testifying our model has an outstanding performance and generalization ability in the recommendation.

5.3 Case Study

Recommendation with WIE. In this section, we choose some queries in which the keyword embeddings are in E_{word} for the case study. Table 5 shows the example of recommendation result for query from **Bureau of natural resources (BNR)** with keyword **enterprises establishment date**. The **BNR** aims to find some information about companies. As shown in the result, the *PRE_aug*, *PRE*, *PRE_u* and *ORD* methods all provide three tables which are closely related to the enterprises' information, outperforming other methods.

The orders of the recommended tables are different in these three methods. It's reasonable that the table **Bussiness registration info** which is subscribed by **BNR** has an attribute **enterprises establishment date** that is ranked to the first place in the recommendation result.

The table **Domestic enterprises info**, which is not subscribed, has an attribute **enterprises establishment date**. The table **Organization code info** which is subscribed but does not have an attribute as same as the keyword, has an attribute that is the synonym of the keyword, the **corporation registration day**. To some extent, the registration day is the legal establishment date of a company. It's reasonable to recommend these tables to the department.

The table subscribed by the department may have a higher possibility of being used, so the table **Organization code info** should be listed in front of **Domestic enterprises info**, indicating that the *PRE_aug* algorithm provides a more reasonable order for the table.

Recommendation with WNIE. In this section, we conduct the case study for the recommendation with the query keyword which belongs to the second or the third type of query keyword. Table 6 shows the recommendation results for query from **Talent Office** using different keywords.

The *PRE_aug* method outperforms the other two methods. The *CF* method gives out three identical results for different keywords since *CF* could not make use of the keywords for the recommendation. At the same time, the *PRE* method gives out some tables which are not related to the characteristics of the departments or the keywords.

For the keyword **enterprise**, *PRE_aug* gives out three results that are closely related to the corporation. Two of them are closely related to technique, which fits in with the characteristic of **Talent Office**.

The keyword **high-tech enterprise** belongs to the third type of query keyword. *PRE_aug* places the tables which are closely related to technique to the first and second places in the list. The table **foreign experts info** has an attribute **current employer name**, which can be used to find out the high-tech enterprises.

For the keyword **talent**, *PRE_aug* recommends three tables which are closely related to personal information. The ideal result, table **Professional personnel info**, is ranked first. The result further shows *PRE_aug* can provide a more reasonable recommendation for some keywords that do not belong to E_{word} than other methods.

6 Conclusions

In this paper, we propose a new method to solve the merely mentioned query-based recommendation problem, which needs to make use of the data in HIN. The key lies in using PRE-LSTM to train the network embeddings of HIN, learning the relationship among the tables, departments and the keywords and converting a query and its source department to the embedding sequence. Besides, by representing the query keywords as word embeddings, our proposed method can make recommendations for any query. The evaluation shows that our method enables the network embeddings to obtain more precise semantic information. Thus, it provides more reasonable recommendation results than other methods and has extensive applicability as shown in the experimental results on the real data application task.

References

1. Allen, C., Hospedales, T.M.: Analogies explained: towards understanding word embeddings. In: ICML 2019, pp. 223–231 (2019)
2. Arora, S., Li, Y., Liang, Y., Ma, T., Risteski, A.: A latent variable model approach to pmi-based word embeddings. TACL **4**, 385–399 (2016)
3. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Netw. **30**(1–7), 107–117 (1998)
4. Cai, X., Han, J., Pan, S., Yang, L.: Heterogeneous information network embedding based personalized query-focused astronomy reference paper recommendation. Int. J. Comput. Intell. Syst. **11**(1), 591–599 (2018)

5. Chen, Y., Wang, T., Chen, W., Li, Q., Qiu, Z.: Type sequence preserving heterogeneous information network embedding. In: AAAI 2019, pp. 9931–9932 (2019)
6. He, Y., Song, Y., Li, J., Ji, C., Peng, J., Peng, H.: HeteSpaceyWalk: a heterogeneous spacey random walk for heterogeneous information network embedding. In: CIKM 2019, pp. 639–648 (2019)
7. Huang, Z., Cautis, B., Cheng, R., Zheng, Y., Mamoulis, N., Yan, J.: Entity-based query recommendation for long-tail queries. TKDD **12**(6), 64:1–64:24 (2018)
8. Jiang, H., Song, Y., Wang, C., Zhang, M., Sun, Y.: Semi-supervised learning over heterogeneous information networks by ensemble of meta-graph guided random walks. In: IJCAI 2017, pp. 1944–1950 (2017)
9. Lu, Y., Shi, C., Hu, L., Liu, Z.: Relation structure-aware heterogeneous information network embedding. In: AAAI 2019, pp. 4456–4463 (2019)
10. Ma, C., Zhang, B.: A new query recommendation method supporting exploratory search based on search goal shift graphs. IEEE Trans. Knowl. Data Eng. **30**(11), 2024–2036 (2018)
11. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS 2013, pp. 3111–3119 (2013)
12. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: NAACL 2013, pp. 746–751 (2013)
13. Shi, C., Hu, B., Zhao, W.X., Yu, P.S.: Heterogeneous information network embedding for recommendation. IEEE Trans. Knowl. Data Eng. **31**(2), 357–370 (2019)
14. Sun, Y., Han, J.: Mining heterogeneous information networks: a structural analysis approach. SIGKDD Explor. **14**(2), 20–28 (2012)
15. Sun, Y., Han, J.: Mining Heterogeneous Information Networks: Principles and Methodologies. Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool Publishers, San Rafael (2012)
16. Sun, Y., Yu, Y., Han, J.: Ranking-based clustering of heterogeneous information networks with star network schema. In: KDD 2009, pp. 797–806 (2009)
17. Szpektor, I., Gionis, A., Maarek, Y.: Improving recommendation for long-tail queries via templates. In: WWW 2011, pp. 47–56 (2011)
18. Wang, C.: Meta-path constrained random walk inference for large-scale heterogeneous information networks. CoRR abs/1912.00634 (2019)
19. Wang, H., Zhang, F., Hou, M., Xie, X., Guo, M., Liu, Q.: SHINE: signed heterogeneous information network embedding for sentiment link prediction. In: WSDM 2018, pp. 592–600 (2018)
20. Wang, X., Zhang, Y., Shi, C.: Hyperbolic heterogeneous information network embedding. In: AAAI 2019, pp. 5337–5344 (2019)
21. Xiang, E.W., Liu, N.N., Pan, S.J., Yang, Q.: Knowledge transfer among heterogeneous information networks. In: ICDM 2009, pp. 429–434 (2009)
22. Xu, J., Ye, F., Yu, H., Wang, B.: Query recommendation based on improved query flow graph*. In: IJCNN 2019, pp. 1–8 (2019)
23. Yang, L., Zheng, Y., Cai, X., Pan, S., Dai, T.: Query-oriented citation recommendation based on network correlation. J. Intell. Fuzzy Syst. **35**(4), 4621–4628 (2018)
24. Yuan, W., He, K., Han, G., Guan, D., Khattak, A.M.: User behavior prediction via heterogeneous information preserving network embedding. Future Gener. Comp. Syst. **92**, 52–58 (2019)
25. Zhu, Y., et al.: Query-based interactive recommendation by meta-path and adapted attention-GRU. In: CIKM 2019, pp. 2585–2593 (2019)

Data Mining Applications

Applications of Big Data in Tourism: A Survey

Malika Becha[1], Oumayma Riabi[1], Yasmine Benmessaoud[1](✉), and Hela Masri[1,2]

[1] Tunis Business School, University of Tunis, Ben Arous, Tunisia
malikabecha@outlook.fr, riabi.oumayma@gmail.com, yasmine.bnm5@gmail.com, masri_hela@yahoo.fr
[2] LARODEC, Tunis, Tunisia
https://www.tunis-business-school.tn/, http://www.larodec.com/

Abstract. Big data has become the focus of many researchers due to its important potential and ability to solve issues related to large-scale data. Tourism is one of the industries that are trying to use Big Data concept in optimizing their business processes. A comprehensive summary of previous studies related to big data implementation in tourism is introduced in this paper. At first, the different data types used and provided in this field are presented: UGC data generated from the part of the users, Device data, extracted from technological devices such as GPS, and Transactional data related to the operations made by the user from website visits, to online booking records. This paper also includes three different real-life applications of big data in Tourism. These applications present different implementations of this concept through recommendation tools based on the users' preferences, the demand forecasting tool, and the bibliometric analyses that study the current state of big data research. This survey provides a general idea about big data implementation in the tourism sector and introduces future improvements.

Keywords: Big data · Tourism · Demand forecasting · Recommendation system · Network analysis · Bibliometric analysis

1 Introduction

With the increasing technological advances and breakthroughs, a huge amount of data is being provided and produced in a speedy matter under multiple formats and data types. As a result, the concept of Big data came to life, and although no exact universal definition is available [7], most agree on its main characteristics, also known as the 3Vs: Volume which refers to the size of the available data, Velocity which is related to the speed of its creation and Variety which means the different types and categories of data available [16].

In fact, the concept of big data is becoming one of the main interests of businesses and researchers all over the world, and in every field, from healthcare,

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 533–546, 2020.
https://doi.org/10.1007/978-3-030-65390-3_40

to finance, and even tourism [10]. This evolving concept is now considered as the way to provide solutions to organizational issues in order to reach the optimal outcomes.

Tourism for example, is considered one of the fields where big data has big potential. It is considered as the answer to the challenges related to the "Smart Growth" of tourism including countries to visit and tourism-oriented companies [14]. Since the Tourism is an information-intense industry, it mainly relies on information technology [15]. In fact, the information derived and provided to the customers are the basic input - that is required by this industry in order to assess, evaluate, and analyze the preference, opinions, and trends related to modern tourism [9].

As a result, research work related to tourism have incorporated big data technique to improve the industry. For example, [29] have indicated that the use of big data and the high levels of information overturns the limitations related to the sample size or selection issues in order to better understand the trends and patterns related to the behavior of tourists. It also provides a deeper understanding of seasonal and annual demand, tourist satisfaction, the competition of touristic destinations, etc.

Thus, considering the potential of big data in changing the way businesses operate, research and studies related to its implementation in the field of tourism is very encouraged by governments and businesses in order to provide them with knowledge and tools to create *smart* destinations and personalize the businesses' goods and services.

In this paper, we will be categorizing the available big data used in this field into three main types: the User-Generated Content data, the device data, and the transactional data. The difference between the aforementioned types is mainly related to the source of the data. However, it is important to note that the level of accessibility also differs from one category to another, with the UGC data being the most accessible of the three.

Based on the different types, we will also be introducing different applications within the same industry in order to highlight the techniques and the processes used to implement big data in smart tourism.

2 Categories Big Data Used in Tourism

Many types of big data are used in tourism research with various resources and accessibility levels. However, all of these data are divided into three main categories [18]: first, one of the most accessible categories of data is the one provided by different users on online platforms from social media to evaluation forums. This is probably one of the most used data, not only because of its availability but also of its high volume. This category is called the user-generated (created) content, mainly known as UGC data. Second, we have the device data related to the data generated by automated devices. This category is characterized by its high-quality data collected thanks to software and complex networks that allow that storage and transmission of such data. Finally, the transaction data, a highly trusted tourism research data related to the different operations and transactional behaviors that are used very often in tourism prediction, website traffic, and marketing strategies.

2.1 UGC Data

With the technological advances and the heavy use of online platforms, people have become more and more active in sharing their opinions and providing feedback. This led to a change in the way businesses operate and enhance their goods and services. In fact, the information provided on the web and especially on social media websites, allow businesses to evaluate their performance and analyze the needs of their target customers at a very affordable price. This is why UGC data is one of the most used and sought types of data for businesses in most fields.

One of the fields that use this type of big data is tourism. It uses UGC data to promote tourism research. This field is heavily based on feedback and clients' interaction. This can be possible due to two types of data within this category: online textual data (reviews, ratings, and sharing tips and experiences) and online photo data.

Although user generated data can be under multiple forms, its source is mainly composed of social networks and forums. In fact, many online travel agencies, online booking websites and even tourism oriented social networks, allow customers to share their feedback, complaints, and recommendations. As a result, this type of data has become very valuable, and businesses like TripAdvisor [26] and Booking.com [27] are proving to be one of the pioneers in providing data related to customer reviews and comparisons.

In addition, data extracted from social networks such as Facebook, Instagram and Twitter have proven to provide information related to tourism that was later used to predict and analyze the geographic distribution of tourists and travelers [4]. Even the pictures shared and their frequency on these platforms allow tourism-oriented businesses to analyze and understand the customer behaviors through the location-based Social Network data in addition to its geographic and age distribution through location tagging [6].

Research Focus - Online Photo Data

UGC data is considered as the most accessible and affordable type of big data available in tourism. And since it is generated directly from the users based on their experiences, preferences, and expectations, it can provide businesses with an idea about the levels of satisfaction of their customers, with a possibility to analyze and determine the trends and attributes related to this parameter [24]. In addition, it is important to admit that tourists tend to rely on people's recommendations regarding tourism, including the itinerary, the hotel, the means of transportation and even the time of the visit [30].

Tourist behavior is one of the most important fields of interest for tourism-oriented businesses. By determining, analyzing and interpreting this indicator, hotels, airlines and even transportation companies would have an idea about the demand at a certain period of time.

Online photo data played a huge part in providing related information through the “Location Tagging” feature in photo-sharing websites, a technique that was used to determine geographic distribution of tourists by [6], and even to determine the most popular destinations by [17].

2.2 Device Data

To provide big data for tourism, the development of complex networks and devices like IoT and sensors has been employed to track the behavior and movements of tourists and the conditions of the environment. All these data sources are called device data (primary category of data) and they generated considerable spatial-temporal data for tourism-related big data [18]. Examples are data of GPS, mobile roaming, WIFI, meteorological data and Bluetooth. GPS is used for behavior tracking, mobile roaming is used for telecommunication, and WIFI – which is considered an alternative to Bluetooth is used for movement tracking [18].

Big data from device sources have structured and unstructured data. They are extracted mostly from:

GPS: Global Positioning System, which has the largest proportion of data compared to the other miscellaneous device data. By tracking travelers’ movements, this tool showed an obvious feasibility in tourism research [18].

Mobile Roaming Data: Mobile network operators provide this service to track tourists behavior by collecting data, stored in mobile log files, via radio waves. Unfortunately, this tool hasn’t been widely used in tourism research for privacy concerns [18].

Bluetooth: It’s a short-range, open and wireless communication technology [18]. Collecting data from Bluetooth is used by sensors placed in a target area. Cleaning this data is done through temporal and spatial filtering. Then, association rule learning is used for pattern mining [18].

WIFI: Wireless Fidelity, which is a wireless connection between your computer and the Internet connection [1]. It is an alternative to Bluetooth in tracing tourists’ behavior. WIFI data has a big role in tourism recommendation (e.g. events) and management of emergency [18].

Meteorological Data: Weather conditions recorded at ground-based stations may be considered the gold standard for meteorological data [5]. Data is collected by sensors of automatic weather stations and there are generally fifteen categories of data in different formats. However, the value for this type of data is underestimated because it has a great effect on tourism recommendations and weather estimations [18].

Research Focus - Geotagged Data

As [18] explained, mobile roaming is an effective tool that tracks tourists' movements outdoors. Bluetooth, on the other hand, is a cost-effective and convenient technique. It can also track tourists without their knowledge either in a crowded indoor place or near tall structures. WIFI tracking data has a bright prospect for tourism recommendation and management of emergencies. Moreover, using MapReduce algorithm and Hadoop architecture, meteorological big data provided recommendations to tourists by calculating tourism indexes. Among all of the aforementioned strong points in each data source device, applied GPS data showed a superiority in tourism research and by that it has three main stages. Stage I focused on the usefulness and feasibility of applying GPS data. Stage II was more concentrated on tourist behavior (spatial, temporal and spatial-temporal behaviors). Finally, stage III was about tourism recommendations (destinations, itineraries, etc.) that the collected GPS data provided.

2.3 Transaction Data

Transaction data has been a valuable asset that researchers have been using to enrich the literature of diverse fields of big data research. In this era of technology, where the Internet and smart devices have invaded our life allowing us to transact in an easy and quick way, thousands transactions are being recorded hourly, pulled into databases, and then processed to deliver knowledge and add value to our life. In our context, transactional records track all operations and transactions that took place in the tourism sector. Examples of operations may include, the purchase of food, airline tickets, and accommodations. This can be extracted from different sources.

Web search data is a type of transaction data that is extracted from search engines. The latter has become a source of big data as it serves to track the tourism-related information that users have been looking for. Moreover, other transaction data are owned by governmental agencies and private organizations in the field such as hotels and restaurants. Researchers have no access to this data since they are controlled by these private parties.

Research Focus - Web Search Data

Web search data have had several applications in the tourism sector. It is an efficient tool to study the behavior of customers and to generate useful insights that support tourism related to decision-making processes that researchers utilized to predict the demand [18]. Actually, this data highlights the terms that tourists have been browsing. Therefore, if well processed and analyzed, it reflects the expectations and the future plans of tourists. In most papers, web search data is viewed as an excellent predictor of tourists' arrivals (i.e., [8]).

Along those lines, in order to grab the attention of tourists across the globe, businesses like tourist attractions, hotels and restaurants promote their services via the Internet. The search engine is the major tool that links the client to the service. It is evident that tourist choices of destination or accommodation can be affected according to the search result pages generated by a search engine.

Previous studies highlighted that web search data can help companies in their online advertising campaigns through gaining more visibility on search result pages [3]. More specifically, through analyzing the most searched keywords, businesses learn about the tourism products customers are seeking and then they tailor their advertising content accordingly [12].

3 Big Data Application in Tourism

3.1 Big Data for Smart Tourism: A Recommendation Tool

Smart tourism has been an interest of many researchers in the past years as a mean to improve tourism and provide an efficient way to optimize tourism management through the use of technology and advanced management tools. Big data was heavily used in this context due the huge amount of data provided directly from the customers on social media and online forums.

In fact, user-generated content, is attracting more and more attention due to its easy accessibility and availability in huge volumes. For these reasons, many tools have been developed to improve tourism, from predicting tools targeting the tourist behavior [20] to recommendation generators for a personalized travel experience based on the user's preferences [21], travel schedule based on travel sequences and tourism locations [30], and the travel destination and the corresponding hot attractions and activities based on the geographical terms [28] in addition to the tourist web profile [25].

Process Analysis

Recommendation tools are based on 3 main functional steps:

Step 1: The collection of the data related to tourism which are available on online forums and social platforms to be stored with semantic web technologies [21]. Web crawling methods are also used to extract UGC tourist blogs from popular Tourism websites [30] and social media sites [28]. Large data objects are also extracted by [25] using the Jsoup java lib package for Hadoop applications.

Step 2: Extracting and analyzing information used to properly aim at potential tourists. This is done through data mining tools and methods. [25] uses the Pig Latin script of Hadoop to map essential datasets' attributes that loads (input), transforms, stores and dumps data (output). Data cleaning is also included in this phase in order to remove non-significant information such as misspelled words, and stop words ([30]; [28]). This is also used by [25] as a de-duplication technique to filter and save just the unique objects in the database by resolving duplication of copies. In addition to that, [21] uses a

system to help detect the data using a range of keywords including (age, gender, interests, etc.) which would be later utilized in the identification of the user type.
Step 3: Providing a personalized recommendation to the customer using the recommendation system based on their preference [21]. Such insightful knowledge was extremely helpful in improving tourism management and offering tourism recommendations ([28,30]. For [25], this recommendation tool recommends the most suitable location through choices and ratings that the user (tourist) provides after filling a tourist profile form.

Limitations and Future Work

Although [21] provided an advanced approach to smart tourism in order to improve the efficiency of touristic management in Morocco, a complete model was not developed, and so there is no possibility to evaluate its actual impact on this field. This paper introduces many different approaches, both old and recent for the recommendation tool. However, it is important to note that there was no comparison between the suggested approaches included in this study, which is needed to provide us with an evaluation of the efficiency of each method used. One way to improve this work, is by developing a recommendation tool to test its actual impact on tourism in developing countries like Morocco.

For [30] and [28], it is important to note that both papers have used samples with sizes that are considered relatively small. Unfortunately, this can lead to both selection and estimation biases. This is unfortunately, bound to lead to misleading outcomes. As a result, it is advised to use larger samples for future work and studies.

As for [25], the system's strong and important point is the de-duplication as it improves storage utilization as well as execution speed, thus improving recommendation's quality. Limitations for this tool are related to Hadoop. If data scraped from the web is small, it would not be suitable to use it for recommendation. Security concerns are also one of Hadoop limitations as it does not implement cryptography at storage and network levels [2]. This can risk cyber spying on information.

3.2 Big Data in Tourism Demand Forecasting

Tourism is considered a complex industry where businesses are striving to meet the fluctuating and seasonal demand. Under such circumstances, being able to anticipate accurate future trends is important to stay one step ahead of the competition. Among the various applications of big data is demand forecasting. Much effort has been invested to determine how big data affects forecasting accuracy and to compare big data based predictions to traditional ones.

Process Analysis

One traditional approach researchers use to generate demand predictions is the autoregressive approach. In this technique, future demand is forecasted based solely on past demand. A group of researchers, namely, [11], attempted to predict the monthly number of tourists' arrivals to the Swedish mountain, following an autoregressive approach that uses KNN and linear regression techniques. Then, in order to capture value added by big data to the performance of those techniques, big data attributes were introduced as part of the inputs. The authors highlighted the fact that lagged web search data is interesting and meaningful in this context since tourists usually look for information regarding their holiday plans in advance. Their results showed that both KNN and linear regression had a lower MAE and RMSE when the demand-lagged variables were combined with big data features. In addition, the empirical results of the study found that big data attribute features had a stronger influence on the tourists' arrivals than the past demand variables. These observations led to the conclusion that big data based techniques lead to more accurate predictions of tourists' arrivals compared to autoregressive techniques.

Another traditional way to predict the demand is the use of time series approaches. In their paper, [13] compared the forecasts delivered from time series approaches to an ARIMAX forecast which was trained on additional big data attributes. They collected web search data from the Chinese Baidu search engine, and extracted UGC data from two well-known review platforms (Qunar and Ctrip) to produce 12-week ahead point forecasts of the tourist arrivals to Mount Siguniang which is a national park in China. Comparing the error measures, they came to the conclusion that ARIMAX trained with big data attributes outperformed the following traditional time series techniques: Seasonal ARIMA, ARIMA, and Exponential Smoothing.

[13] also compared single-source big data based models to multi-source big data based models. They constructed several ARIMAX models having different combinations of big data predictors in the aim of evaluating the performance accuracy when introducing multiple big data types and features to the model instead of one unique feature or type. The research indicated that the model that had web search and UGC data from two different platforms had a better performance than the one with web search data only. In addition, the model that had data from two review platforms outperformed the two models that were trained using a single platform data each.

Limitations and Future Work

The aforementioned papers drew similar conclusions. Big data based models are more accurate than the autoregressive and time series approaches. Also, big data increases the performance of predictive techniques. However, [13] used tourism related Internet big data, lagged demand variable, and neglected important exogenous factors. In fact, political stability, cleanliness, attractions, health issues can exemplarily have a significant impact on future demand. Moreover, the analysis conducted by both groups of researchers were based on a case-study and cannot be generalized.

3.3 Big Data in Tourism: Research Analysis

Development of complex networks and devices like IoT and mobile devices play a big role in delivering considerable big data. Conducting researches in tourism & hospitality fields for a better use of these data has been of a great effect for them. As big data-related research in general became popular, some researchers applied bibliometric analysis to methodologies used that improved decision making in tourism field. While other researchers preferred to know how tourism research evolved in recent years, and how the tourism-big data research can be drawn from other relevant disciplines; knowing that tourism studies are the intersections of various disciplines (sociology, economics, business, culture and management) [23].

To apply bibliometric analysis, [22] needed to gather publications between 1994 and 2015. They highlighted the innovation of data techniques in tourism field. Authors focused on the analysis of "the usage that research on tourism has made of new methodologies" concentrating on big data BD, data mining DM and structural equations models SEM.

For other researchers, [19], they emphasized how appropriate solutions and algorithms should be applied to understand consumer satisfaction, detect patterns and get knowledge for tourism recommendations. This research was done by analyzing big data-related published papers and journals from 2008 to 2017 in both tourism and relevant fields. To obtain comprehensive outputs of the progressing research trends and state, a modelling & visualization tool was used. CiteSpace was the used tool that helped shaping the most cited articles of a certain topic in a certain time period. Also, a systematic network analysis was conducted on the covered publications.

Process Analysis

To follow progress of big data-related research, two major steps were used: data collection and research analysis.

Step 1 - Data collection: [22] selected random specific articles from SCOPUS as a database for analysis. Specific keywords were used to screen articles process (big data, data mining, SEM). [19] collected 2 datasets from academic journals' articles, conference editorials, etc. from Web of Science. Keywords and retrieval search engines were selected as follows: Dataset 1 was collected using the keyword 'big data' alongside 'tourism', 'travel', 'hotel' and 'hospitality' to analyze tourism field only. Dataset 2 covers relevant publications and only 'big data' keyword was used.

Step 2 - Research Analysis: after coding and analyzing important variables, [22] performed quantitative analysis using R which then confirmed the increasing importance of research in tourism and hospitality fields. [19] considered performing **preliminary analysis** (Data attributes were examined through basic descriptive analysis to indicate a publication's research impact), then network analysis that assesses co-citation, cluster and trend metrics. Finally, a **comparative analysis** to acquire a general assessment (similarities and differences) of research topics in both network analysis in tourism research and its connection to relevant domains.

Limitations and Future Work

[22] believe that more efforts into bibliometric analysis should be performed. Future work included identifying the most relevant subjects using BD, DM and SEM techniques as well as collaborating with tourism institutions. [19]'s study's limitations consisted of not covering all bigdata-related research in the used dataset. Covered publications and references were selected through keywords "big data" and "social media" with tourism-related terms. Data was retrieved from one source: Web of Science. Authors future work consists of including additional relevant keywords and get data from different providers (e.g. Google Scholar). It also include applying machine learning methods to filter publications for a more efficient and accurate database.

4 Conclusion

Since its introduction to our world, big data has been a game-changing opportunity that revolutionized the way we grow and make decisions. Businesses have been investing to gain knowledge from such a timely, high frequency, and voluminous data and to make in place optimized processes.

Big Data Benefits

Big data impacted different fields, the one relevant to this study is tourism. This sector has been gaining the interest of several researchers. It is considered a valuable asset that can add values to both research and academia, since it covers a broader scope of knowledge than traditional data.

In tourism literature, several papers were published highlighting the benefits of the different types of big data when applied to address real world applications. The choice of the type depends on the application and can be grouped into three main categories [18].

User-generated content is one type of tourism big data that is published by the users themselves on review platforms and online forums. Device data is another type, thanks to the introduction of Internet of Things, our smart devices are being used to collect this data. A third type of big data is the transaction data, which relates to any transaction or operation that takes place in the tourism market. Transaction data includes web search data that can be retrieved from internet search engines.

One of the main focuses of big data is the optimization of recommendation systems that can only function accurately when sufficient data is provided. These recommendation systems are able to analyse the interest of the individual and build a set of suggestions and recommendations that would match the information collected automatically. This will not only provide businesses with an analysis tool that can build recommendations, but also an excellent tool to segment the interests and provide a personalised user experience.

In addition, Internet big data in demand forecasting has been a topic of debate and interest in tourism field of research. Multiple papers confirmed that big data improves the performance of predictive models, and delivers better predictions than do traditional forecasting approaches, since it covers up-to-date information regarding the expectation of the potential tourists.

Drawbacks of Existing Works

Amongst previous research, some papers studied recommendation systems that are trained on big data. In fact, the authors [21,25,28,30] used user-generated data that reflects the expectations and the needs of tourists.

These papers however, had a number of pitfalls. [21], focused only on the theoretical aspect of the problem. The absence of the implementation made the model performance ambiguous. In addition, the paper mentioned several filtering techniques that the model can use, however, no information was provided regarding the performance of each technique, or which technique outperforms the other. [30] and [28] used small size samples that cannot be generalized and it could even lead to misleading results due to the estimation and selection biases. As a result, the use of large-scale samples is highly recommended for more efficient outcomes. This is also the case of [25], in which Hadoop raises concerns related to the risk of the low volume of scrapped data. In addition, this paper also is prone to security issues related to the lack of cryptography implementation for storage and network.

[11] applied big data to forecast the monthly number of tourists' arrivals to the Swedish mountain. They found out that an ARIMAX model with additional lagged big data features outperforms an auto-regressive model that uses past demand only. Also, [13] compared the accuracy of an ARIMAX model trained on big data attributes to Seasonal ARIMA, ARIMA, and Exponential Smoothing to predict the weekly demand of a Chinese national park and concluded that a model that is based on big data features can outperform traditional time series approaches. Moreover, [13]'s analysis revealed that a model that used multi-source big features yields better forecasts than a model trained on a single-source big data. This can be explained by the fact that big data is big and diverse, so it can incorporate the different side and aspects of a problem.

We believe that the aforementioned papers that dealt with the forecasting issues present some limitations. First, both analyses built use-case based models, the results of which cannot be generalized to other applications. Second, [13] predictive model overlooked important exogenous factors that can enhance the forecasts. Future studies can incorporate, for instance, the political stability, the health issues, and the cleanliness of the destination.

Other papers include the research analyses of the publications that are related to bigdata in tourism in a specific time period. Authors [22] highlighted the role of innovation that big data research in tourism has. They focused on techniques of big data, data mining, and structural equations modelling. Their goal was to confirm the importance of the previously mentioned techniques in tourism field when conducting a research.

For [19], they emphasized giving more importance to using big data in the tourism field as it improves the functions of many aspects in the tourism and hospitality fields such as tourism recommendations. Their paper offered multiple implications for practice. It performed network analysis and applied a comparative analysis. First, it extracted publications from Web of Science and divided them into two sets of data using search engine (multiple keywords "big data", "tourism", and any related word for dataset #1 and only "big data" for dataset #2). These datasets were then used to extract connections between publications through network analysis (co-citation, clustering and trend analysis). Lastly, a comparative analysis using only the second dataset took place to highlight similarities and differences in network analysis and its connection to relevant domains.

Not covering all big data related publications was a major limitation in this study. Future work includes using more words in search engine and reducing manual work through applying machine learning techniques for filtration purposes.

References

1. Al-Alawi, A.I.: WiFi technology: future market challenges and opportunities. J. Comput. Sci. **2**(1), 13–18 (2006)
2. Barolli, L., et al.: Advances in Internet, Data & Web Technologies. The 6th International Conference on Emerging Internet, Data & Web Technologies (EIDWT-2018). Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-319-75928-9
3. Pan, B., Xiang, Z., Law, R.: Daniel R Fesenmaier. J. Travel Res. **50**(4), 365–377 (2011)
4. Cheng, M., Edwards, D.: Social media in tourism: a visual analytic approach. Curr. Issues Tour. **18**(11), 1080–1087 (2015)
5. Colston, J.M., et al.: Evaluating meteorological data from weather stations, and from satellites and global models for a multi-site epidemiological study. Environ. Res. **165**, 91–109 (2018)
6. Da Rugna, J., Chareyron, G., Branchet, B.: Tourist behavior analysis through geotagged photography: a method to identify the country of origin. In: 2012 IEEE 13th international symposium on Computational Intelligence and Informatics (CINTI), Budapest (2012)

7. Gandomi, A., Haider, M.: Beyond the hype: big data concepts, methods, and analytics. Int. J. Inf. Manage. **35**(2), 137–144 (2015)
8. Gawlik, E., Kabaria, H., Kaur, S.: Predicting tourism trends with google insights (2011)
9. Hall, C.M., Williams, A.: Tourism and Innovation (2008)
10. Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A., Khan, S.U.: The rise of "big data" on cloud computing: review and open research issues. Inf. Syst. **47**, 98–115 (2015)
11. Höpken, W., Ernesti, D., Fuchs, M., Kronenberg, K., Lexhagen, M.: Big data as input for predicting tourist arrivals. In: Schegg, R., Stangl, B. (eds.) Information and Communication Technologies in Tourism 2017, pp. 187–199. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-51168-9_14
12. Huang, X., Zhang, L., Ding, Y.: The baidu index: uses in predicting tourism flows CA case study of the Forbidden City. Tour. Manag. **58**, 301–306 (2017)
13. Li, H., Hu, M., Li, G.: Forecasting tourism demand with multisource big data. Ann. Tour. Res. **83**, 102912 (2020)
14. Jackson, S.: Prediction, explanation and big(ger) data: a middle way to measuring and modelling the perceived success of a volunteer tourism sustainability campaign based on nudging. Curr. Issues Tour. **19**, 643–658 (2016)
15. Koo, C., Gretzel, U., Hunter, W.C., Chung, N.: The role of IT in tourism. Asia Pac. J. Inf. Syst. **25**(1), 99–104 (2015)
16. Laney, D.: 3D data management: controlling data volume, velocity and variety. META Group Res. Note **6**, 70 (2001)
17. Lee, I., Cai, G., Lee, K.: Exploration of geo-tagged photos through data mining approaches. Expert Syst. Appl. **41**(2), 397–405 (2014)
18. Li, J., Xu, L., Tang, L., Wang, S., Li, L.: Big data in tourism research: a literature review (2020)
19. Li, X., Law, R.: Network analysis of big data research in tourism. Tour. Manage. Perspect. **33**, 100608 (2020)
20. Miah, S.J., Vu, H.Q., Gammack, J., McGrath, M.: A big data analytics method for tourist behaviour analysis. Inf. Manage. **54**(6), 771–785 (2017)
21. Boulaalam, O., et al.: Proposal of a big data system based on the recommendation and profiling techniques for an intelligent management of moroccan tourism. Proc. Comput. Sci. **134**, 346–351 (2018)
22. Palomoa, J., Figueroa-Domecqa, C., Flecha-Barrioa, M.D., Segovia-Pérez, M.: The use of new data analysis techniques in tourism: a bibliometric analysis in data mining, big data and structural equations models (2017)
23. Ritchie, J.R., Sheehan, L.R., Timur, S.: Tourism sciences or tourism studies? Implications for the design and content of tourism programming. Téoros. Rev. Rec. Tour. **27**(27–1), 33–41 (2008)
24. Liu, Y., Teichert, T., Rossi, M., Li, H., Hu, F.: Big data for big insights: investigating language-specific drivers of hotel satisfaction with 412,784 user generated reviews. Tour. Manage. **59**, 554–563 (2017)
25. Thasal, R., et al.: Information retrieval and de-duplication for tourism recommender system. Int. Res. J. Eng. Technol. (IRJET) **05**(03) (2018)
26. Xie, K.L., Zhang, Z., Zhang, Z.: The business value of online consumer reviews and management response to hotel performance. Int. J. Hospit. Manage. **43**, 1–12 (2014)
27. Xu, X., Li, Y.: The antecedents of customer satisfaction and dissatisfaction toward various types of hotels: a text mining approach. Int. J. Hospit. Manage. **55**, 57–69 (2016)

28. Xu, H., Yuan, H., Ma, B., Qian, Y.: Where to go and what to play: towards summarizing popular information from massive tourism blogs. J. Inf. Sci. **41**(6), 830–854 (2015)
29. Yang, X., Pan, B., Evans, J.A., Lv, B.: Forecasting Chinese tourist volume with search engine data. Tour. Manage. **46**(C), 386–397 (2015)
30. Yuan, H., Xu, H., Qian, Y., Li, Y.: Make your travel smarter: summarizing urban tourism information from massive blog data. Int. J. Inf. Manage. **36**(6), 1306–1319 (2016)

High-Quality Plane Wave Compounding Using Deep Learning for Hand-Held Ultrasound Devices

Baozhu Guo[1,2](✉), Bin Zhang[1,2], Zhuang Ma[1,2], Ning Li[1,2], Yiping Bao[1,2], and Dan Yu[1,2]

[1] Dalian Neusoft University of Information, Dalian, Liaoning, China
{jt_guobaozhu,zhangbin_jt,jt_mazhuang,jt_lining,baoyiping, yudan}@neusoft.edu.cn

[2] Dalian Neusoft Educaiton Technology Group Co. Limited, Dalian, Liaoning, China

Abstract. Ultrasound imaging has been widely used in clinical diagnosis because of the characteristics of non-radiation, cost effective and real-time interaction. Hand-held ultrasound device has shown great potential in community health care and telemedicine. However, the image quality and frame rate are compromised due to simplification of hand-held hardware which severely limit its applications. The coherence plane-wave compounding imaging (CPWC) technology is usually used to generate high-quality ultrasound images, but compounding dozens of plane waves (PW) with different incident angles will dramatically reduce the frame rate. To get high-quality image while maintaining the frame rate for hand-held ultrasound imaging, in this paper we propose a LG_Unet model, which combines local detail information and global structure information through residual learning. Additionally, the natural image-based transfer learning with cascade training is adopted to alleviate the learning failure and overfitting issues. The experimental results demonstrated that our model can provide high-quality ultrasound images through only 3 PWs, in which the contrast and lateral resolution of the reconstructed images are better than those of the CPWC results from 31 PWs. The proposed method shows the feasibility of generating high-quality images for hand-held ultrasound device using plane wave while maintaining a promising balance between the image quality and frame rate.

Keywords: Ultrasound imaging · Coherence plane wave compounding · LG_Unet

1 Introduction

Comparing with other clinical diagnostic imaging modalities, ultrasound imaging has the advantages of non-invasive, safe, convenient and real-time. It is widely used in examining of fetus, heart and abdomen. At present, ultrasound imaging has become one of the most important imaging modes in clinical diagnosis.

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 547–559, 2020.
https://doi.org/10.1007/978-3-030-65390-3_41

Hand-held ultrasound is an ultrasound device with the shape close to a mobile phone. Comparing with the conventional cart-based ultrasound instruments, hand-held ultrasound has many unique advantages: (1) it is widely used and can be flexibly adapted to many special scenarios such as home care, bedside, emergency settings etc. [1–3]. (2) It is much cost-effective than traditional large-scale ultrasound instruments. (3) It has attracted great attention from research institutes and companies because of the potential market. Additionally, in developing countries where people can't afford expensive high-end ultrasound instrument, the hand-held ultrasound device can provide cost-effective point-of-care. So it is very significant for global healthcare [4, 5].

Due to the limitation of the device size, the imaging capability of the hand-held ultrasound is greatly restricted comparing to the traditional cart-based large-scale equipment. In ultrasound imaging, the delay-and-sum (DAS) method is widely adopted where the number of transmissions is proportional to the number of lines of each image because the image is generated by repeating the line-by-line transmitting and receiving procedure in the whole interested regions. So the frame rate is limited, usually about 20 to 60 frames per second (fps), not to mention the hand-held device. In order to improve the frame rate, plane wave imaging (PWI) is proposed where a plane wave (PW) is transmitted to insonify the entire region while a whole image is obtained in parallel by beamforming the acquired data. This can greatly increase the frame rate to as high as 4,000 to 15,000 fps [6]. The PWI method paved the way for a series of new clinical applications, such as ultrafast Doppler imaging, shear wave elastography, brain functional imaging, etc. [7–9].

Although high frame rate is achieved, PWI suffered from a decline in image quality (low spatial resolution and contrast) because of the lack of focusing in the plane wave transmission. Therefore, improving the quality of plane wave ultrasound images has become a research hotspot in this field. Coherence compounding is a popular research direction in this field. In this method, a series of tilted PWs are transmitted, and then the images obtained from each transmission are coherently added together to yield a final compounded image [10]. Although coherence plane wave compounding (CPWC) improves the image quality, the image is not uniformly distributed along the depth due to the different insonified regions during different transmissions. With respect to the increased tilted angle, the PWs are not concentrated in deeper area where the image brightness is not uniform with the shallower area. In addition, as the number of tilted PWs increasing, the effective frame rate decreases. There have to be a trade-off between frame rate and image quality. Therefore, the research of high-performance plane wave imaging technology is particularly important in order to achieve high-quality imaging without sacrificing frame rate.

With the continuous progress in deep learning area, it has achieved good results in various fields such as computer vision and natural language processing. Many deep learning algorithms have been introduced as well to solve the problems of medical image processing, such as medical image segmentation, tumor detection, etc. [11, 12]. These algorithms have indicated prominent improvements when comparing with the traditional methods. Recently, there are researches tried to combine deep learning with the reconstruction of high-quality ultrasound image. Zhang *et al.* generated high-quality ultrasound images by using only 3PWs images through the Generative Adversarial Network (GAN) [13]. Wang *et al.* effectively improved the image quality of ultrasound images obtained from hand-held ultrasound devices through a modified version of U-net [14]. Zhou *et al.* made full use of the plane wave images obtained from different angles and designed a multi-channel multi-scale two-level convolutional neural network model, which perfectly combined local information with context information and obtained more realistic reconstruction results [15]. Their studies show that the deep learning algorithm can achieve a balance between frame rate and image quality, and demonstrated the feasibility of wide applications of hand-held ultrasound equipment based on PWI.

Although the image quality of hand-held ultrasound can be improved through end-to-end deep learning method, there are still some challenges as follows. (1) Most deep learning algorithms are data-driven, and massive data is needed to train the model in order to take the advantage. A small amount of data will directly lead to failure of model training or overfitting of training results. Currently, the resources of ultrasound data are generally divided into three aspects: simulated data obtained from simulation, phantom data acquired from ultrasound platform, and *in vivo* data collected from a pioneer or a patient. However, the number of public ultrasound raw datasets is relatively small, which is far from sufficient for model training. (2) During the propagation of ultrasonic waves in human body, the signal will be attenuated. This leads to a non-uniform brightness distribution along the imaging depth, and the deeper the depth, the darker of the image. It increases difficulty for training deep learning network and affects the quality of training results.

In this paper, we propose a deep learning network, namely LG_Unet, to improve the image quality of hand-held ultrasound device with respect to the abovementioned two problems. The proposed model is obtained through cascade training and can directly reconstruct high-quality ultrasound images from 3 PWs images. The reconstructed image quality is close to that of 31 PWs. The proposed model are of the following two advanced characteristics:

(1) We use natural image dataset to pre-train the network so that it can learn information of contrast and texture from natural images. Next, we use ultrasound images to perform a process of transfer learning on the pre-trained network. This cascade training method solves the problem of insufficient number of ultrasound images.
(2) In order to learn local detail information of ultrasound images separately, we combine another network, S_Resnet, which uses the patch dataset obtained from cropping the ultrasound training dataset. Then we add the reconstructed image to the global features extracted from the LG_Unet model, and finally get the reconstructed image. This method solves the problem of the non-uniform brightness distribution of ultrasound images.

Our experiments show that the proposed model has significantly improved the image quality of hand-held ultrasound device.

The rest of this paper is organized as follows. Section 2 describes our proposed method and purpose. Section 3 includes the introduction of our experimental data, training steps and evaluation indicators. Section 4 shows our experimental results including the ultrasound image reconstructed by different methods and the comparison data of various evaluation indicators. In Sect. 5, we summarize the entire work and point out some deficiencies in our algorithm.

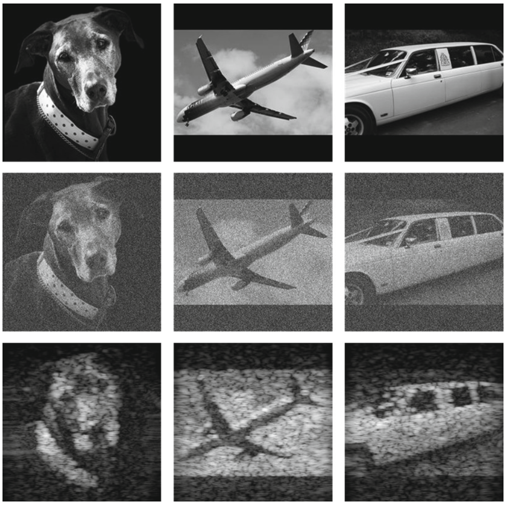


Fig. 1. The pre-training data. The 1st row are natural images, the 2nd row are label images, and the 3rd row are simulated ultrasound images (generated by the ultrasound simulation software, Field II, from natural images).

2 Method

2.1 Data-Based Transfer Learning

In order to alleviate the problem of insufficient training datasets in deep learning-based methods for ultrasound image enchantment, we consider the data-based transfer learning strategy. Firstly, we select natural images with rich texture features, high contrast and high resolution to pre-train the network. After that, we use ultrasound images to carry out transfer learning on the network. Since the number of ultrasound images is limited and the resolution is low, the ultrasound-image-trained network, is prone to overfitting or non-convergence. In contrary, natural images can provide the network with wide dynamic features and more natural contrast for learning. Yet, the number of natural images is sufficient, and these aforementioned problems did not occur when we train the network.

We extract 2,000 natural images from the public COCO dataset. In order to reduce the difference between the ultrasound image and the natural image, and ensure the validity of the transfer learning, we use the ultrasound simulation software, Field II [16], to generate simulated ultrasound images from natural images as input to the pre-training network [17]. During this procedure, 5 dB Gaussian noise has been added to the original natural image as the label data when pre-training the network. Figure 1 shows the ultrasound simulation data, label data and natural images used for network training.

2.2 Network Architecture

In CPWC imaging, multiple low-quality images (usually 31 or 75) are compounded to generated one high-quality image. Here, we are trying to generate a high-quality image by only 3 low-quality images with the advantage of deep learning network.

First, we use a simplified version of the residual network to learn the local information from high-quality images. The ultrasound image is inherently inclined to be non-uniform as mentioned before, especially in PW imaging mode. This non-uniformity would negatively affect the quality of reconstructed ultrasound image, cause the increased difficulty for network learning. To solve this problem, we cut the ultrasound image into small pieces and design a simplified version of the residual network called S_Resnet. The S_Resnet focuses on learning the local detail information between low-quality images and high-quality images. The S_Resnet network structure is shown in Fig. 2.

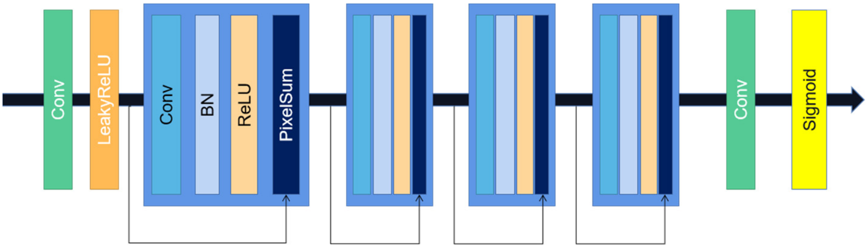


Fig. 2. The framework of the S_Resnet model

The S_Resnet is composed of an input convolution layer, an input activation layer, four residual blocks, an output convolution layer, and an output activation layer. All convolutional layers in the network used a 3 × 3 convolution kernel with a stride size of 1, and the image size does not change during the execution of each layer. The inputs of the network are 3-channel low-quality ultrasound image patches cropped from three low-quality ultrasonic images of different angles in the same area. The output of the network is a single-channel ultrasound image patch. The target high-quality ultrasound image patches are cropped from the high-quality ultrasound image obtained by 31PWs at the same position. The value of the loss function is calculated according to the network output and the target high-quality ultrasound image patch. The trained S_Resnet network model can directly reconstruct the local ultrasound information of the entire ultrasound image.

Second, we adopt the Unet network to learn the global structural information from high-quality images for ultrasound image enhancement.

By comparing the single-angle low-quality ultrasound image with the high-quality ultrasound image obtained from 31 PWs, it can be found that the difference between the local detail content was notable, as well as the global structural information.

In the original Unet model [18], the model uses a standard encoding-decoding structure. The network gradually extracts high-level features through the down-sampling layer, and then restores the image to its original size through the up-sampling layer. After multiple down-sampling operations, the network can remove part of the noise in the image, while effectively retaining the structural information in the image. But at the same time, a lot of important details that are closely related to the quality of the final reconstructed image will be lost, so that the final generated image becomes blurred.

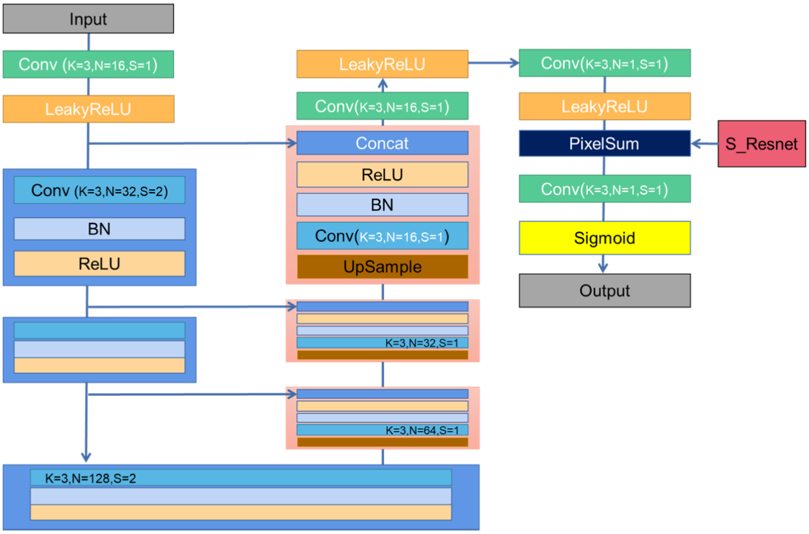


Fig. 3. The framework of the proposed LG_Unet model.

We propose a modified Unet model, namely LG_Unet (Local Global Unet), to improve the ultrasound image enhancement. In LG_Unet, we combine the local detail information of the image with the global structure information. The network architecture of LG_Unet is shown in Fig. 3.

The LG_Unet consists of three down-sampling blocks and three up-sampling blocks. Down-sampling is achieved by setting the stride of the convolution kernel to 2, and up-sampling is achieved by bilinear interpolation. After the last up-sampling block, residual learning is introduced. That is, an ultrasound image containing complete local detail information is added before the last convolution layer. This image is obtained by reconstructing local detail information from the input image of LG_Unet through the S_Resnet network. This image is directly added to the feature map output from the penultimate convolutional layer, and then input to the last convolutional layer to obtain the final reconstructed image.

Since the ultrasound datasets for model training are not sufficient, so we first use the simulated dataset obtained from the natural image as described in Sect. 2.1 to pre-train LG_Unet. However, for the training of S_Resnet, due to the patch dataset obtained by sliding cutting on the original ultrasound dataset, the acquisition method is simple and the number is sufficient, so no pre-training is required.

Table 1. The training procedure

Input Output	3 low-quality ultrasound images a high-quality ultrasound image
Step 1	Input 3 low-quality ultrasound images into LG_Unet and obtain the feature map output from the penultimate convolutional layer
Step 2	Input 3 low-quality ultrasound images mentioned in Step 1 into the S_Resnet model trained with patch images to obtain the partial reconstructed image
Step 3	Add the feature map in Step 1 and the partial reconstructed image in Step 2. Put the result into the last convolutional layer of LG_Unet to obtain the final reconstructed image
Step 4	Use the final reconstructed image and the high-quality ultrasound image from 31 PWs to calculate the loss function. Update the weight of the LG_Unet by gradient descent method
Step 5	Repeat Steps 1–4 until the loss function is small enough and then save the LG_Unet model
Step 6	Input 3 low-quality ultrasound images that are not involved in training into the LG_Unet model to obtain a high-quality ultrasound image

2.3 Loss Function

To generate a high-quality image from low-quality images using only 3 PWs, the network needs to learn not only the global structural features but also the local detail features. For these two different purposes, we use different loss functions for network training. For S_Resnet, we use *L1* loss function to maintain pixel-level consistency between low-quality and high-quality images, which is shown in Eq. (1).

$$L_{L1} = \frac{1}{N}\sum_{\mathrm{p}\in P} |y(p) - r(p)|, \tag{1}$$

where N represents the total pixel number of the ultrasound image, r represents the network reconstructed image, y represents the target high-quality image, and p represents the pixels on the image.

In LG_Unet training, we use a combined loss function consisting of Mean Squared Error (MSE) loss and Structural Similarity Index Measure (SSIM) loss, which reflect pixel-based information and structure-based information, respectively. This can maintain the main structure during the training process while taking into account the fusion of local features and global features, so as to obtain more information and improve the performance of the network. The loss function is defined as shown in Eqs. (2)–(4).

$$L_{MSE} = \frac{1}{N}\sum_{\mathrm{p}\in P} (y(p) - r(p))^2, \tag{2}$$

$$L_{SSIM} = 1 - \frac{1}{N}\sum_{p\in P} (\frac{2\mu_r\mu_y + C_1}{\mu_r^2 + \mu_y^2 + C_1} \cdot \frac{2\delta_r\delta_y + C_2}{\delta_r^2 + \delta_y^2 + C_2}), \tag{3}$$

$$L_{sum} = L_{MSE} + L_{SSIM}, \tag{4}$$

where N represents the total number of pixels of the ultrasound image, r represents the network reconstructed image, y represents the target high-quality image, p represents the pixels on the image, μ indicates the mean calculation, δ indicates the variance calculation, and C_1, C_2 are two constants.

The main training procedures mentioned above are shown in Table 1.

3 Experiment

3.1 Training Datasets

The ultrasound dataset we use includes three parts: simulation data, phantom data and *in vivo* data. The image size is 450 × 200. The 3 tilted PWs at −15°, +15° and 0° are selected as the three channels of the network input image. The 31 PWs between +15° and −15° with 1° increment are obtained and compounded by the traditional CPWC method. The compounded image from 31 PWs is regarded as the high-quality image and the label data for network training. The phantom data is from the dataset of 2016 IEEE IUS Plane-wave Imaging Challenge in Medical Ultrasound [19]. The *in vivo* data is kindly provided by Dr. Luo from Tsinghua University with the specification detailed in Ref [13].

The patch dataset used for S_Resnet training is obtained by sliding clipping of the above ultrasound dataset. The sliding window is a square with the side length of 128 pixels. The upper left corner of the original ultrasound image is set as the coordinate origin, and the slide is gradually moved from the upper left to the lower right with a stride size of 8 pixels. Finally, 9,225 patch images are obtained, of which 9,000 are used for training and 225 are used for testing.

3.2 Neural Network Training

In the training process of LG_Unet and S_Resnet, we chose the Adam operator to perform gradient descent calculation and to minimize the loss function, with the learning rate of 0.0001. The training was performed on the NVIDIA Tesla V100 GPU. The training of the S_Resnet model took 500 iterations. The LG_Unet model first trained 500 iterations using the ultrasound simulation dataset and another 200 iterations using the ultrasound dataset.

3.3 Image Reconstruction Quality Metrics

In order to evaluate the quality of the reconstructed image and the performance of the trained network model, we calculated the peak signal-to-noise ratio (PSNR), the full width at half maximum (FWHM) of a wire target at 49.1 mm depth, the contrast ratio (CR) and contrast-to-noise ratio (CNR) on the phantom image and the *in vivo* image.

PSNR measures the similarity between two images. The calculation formula is shown in Eq. (5). We chose the compounded image from 31 PWs as the target image. The larger the PSNR value is, the closer it is to the target image.

$$PSNR = 10\log(\frac{255^2}{MSE}), \tag{5}$$

where *MSE* represents the mean square error of two images.

CR and CNR illustrate the contrast of gray levels between different areas in an image. The calculation formula is shown in Eqs. (6)–(7).

$$CR = \frac{|\mu_{R1} - \mu_{R2}|}{\mu_{R1} + \mu_{R2}}, \tag{6}$$

$$CNR = 10\log[\frac{(\mu_{R1} - \mu_{R2})}{\sqrt{\delta_{R1}^2 + \delta_{R2}^2}}], \tag{7}$$

Where R_1, R_2 represent two areas of the same size on the image, μ represents average value of these areas, and δ represents variance of these areas.

4 Result

4.1 Phantom Data Results

Figure 4 and Fig. 5 show the reconstructed results of phantom data, including the image compounded using 3 PWs, the image compounded using 31 PWs, and the reconstructed image obtained through the proposed LG_Unet network, respectively. From these results, we can see that the image generated by our network is better than the one generated from 3 PWs, and is similar to that of the 31 PWs.

The quantitative comparison of different results is shown in Table 2. The FWHM refers to the lateral FWHM of a point target (as shown in Fig. 5(a)). The FWHM value reveals the lateral resolution of the image, and our model shows good lateral resolution, even superior than that of the 31 PWs image.

The CR and CNR between the cystic region and the background (red and green circles in Fig. 4(a)) are also compared, as shown in Table 2. It is noticeable that the reconstruction results obtained by the proposed model have the highest contrast among the three methods.

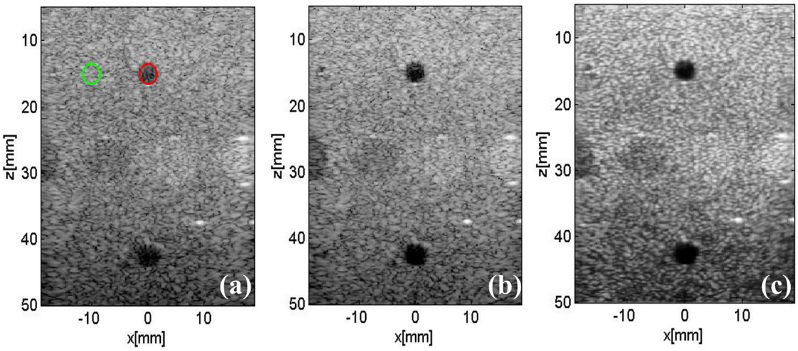


Fig. 4. Results of phantom data. (a) Compounded image using 3 PWs. (b) Compounded image using 31 PWs. (c) Generated image using the proposed LG_Unet. The green and red circles are background and cystic regions respectively for CR and CNR calculation. Each image displays 60 dB of dynamic range. (Color figure online)

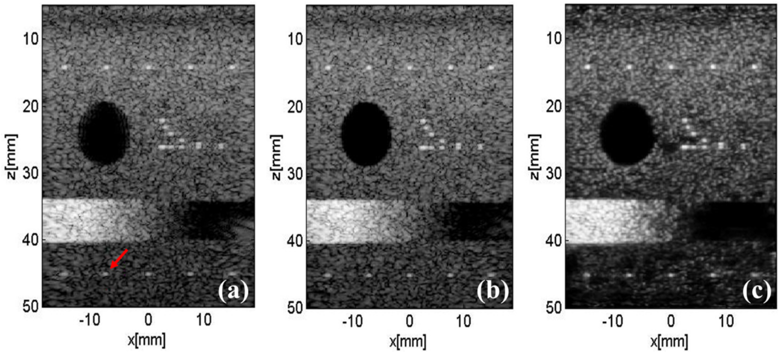


Fig. 5. Result of phantom data. (a) Compounded image using 3 PWs. (b) Compounded image using 31 PWs. (c) Generated image using the proposed LG_Unet. The red arrow indicates the point target for FWHM calculation. Each image displays 60 dB of dynamic range. (Color figure online)

Table 2. Quantitative evaluation of phantom results

	3 PWs	31 PWs	LG_Unet
FWHM (mm)	1.8	1.7	1.3
CR (dB)	0.36	0.68	0.79
CNR (dB)	24.93	30.38	32.81

4.2 *In vivo* Data Results

We further assessed the performance of the proposed model on *in vivo* data. Figure 6 shows the reconstruction results. Through the experimental results, we can find that the image generated by our network is better than the 3 PWs image, and is similar to the 31 PWs image but much smoother.

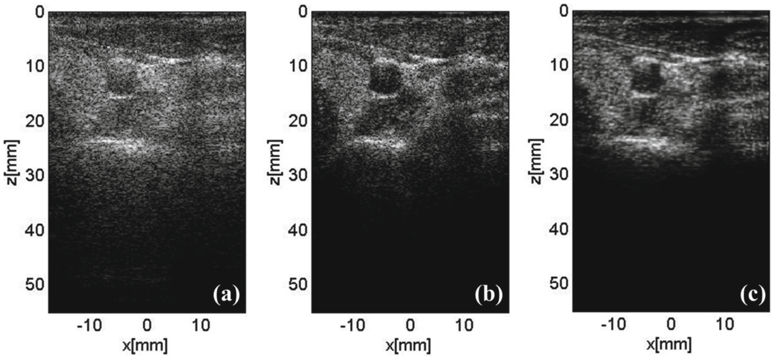


Fig. 6. Result of *in vivo* data. (a) Compounded image using 3 PWs. (b) Compounded image using 31 PWs. (c) Generated image using the proposed LG_Unet. Each image displays 60 dB of dynamic range.

To quantitatively evaluate the *in vivo* results, the PSNR value is calculated as shown in Table 3. In this experiment, we use the image from 31 PWs as the target image. The PSNR values are obtained for the image from 3 PWs and the image are reconstructed by the proposed LG_Unet network. It can be seen that the image obtained by our network is much higher than that obtained from 3 PWs and is closer to the result compounded from 31 PWs.

Table 3. Quantitative evaluation results of *in vivo* images

	3PWI	LG_Unet
PSNR	16.04	18.94

5 Conclusion and Discussion

In this paper we proposed an LG_Unet model to improve the image quality for handheld ultrasound device using plane wave. To deal with the issues of insufficient training ultrasound datasets and non-uniform distribution of the ultrasound image, the transfer learning method based on natural images is combined with cascade training in the proposed model. This method can effectively solve the problem of model learning failure and overfitting. By learning the texture features and wide dynamic contrast from natural images, and combining local details with global structures via residual learning, the quality of the reconstructed images obtained by our network has been significantly improved.

Although our method has achieved good results in multiple sets of experiments, there are still some limitations needed to be improved in the future:

(1) The number of *in vivo* ultrasound images used to verify the generalization of the model is very small and mostly concentrated on the thyroid, which cannot fully verify the performance of the proposed model. In the future, it is necessary to add more *in vivo* data from other organs to completely evaluate the performance of the proposed model.

(2) Here we only add the residual structure on the basis of Unet and cooperate with transfer learning to test the feasibility of low-quality ultrasound image improvement. There are many variants of Unet in the field of image segmentation, which can be combined to further improve the image quality. Additionally, the effectiveness of the specific models needs further evaluation and optimization in the hand-held ultrasound device.

References

1. McBeth, P.B., Hamilton, T., Kirkpatrick, A.W.: Cost-effective remote iPhone-teathered telementored trauma telesonography. J. Trauma **69**(6), 1597–1599 (2010)
2. Evangelista, A., Galuppo, V., Mendez, J.: Hand-held cardiac ultrasound screening performed by family doctors with remote expert support interpretation. Heart **102**(5), 376–382 (2016)
3. Bornemann, P., Bornemann, G.: Military family physicians' perceptions of a pocket point-of-care ultrasound device in clinical practice. Mil. Med. **179**(12), 1474–1477 (2014)
4. Stock, K.F., Klein, B., Steubl, D.: Comparison of a pocket-size ultrasound device with a premium ultrasound machine: diagnostic value and time required in bedside ultrasound examination. Abdom. Imaging **40**(7), 2861–2866 (2015). https://doi.org/10.1007/s00261-015-0406-z
5. Becker, D.M., Tafoya, C.A., Becker, S.L.: The use of portable ultrasound devices in low- and middle-income countries: a systematic review of the literature. Trop. Med. Int. Health **21**(3), 294–311 (2016)
6. Montaldo, G., Tanter, M., Bercoff, J.: Coherent plane-wave compounding for very high frame rate ultrasonography and transient elastography. IEEE Trans. Ultrason. Ferroelectr. Freq. Control **56**(3), 489–506 (2009)
7. Tanter, M., Fink, M.: Ultrafast imaging in biomedical ultrasound. IEEE Trans. Ultrason. Ferroelectr. Freq. Control **61**(1), 102–119 (2014)
8. Urban, A., Dussaux, C., Martel, G.: Real-time imaging of brain activity in freely moving rats using functional ultrasound. Nat. Methods **12**(9), 873–878 (2015)
9. Eby, S.F., Song, P., Chen, S.: Validation of shear wave elastography in skeletal muscle. J. Biomech. **46**(14), 2381–2387 (2013)
10. Denarie, B., Tangen, T.A., Ekroll, I.K.: Coherent plane wave compounding for very high frame rate ultrasonography of rapidly moving targets. IEEE Trans. Med. Imaging **32**(7), 1265–1276 (2013)

11. Lai, M.: Deep learning for medical image segmentation (2015)
12. Shkolyar, E., Jia, X., Chang, T.C.: Augmented bladder tumor detection using deep learning. Eur. Urol. **76**(6), 714–718 (2019)
13. Zhang, X., Li J., He, Q.: High-quality reconstruction of plane-wave imaging using generative adversarial network. In: IEEE International Ultrasonics Symposium (IUS), pp. 1–4 (2018)
14. Wang, R., Fang, Z., Gu, J.: High-resolution image reconstruction for portable ultrasound imaging devices. EURASIP J. Adv. Signal Process. **2019**(1), 1–12 (2019). https://doi.org/10.1186/s13634-019-0649-x
15. Zhou, Z., Wang, Y., Yu, J.: High spatial-temporal resolution reconstruction of plane-wave ultrasound images with a multichannel multiscale convolutional neural network. IEEE Trans. Ultrason. Ferroelectr. Freq. Control **65**(11), 1983–1996 (2018)
16. Jensen, J.A.: A multi-threaded version of Field II. In: 2014 IEEE International Ultrasonics Symposium, pp. 2229–2232 (2014)
17. Hyun, D., Brickson, L.L., Looby, K.T.: Beamforming and speckle reduction using neural networks. IEEE Trans. Ultrason. Ferroelectr. Freq. Control **66**(5), 898–910 (2019)
18. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
19. Liebgott, H., Molares, A.R., Jensen, J.A.: Plane-wave imaging challenge in medical ultrasound. In: Proceedings of 2016 IEEE International Ultrasonics Symposium (2016)

IPMM: Cancer Subtype Clustering Model Based on Multiomics Data and Pathway and Motif Information

Xinpeng Guo[1,2], Yanli Lu[2], Zhilei Yin[1], and Xuequn Shang[1](✉)

[1] School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an 710072, People's Republic of China
shang@nwpu.edu.cn

[2] School of Air and Missile Defense, Air Force Engineering University, Xi'an 710051, People's Republic of China

Abstract. Multiomics compiles data from different genome levels to study the effects of interactions between various omics molecules on disease processes. Integrated analysis of different omics data can more comprehensively evaluate their role in human health and complex diseases. Previous studies have used SNF and SNF-CC for multiomics integration. Although the effect of multiomics integrative algorithm is significantly increased, these methods did not consider the effects of a biologically significant correlation within and between omics. A large body of evidence has shown that cancer occurs due to interactions and synergistic effects of multiple genes. The correlation relationships between genes can be reflected through gene pathway and motif information. In this paper, we define the IPMM(Integration Pathway and Motif information Model), which combines pathway and motif information with multiomics data to study their effects on cancer subtype classification. To facilitate the use of gene association information, we employ the Isomap method for dimensionality reduction analysis of expression data from the genomes in a pathway and motif. Selection of K values in Isomap dimensionality reduction is used to maximize the presentation of the relationship of genes in pathway and motif data with dimensionality reduced to one. SNF and SNF-CC are used for integrative analysis of gene-expression data, methylation data, miRNA data, and pathway and motif data after dimensionality reduction in two cancer datasets. Results show that clustering effects display varying increases in different methods after pathway and motif information are integrated.

Keywords: Multiomics · Pathways · Motifs · Subtype classification · Data integration

1 Introduction

The continuous development of high-throughput technologies and decline in sequencing costs has made it easier to collect various types of genomics data that can support the study of interactions between multiple omics. These effects in various omics layers

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 560–568, 2020.
https://doi.org/10.1007/978-3-030-65390-3_42

during life processes can only be captured by the integrated study of multiple molecular layers [1].

Methods to integrate various omics data, such as SNF and SNF-CC, use different perspectives to analyze their relationship to cancer [2, 3], and have been continuously improved, with a focus on algorithm optimization [4, 5]. However, these methods have not considered biologically significant relationships within and between omics data. For example, the relationship between gene interaction and cancer formation and development [6], the influence of genes containing the same sequence on the generation and classification of cancer [7], whether the generation of cancer is related to the position of genes in chromosomes [8], and whether the omics interaction is related to the formation of cancer [9, 10], Cell function is achieved through mutual coordination of genes or synergistic effects of similar genes. This promotes a series of associations between genes that produce diverse behaviors from cellular metabolism to signal transduction [11]. These genes are often present in the same pathway or have similar gene sequences. Considering the above, IPMM was used to fuse omics data with pathway data sets representing gene pathways and motif data information representing similar gene sequences for analysis. Results showed varying increases in clustering effects in different methods after pathway and motif information were integrated.

2 Algorithm Introduction

We describe some basic methods used in this paper. Isometric mapping (isomapping) is a nonlinear dimensionality-reduction algorithm. We also discuss similarity network fusion (SNF) and SNF-consensus clustering (SNF-CC) for multiomics data integration.

2.1 Isomap Algorithm

There are two common methods of dimensionality reduction. One is dimensionality reduction on the coordinate relationship to provide points, such as principal component analysis (PCA), which maps n-dimensional features to k dimensions, which are new orthogonal features, or principal components. PCA is often used to reduce the number of dimensions in a dataset while retaining features with the highest contribution to its variance. Another method is to carry out dimensionality reduction on the distance matrix of points, such as multidimensional scaling (MDS) [12], which requires that distances between all samples in the resultant low-dimensional space are equal (or close) to those before dimensionality reduction and retains the relative relationships of the original data. PCA and MDS are linear dimensionality-reduction methods with limited scopes of application. The Isomap algorithm [13] is derived from MDS, and has become a widely used nonlinear dimensionality-reduction method [11, 14]. Isomap differs from MDS mainly in the calculation of the distance matrix of the original space. MDS uses the simplest and most direct method, the Euclidean distance. Isomap retains intrinsic geometric structures in nonlinear data through the geodesic distance, which is unchanged after dimensionality reduction.

Many papers have compared Isomap to PCA in dimensionality reduction of gene-expression data [11, 14]. Results have shown that Isomap is better at visualization and clustering analysis for complex gene-expression data, and a lower-dimensional embedded space can be used to represent the original data. Hence Isomap can better reflect the true relationships between genes. Therefore, we use Isomap for dimensionality reduction of relationships between genes.

2.2 SNF Algorithm and SNF -CC Algorithm

The SNF algorithm (Wang et al.) constructs sample similarity networks for various omics data, and their complementarity calculation is used to integrate patient similarity networks. The patient similarity network is expressed as a graph G = (V, E). The vertex V corresponds to n samples $\{x_1, x_2, \cdots, x_n\}$; the weight of edge E is represented by an n × n similarity matrix W. The matrix W is standardized to obtain P, with solution $P = D^{-1}W$, where D is a diagonal matrix whose entries $D(i, i) = \sum_j W(i, j)$. However, this type of standardization may cause a self-similarity problem on the diagonal, resulting in unstable values. Self-similar scales on the diagonal can be removed through the equation

$$P(i,j) = \begin{cases} \frac{W(i,j)}{2\sum\limits_{k\neq i} W(i,k)}, & j \neq i \\ 1/2, & j = i \end{cases} \tag{1}$$

with the result that $\sum_j P(i, j) = 1$.

The matrix P represents all information on the similarity between samples. To increase calculation efficiency when iterating on other omics data, the SNF algorithm employs the k-nearest neighbors (KNN) algorithm, which defines the similarity of the top N_i samples that are closely associated with sample x_i, while similarity with the remaining samples is defined as 0. The similarity is defined as follows:

$$S(i,j) = \begin{cases} \frac{W(i,j)}{\sum\limits_{k\in N_i} W(i,k)}, & j \in N_i \\ 0, & j \notin N_i \end{cases} \tag{2}$$

During integration of various omics information, SNF iteratively calculates the matrices P and S to capture the local structure of the graph and increase calculation efficiency. the similarity matrix for various omics data can be obtained as

$$P^{(v)} = S^{(v)} \times \left(\frac{\sum_{k\neq v} P^{(K)}}{m - 1} \right) \times \left(S^{(v)} \right)^T, \ v = 1, 2, \cdots, m \tag{3}$$

The final sample similarity matrix P can be used for classification and clustering. This paper focuses on cancer subtype clustering and prediction.

SNF-CC is the joint execution of SNF and consensus clustering (CC) [15, 16] in cancer subtype classification. SNF obtains the similarity matrix of patients after integration, and this is used as a consensus input for clustering. CC determines the possible number of clusters in a dataset and provides quantitative evidence for members. CC aims to evaluate cluster stability, and resampling-based methods can be used to validate cluster rationality and optimization of the original SNF algorithm.

3 Execution Process

IPMM is adopted to integrate the prior knowledge into the multi-omics fusion method, which does not consider the associations in the intra-omics data. The primary consideration is the association relationship between genes, where pathways represent gene–gene interactions and motif represents gene structural relationships. As pathway and motif data are processed similarly, pathway processing data are used as an example to illustrate data processing.

Assume a gene is associated with its pathway. Hence changes in its expression level will affect the function of the pathway. Based on this assumption, we extracted the expression values of various pathway-related genes from all gene-expression data. Every pathway forms a relationship matrix between one sample and genes associated with that pathway. As it is difficult to use multi-dimensional gene-expression data features in subsequent clustering algorithms, a suitable dimensionality-reduction method is required for processing. We found Isomap to be better at visualization and clustering analysis than other dimensionality-reduction methods for complex gene-expression data, and reduction to one dimension can maximally show all the pathway information. Information from p pathways was used for dimensionality reduction, after which the first dimensional vectors of all pathways were combined to form a pathway expression matrix (n samples * p pathways). The motif data were processed similarly to obtain n samples * q motif matrices. Our method still adopts the previous data integration method (SNF and SNF-CC) for cancer subtype clustering analysis. Figure 1 shows the algorithm framework.

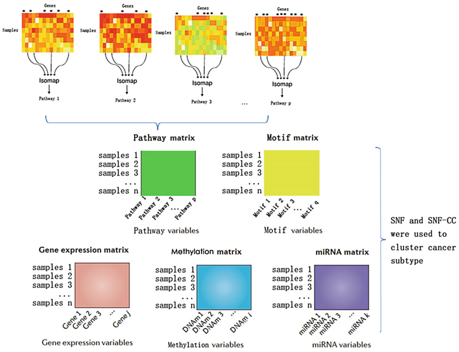


Fig. 1. IPMM model framework with integrated pathway and motif information.

Compared to SNF, which does not consider the association relationships of omics data, in our model we added the two inter-gene association relationships of functional and structural association. To directly use the previous multiomics integration methods(SNF and SNF-CC), we reduced the dimensionality of these two association relationships to convert them to matrix formats like those of other types of omics data. Two types of content should be considered in dimensionality reduction. First, a type of dimensionality reduction that is suitable for the gene relationship matrix should be considered to ensure that the low-dimensional space maximally retains original information. Second, parameter selection should retain maximal original information when dimensionality is reduced to one.

We mentioned above that the nonlinear Isomap dimensionality-reduction method can better reflect the true association relationship between genes. Therefore, we employ Isomap for dimensionality reduction of relationships between genes. The optimal nearest neighbor K value and manifold and the optimal number of embedded dimensions should be considered when using Isomap, as they determine the final result of dimensionality reduction. If K is too low, then continuous manifolds will be erroneously separated into disjointed submanifolds and mapping will not reflect global characteristics. If K is too high, then the entire dataset will become a local neighborhood. A greater parameter d and a smaller loss-cost function can better maintain the intrinsic geometric structures in the data. However, any dimensionality-reduction method will result in information loss. Only by minimizing information loss and compressing data to a sufficiently small dimension can dimensionality reduction be achieved. The aim of dimensionality reduction is to minimize information loss while compressing data to one dimension. Therefore, the parameter d is fixed, and we consider a K value such that data information is concentrated in one dimension as much as possible for optimal dimensionality reduction. To this end, we reference the "residual method" (Tenenbaum et al.) [17], in which the "elbow" inflection point of the residual curve, i.e. the point where the residual curve stops decreasing as the number of dimensions increases, is identified to estimate the optimal number of embedded parameters. We use the eigenvalue of the Isomap algorithm to calculate that point. The eigenvalue of Isomap is similar to that in PCA dimensionality reduction, and it reflects the weight of values in this dimension. We obtained eigenvalues of various groups using different K values. The eigenvalues of a K value were ranked ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$). For the inflection point to appear between the first and second eigenvalues, we can solve the equation $E = (\lambda_1 - \lambda_2)/(\lambda_2 - \lambda_3)$ for different K values. The maximum E value shows that original data information can be maximally retained by decreasing to one dimension, at which point K has the optimal k-nearest neighbor value. Wilk et al. [11] used this concept and validated the method's feasibility.

4 Data and Results

4.1 Source of Data

We downloaded these data for breast and lung cancer and removed eigenvalues for low expression levels and low variation in the samples. Table 1 shows the data.

Pathway and motif information were obtained from the C2 and C3 gene datasets in the Molecular Signatures Database v7.1 (updated March 2020). C2 includes subsets of the classical pathway and biologically significant signaling pathways. C2 was generated from nine databases and includes 5529 genes. C3 includes gene sets such as miRNA target genes and transcription factor binding sites.

4.2 Introduction to Evaluation Markers

In terms of the evaluation of the method, we still use the method of SNF and SNF-CC for verification, and comparative validation using the same criteria can demonstrate the

Table 1. Number of patients and number of genes in mRNA expression, DNA methylation, and miRNA expression datasets.

Cancer Type	mRNA expression	DNA methylation	miRNA expression	Number of patients
BRCA	17814	23094	354	105
LUNG	12042	23074	352	106

superiority of IPMM. SNF validation was carried out from the perspectives of silhouette [18] and P-value [19].

The cluster silhouette s(i) = (b(i)–a(i))/(max a(i), b(i)), where a(i) refers to non-similar mean values of patients of the same subtype, and b(i) to lowest non-similar mean values of all patients in different subtypes. The mean of all patients s(i) measures the silhouette of all data in the set. If s(i) is close to 1, then the data should belong to the same type.

The P-value validates the logrank test for survival separation in the Cox regression model. The threshold value was set to < 0.05.

Silhouette shows whether there is interference between subtypes. Because survival can intuitively reflect clustering results, we focus more on the P-value when the silhouette reaches a certain level.

4.3 Experimental Results

Gene association relationships were integrated with multiomics data, and SNF and SNF-CC were used for comparison to show the effects of association relationships in multi-omics on multiomics integration algorithms. We also attempted to integrate a single type of association relationship for comparison, such as pathway data information alone or motif data information alone. The validation results did not show significant improvement, and the effects on the overall results were not significant. Therefore, we only compared results of various diseases without integration with gene association relationships and when both pathway and motif relationships were integrated.

The results are compared and found that after gene association relationships were integrated, the survival curves for various clusters were more separated. This shows that IPMM can improve the overall clustering results. From Fig. 2, it can be seen that the differences in survival curves were more substantial and P-values were smaller after pathway and motif data were integrated into breast cancer subtype classification and SNF and SNF-CC were used, showing that classification accuracy was higher. The silhouette values were also significantly increased, showing that cluster delineation for every sample was clearer. From Fig. 3, it can be seen that the differences in survival curves were more significant and P-values were smaller after pathway and motif data were integrated into lung cancer subtype classification and SNF and SNF-CC were used, and accuracy was higher. Although the silhouette value is decreased in the one of Fig. 3, resulting in some interference in the clustering heat map, this does not affect the overall clustering result, and the survival curve was significantly improved, showing that the overall clustering effect was more accurate. From Table 2, it can be seen that SNF-CC is better than SNF in many aspects, which is consistent with previous conclusions. In addition, SNF-CC can more intuitively show improvements in performance when gene association relationships are integrated.

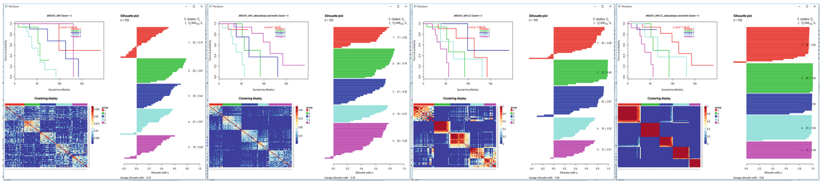


Fig. 2. Integrated pathway and motif information. SNF and SNF-CC were used for comparison with subtype classification differences in breast cancer data.

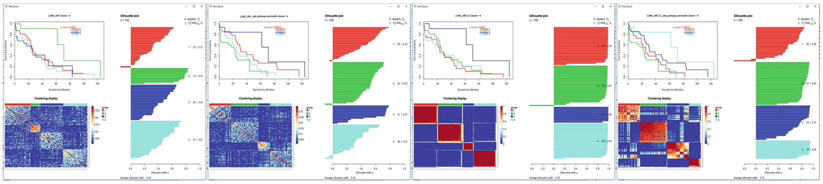


Fig. 3. Integrated pathway and motif information. SNF and SNF-CC were used for comparison with subtype classification differences in lung cancer data.

Table 2. Comparison of two diseases with and without gene integration association relationships.

		SNF		SNF-CC	
		P-value	Silhouette	P-value	Silhouette
BRCA	Non-integrated gene association information	4.14×10^{-5}	0.34	3.28×10^{-5}	0.64
	Integrated gene association information	1.5×10^{-5}	0.58	1.5×10^{-5}	0.94
LUNG	Non-integrated gene association information	0.0351	0.44	0.0351	0.93
	Integrated gene association information	0.00514	0.44	0.00514	0.74

5 Discussion

Although the clustering effect was increased after integrating pathway and motif information, there is room for improvement in the following areas.

(1) After integrating the gene correlation information, the clustering effect is improved to some extent. However, when the gene correlation information is reduced to one dimension, although the problem can be simplified and processed more conveniently, certain gene correlation information will obviously be lost.

(2) We only considered the effects of inter-gene association relationships on subtype classification. We considered neither other intra-omics association relationships nor those between omics, such as between genomics and epigenomics or epigenomics and transcriptomics. How to integrate this information will be a focus of our future studies.

References

1. Yugi, K., Kubota, H., Hatano, A., Kuroda, S.: Trans-omics: how to reconstruct biochemical networks across multiple'omic' layers. Trends Biotechnol. **34**, 276–290 (2016)
2. Lin, E., Lane, H.Y.: Machine learning and systems genomics approaches for multi-omics data. Biomark. Res. **5**, 2 (2017)
3. Ritchie, M.D., Holzinger, E.R., Li, R., Pendergrass, S.A., Kim, D.: Methods of integrating data to uncover genotype-phenotype interactions. Nat. Rev. Genet. **16**, 85–97 (2015)
4. Guo, Y., Liu, S.: BCDForest: a boosting cascade deep forest model towards the classification of cancer subtypes based on gene expression data. BMC Bioinf. **19**, 118 (2018)
5. Hasin, Y., Seldin, M.: Multi-omics approaches to disease. Genome Biol. **18**, 1–5 (2017)
6. Torshizi, A.D., Petzold, L.R.: Graph-based semi-supervised learning with genomic data integration using condition-responsive genes applied to phenotype classification. J. Am. Med. Inform. Assoc. **25**, 99–108 (2018)
7. Zhao, J., Cheng, F., Jia, P., Cox, N., Denny, J.C., Zhao, Z.: An integrative functional genomics framework for effective identification of novel regulatory variants in genome-phenome studies. Genome Med. **10**, 7 (2018)
8. Romanowska, J.: From genotype to phenotype: through chromatin. Genes **10**(2), 76 (2019)
9. Chu, S.H., Huang, Y.T.: Integrated genomic analysis of biological gene sets with applications in lung cancer prognosis. BMC Bioinf. **18**, 336 (2017)
10. Yuan, L., Huang, D.S.: A network-guided association mapping approach from DNA methylation to disease. Sci. Rep. **9**, 5601 (2019)
11. Wilk, G., Braun, R.: Integrative analysis reveals disrupted pathways regulated by microRNAs in cancer. Nucleic Acids Res. **46**, 1089–1101 (2018)
12. Jung, K.: Multidimensional Scaling I. In: Wright, J.D. (ed.) International Encyclopedia of the Social & Behavioral Sciences, 2nd edn, pp. 34–39. Elsevier, Oxford (2015)
13. Tenenbaum, J.B.: A global geometric framework for nonlinear dimensionality reduction. Science **290**, 2319–2323 (2000)
14. Shi, J., Luo, Z.: Nonlinear dimensionality reduction of gene expression data for visualization and clustering analysis of cancer tissue samples. Comput. Biol. Med. **40**(8), 723–732 https://doi.org/10.1016/j.compbiomed.2010.06.007
15. Sebastiani, P.: Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. Mach. Learn. **52**(1–2), 91–118 (2003)
16. Wilkerson, M.D., Hayes, D.N.: ConsensusClusterPlus: a class discovery tool with confidence assessments and item tracking. Bioinformatics **26**, 1572–1573 (2010)
17. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**, 2319–2323 (2000)

18. Silhouettes, R.P.J.: A graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. **20**, 53–65 (1987)
19. Hosmer Jr, D.W., Lemeshow, S.: Applied survival analysis: regression modeling of time to event data. J. Am. Stat. Assoc. (2000)

Personal Health Index Based on Residential Health Examination

Guanjun Lai[1,2](✉), Dan Yu[1,2], Shuai Zhang[1,2], Zelin Wei[1,2], and Xiaoyu Sun[1,2]

[1] Dalian Neusoft University of Information, Dalian, China
{laiguanjun,yudan,zhangshuai,weizelin,
jt_sunxiaoyu}@neusoft.edu.cn

[2] Dalian Neusoft Education Technology Group Co. Limited, Dalian, China

Abstract. Due to the problems in health examination records such as irregular examinations, missing values in examination indicators, and various time span, it is difficult to give a holistic view on personal or population health situations. In this paper, we propose a tensor decomposition method to deal with the missing data problem in health examination records by using a big data set collected over seven years from a medium size Chinese city. According to TOPSIS (Technique for Order Preference by Similarity to an Ideal Solution) and entropy weighting method, the weights of indicators reflecting the health status are calculated to establish a health index for residential health examination. In the forms of grade diagram and fingerprint diagram, the changes of the individual health index as well as corresponding ranking position and development trend are visualized. In our experiments, we demonstrate the useful-ness and the effectiveness of our approach.

Keywords: Personal health index · Big data of health examination · TOPSIS · Tensor decomposition · Fingerprint diagram · Health management

1 Introduction

As an important way to manage population health, regular individual health examinations have been widely concerned. In 2018, 575 million people took part in health examinations in China and accounted for 41.01% of the total population [1]. The display, interpretation, recording and preservation of health examination data are important for understanding the health status of individuals. Therefore, the utilization of health data to timely understand the population health of a country [2, 3] has become a new effort. However, many problems with respect to the interpretation and application of health examination data still remain to be solved.

We identify the following important analytical aspects of health records.

- **Interpretability**. Most people do not have sufficient medical knowledge to interpret health examination reports that are usually lengthy and involve professional indicators.

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 569–581, 2020.
https://doi.org/10.1007/978-3-030-65390-3_43

- **Examination Standard.** The types and ranges of health indicators, collected from different health examination institutions, show significant differences. It is difficult to uniformly summarize the overall health status for a person or for a population.
- **Baseline Comparison of Indicators.** Only the current values of indicators might be available, the comparative analysis of the baseline situation of the same age group would be needed [4].
- **Trend of Indicators.** It is difficult to grasp the changes of the individual health status over a long period. In most of the reports, only the current health examination results are available, the analysis of health changes for individuals in a long period would be helpful.
- **Discontinuity of Examination**. Most people carry out health examinations casually without in a fixed institution. The identity of examinees among different institutions are not national-wide unified in China. Therefore, it is difficult to associate health examinations of the same people from different institutions, leading to discontinuity of health examination data [2, 4].

We need to deal with the following challenges in order to solve the above problems.

- **Holistic View of Health Data.** It is necessary to design a comprehensive index to reflect individuals' health status. There are totally more than 400 examination indicators in the health examination process. It is difficult to evaluate individual health status based on a small subset of those indicators.
- **Missing Data.** The knowledge on how to deal with missing data in examination records is required. People may take different examinations in different institutes as well as in different time periods [2, 4]. Therefore, there are many missing data items in people's examination data records. Data can be missed in many different reasons, such as the accidents in data collection processes, or indicators that are considered quite normal, therefore not been included in the examination, or the values that could be inferred by using other variables, so were not recorded.
- **Benchmark Evaluation**. It is about how to deal with the examination data without labeling. The health examination records normally do not contain labels given by specialists or doctors and there is no ground truth for differentiating examinee's state of health [5].

Aiming at the above problems and challenges, in this paper we design a comprehensive health examination health index based on a few factors including multiple critical examination indicators, examination data distribution, and medical knowledge. We use tensor decomposition to recover the missing data on the health examination data of 7 years (i.e., between May, 2012 to November, 2019). We adopt TOPSIS (Technique for Order Preference by Similarity to an Ideal Solution) and entropy weighting method [6] to differentiate and rank the individual health. To this end, the health status is comprehensively explored and visually displayed by means of fingerprint diagrams. The analysis results indicate that the designed health examination health index could evaluate the health of individuals effectively and the designed diagrams could reflect individual health status and development trends effectively.

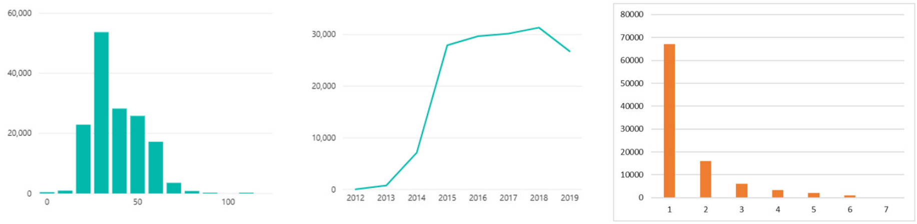
2 Related Work

In the last decade, increasing number of data mining applications has been developed to support health risk prediction and assessment. Ling Chen *et al.* designed a method called MyPHI to predict personal health index on geriatric medical examination records and Cause of Death data in 2016 [7]. However, there are no Cause of Death data linked with usual examination data as a ground truth in common examination institutes. So it is constrained by the availability of crucial data. Fu Qi *et al.* conducted analysis of health index on the method of mathematical statistics based on a small amount of data in 2015 [4]. However, they did not deal with the missing data which is important in data preprocessing. Mao Xiao *et al.* used TOPSIS to assess individual health on health indicators [8]. However, the authors did not provide a scientific method on weight evaluation of various indicators. The selection of weights involves many subjective decisions and is not completely based on the data itself. In addition, in the weight selection, the distribution of health examination data and medical knowledge are of importance if considered.

3 Health Examination Data

The data in this study has a total of 153,744 de-identified health examination reports from a health examination center between May 2012 to November 2019. The items in health examination reports included basic information of examinees, health examination items, abnormal items, conclusions and follow-up examination recommendations. The total data volume is 10.6 Gb.

The examinees included 50,505 males (52.78%) and 45,185 females (47.22%). The age distribution of health examinees is shown in Fig. 1(a). The proportion of people aged 30 to 40 was the highest (34.93%). The distribution of the number of health examinees in each year is shown in Fig. 1(b). It can be seen that the number of health examinations has increased year by year in recent years (the data recorded until November 11, 2019). The number of health examinations of an examinee at the health examination center is shown in Fig. 1(c). Among them, 67,072 people participated one health examination,



(a) (b) (c)

Fig. 1. Health examination data summary. (a) Age distribution of health examinees; (b) Distribution of annual health examinees; (c) Distribution of the number of health examinations of an examinee

accounting for 70.1%, and the maximum number of health examinations of an examinee was 7, including 122 people (0.001%).

4 Technical Routes and Methods

The establishment and visualization of the residential health examination health index mainly involve five steps, as shown in Fig. 2.

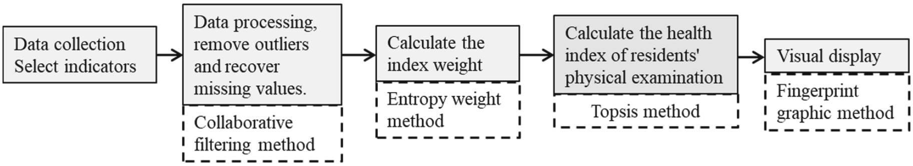

Data collection
Select indicators
Data processing, remove outliers and recover missing values.
Collaborative filtering method
Calculate the index weight
Entropy weight method
Calculate the health index of residents' physical examination
Topsis method
Visual display
Fingerprint graphic method

Fig. 2. Technical route of establishment of the health index.

4.1 Data Collection and Select Indicators

According to the purpose of health assessment, the relevant health examination indicators are selected. The indicator system data set $\left(X_{ij}\right)_{m\times n}$ is established where n is the number of health examination indicators; m is the number of health examinations; x_{ij} represents the *j-th* indicator of the *i-th* health examination.

4.2 Data Preprocessing

Data preprocessing may involve removing outliers and inserting missing data for learning purposes. The deletion of outliers is based on the degree of deviation from averages. Average value and standard deviation of the health examination indicator X_i are respectively defined as μ_{i} and σ_{i}, then we get

$$\mu_{\mathrm{i}} = \frac{1}{m}\sum_{j=1}^{m} X_{ij} \tag{1}$$

$$\sigma_{\mathrm{i}} = \frac{1}{m}\sum_{j=1}^{m} (X_{ij} - \mu_{\mathrm{i}})^2 \tag{2}$$

The abnormal data items were determined according to the 6σ method [9]. When the value of X_i is less than $\mu - 6\sigma$ or greater than $\mu + 6\sigma$, the value is an outlier and will be deleted to obtain the dataset *XA*.

As for the omitted examination items and outliers, the recovery of missing data is performed with the tensor decomposition [10] as follows:

First, the obtained dataset *XA* is subjected to matrix decomposition as below

$$XA \approx X\hat{A} = UV^T \tag{3}$$

where $X\hat{A}$ is the approximate matrix of *XA*; U is the $m \times k$ matrix; V is the $n \times k$ matrix; m and n are the same as above; k is a number less than m and n.

Then error J between *XA* and $X\hat{A}$ is

$$J = \left\| XA - X\hat{A} \right\|^2 = \left\| XA - UV^T \right\|^2 = \sum_{i,j,x_{ij} \neq nan} \left(x_{ij} - \sum_{l=1}^{k} u_{il} v_{jl} \right)^2 \quad (4)$$

where x_{ij}, u_{il} and v_{jl} are respectively the elements in the matrices *XA*, U and V. u_{il} and v_{jl} are the elements to be solved. The solution is realized by establishing the optimized target with gradient descent method. Finally, the data of $X\hat{A}$ corresponding to the missing data in *XA* are added to complete *XA* and the matrix *XB* is obtained.

4.3 Weighted Indicators

The weight of an indicator is determined by the entropy weight method [8]. Firstly, the matrix *XB* is normalized to obtain the matrix *XC* as follows:

$$y_{ij} = \frac{x_{ij} - \min(x_i)}{\max(x_i) - \min(x_i)} \quad (5)$$

where y_{ij} is the normalized value; x_{ij} is the value in the matrix *XB*; x_i is the health examination indicator. Secondly, the weight of the *j-th* health examination indicator of the *i-th* examinee in the matrix *XC* is solved as follows:

$$p_{ij} = \frac{y_{ij}}{\sum_{i=1}^{\mathrm{m}} y_{ij}} \quad (6)$$

Then, the information entropy of each health examination indicator is calculated as follows:

$$E_j = -\frac{\sum_{i=1}^{\mathrm{m}} p_{ij} \ln(p_{ij})}{\ln(m)}, j \in [1, n] \quad (7)$$

where j is the health examination indicator and n is the total number of selected health examination indicators.

Finally, the weight of each health examination indicator is calculated as:

$$W_j = \frac{1 - E_j}{n - \sum_{i=1}^{n} E_j}, j \in [1, n] \quad (8)$$

4.4 Personalized Health Index

Firstly, for matrix *XB* obtained in the second step is subjected to a direct process to yield matrix *XD*. The health examination indicators can be divided into two categories: minimal indicators and interval indicators. The smaller the value of a minimal indicator is, the healthier the result is. An interval indicator with a value within a certain interval indicates the healthy state.

As for minimal indicator, the direct method is expressed as:

$$\hat{x}_{ij} = \frac{1}{x_{ij}} \tag{9}$$

In interval indicators, if the value of indicator is in a healthy range, its value is within an interval [a, b]. Otherwise, it is unhealthy. The direct function of an interval indicator is shown in Fig. 3. For a health examination indicator, a^* is the minimum acceptable value; *a* is the minimum value in the normal range; *b* is the maximum value of the normal range; *c* is a basic health value; b^* is the maximum acceptable value. The value of the direct function is in the interval [0,1]. The larger the value is, the healthier the indicator is.

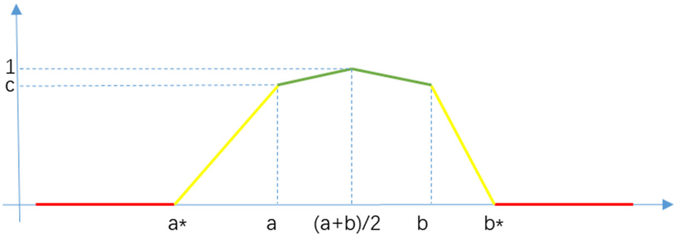


Fig. 3. Interval direct function.

The interval function is expressed as follows:

$$\hat{x}_{ij} = \begin{cases} 0, x_{ij} \le a^* \\ c \times \dfrac{x_{ij} - a^*}{a - a^*}, a^* < x_{ij} \le a \\ c + 2 \times (c - 1) \times \dfrac{x_{ij} - a}{b - a}, a < x_{ij} \le \dfrac{a+b}{2} \\ 1 + (c - 1) \times \dfrac{2x_{ij} - a - b}{b - a}, \dfrac{a+b}{2} < x_{ij} \le b \\ c \times \left(1 - \dfrac{x_{ij} - b}{b^* - b}\right), b < x_{ij} < b^* \\ 0, x_{ij} \ge b^* \end{cases} \tag{10}$$

The matrix *XD* is normalized to obtain the matrix *XE* as follows:

$$z_{ij} = \frac{\hat{x}_{ij}}{\sqrt{\sum_{i=1}^{m} \hat{x}_{ij}^2}} \tag{11}$$

The person with the most healthy status *ZBest* is composed of the maximum value of each column. The person with the least healthy status *ZWorst* is composed of the minimum value of each column.

$$ZBest = (\max\{z_{11}, \cdots, z_{m1}\}, \cdots, \max\{z_{1n}, \cdots, z_{mn}\}) \tag{12}$$

$$ZWorst = (\min\{z_{11}, \cdots, z_{m1}\}, \cdots, \min\{z_{1n}, \cdots, z_{mn}\}) \tag{13}$$

The distance $DBest_i$ between the *i-th* examinee and *ZBest* is expressed as:

$$DBest_i = \sqrt{\sum_{j=1}^{n} W_j \left(ZBest_j - z_{ij}\right)} \tag{14}$$

The distance $DWorst_i$ between the *i-th* examinee and *ZWorst* is expressed as:

$$DWorst_i = \sqrt{\sum_{j=1}^{n} W_j \left(ZWorst_j - z_{ij}\right)} \tag{15}$$

Finally, the health index of each examinee PHI_i is calculated as:

$$PHI_i = \frac{DWorst_i}{DWorst_i + DBest_i} \tag{16}$$

where PHI_i is between 0 and 1. A bigger PHI_i means an examinee with more healthy status.

5 Indicator System and Weight Calculation

In order to establish a comprehensive, objective, scientific, and practical health index, based on the analysis of Chinese disease types, death causes and the requirements of comprehensive health management, with the help of domain experts, nine physiological health indicators are selected to establish a health examination health index system (see Table 1).

The indicator of systolic blood pressure is defined as follows. The normal systolic blood pressure of an adult is 90–140 mmHg. If its values in multiple examinations were beyond the above range, it is determined as hypertension, which may cause heart diseases, stroke, memory loss, kidney damage and other diseases. The disappearance of blood pressure is a precursor to death, indicating that blood pressure has an important biological significance.

Table 1. Indicator system and corresponding normal medical ranges.

No.	Indicators	Types	Normal ranges	Main diseases
A1	Systolic blood pressure	Blood pressure	90–140 mmHg	Hypertension and ischemic heart disease
A2	Diastolic blood pressure	Blood pressure	60–90 mmHg	Hypertension
A3	Body mass index (BMI)		18.5–23.9 kg/m3	Diabetes, osteoarthritis, etc.
A4	Total cholesterol	Cholesterol	3.1–5.2 mmol/L	Cardiovascular diseases
A5	High-density lipoprotein (HDL) cholesterol	Lipoprotein	0.8–2.0 mmol/L	Heart diseases, arteriosclerosis, and stroke
A6	Low-density lipoprotein (LDL) cholesterol	Lipoprotein	0–3.7 mmol/L	Heart diseases, arteriosclerosis, and stroke
A7	Fasting blood glucose	Blood glucose	3.9–6.0 mmol/L	Diabetes
A8	Triglycerides	Blood lipids	0.4–1.7 mmol/L	Heart diseases and diabetes
A9	Thyrotropin	Thyroid	0.3–4.5 mmol/L	

The indicator of diastolic blood pressure is defined as follows. In a diastolic phase, the aortic pressure drops. At the end of the diastolic phase, the arterial blood pressure is the lowest and called the diastolic blood pressure. The normal diastolic blood pressure in adults is 60–90 mmHg.

Body mass index (BMI) = *body mass (*kg*)/the square of height* (m^2). It reflects the relationship between body mass, height and the amount of fat and is used to define obesity, normal body mass and underweight.

The indicator of total cholesterol is defined as follows. To know whether there are three highs, it is necessary to determine total cholesterol. The ideal total cholesterol is below 5.2 mmol/L. If it is more than 6.2 mmol/L, the risk of heart diseases is greatly increased.

The indicator of high-density lipoprotein (HDL) cholesterol is defined as follows. HDL cholesterol is good cholesterol in the body and can help the body to remove low-density lipoproteins from blood vessels to protect the heart.

The indicator of low-density lipoprotein (LDL) cholesterol is defined as follows. LDL cholesterol is bad cholesterol in the body and a killer that causes heart diseases, arteriosclerosis, and stroke. Especially those who already have heart diseases and diabetes need to control this index carefully.

The indicator of fasting blood sugar is defined as follows. The sugar in the blood is collectively called blood sugar. Most of the energy required for the activities of various tissues and cells in the body comes from glucose. Fasting blood sugar refers to the blood sugar value measured after fasting for at least *8 h*. Patients with diabetes can be easily subjected to kidney diseases and diabetic foot.

The indicator of triglycerides is defined as follows. The content of triglycerides in plasma can be used as the basis for diagnosing hyperlipidemia. If this index is too high, it easily leads to cardiovascular, cerebrovascular diseases and type-2 diabetes.

The indicator of thyrotropin is defined as follows. As an endocrine hormone, it can help to maintain body temperature and heart rate and regulate body metabolism, hair growth, mood, muscles and brain fluctuations. When it exceeds 4.5 mmol/L, the thyroid cannot produce enough thyroxine to supply the body, thus leading to the failure in maintaining normal body metabolism.

According to Eq. (8), the weights of the selected 9 indicators are calculated (see Table 2).

Table 2. Weights of selected indicators.

No.	Indicators	Weights
A1	Systolic blood pressure	11.34%
A2	Diastolic blood pressure	11.32%
A3	Body mass index (BMI)	11.33%
A4	total cholesterol	11.28%
A5	High-density lipoprotein (HDL) cholesterol	9.84%
A6	Low-density lipoprotein (LDL) cholesterol	11.18%
A7	Fasting blood glucose	11.19%
A8	Triglycerides	11.29%
A9	Thyrotropin	11.21%

Among 9 selected indicators (see Table 2), the indicator of systolic blood pressure has the greatest impact on health, and the high-density lipoprotein cholesterol has a relatively smaller impact. The weights of the selected 9 indicators showed no significant difference in the establishment of the health index.

6 Results Analysis

In order to help people to understand their health status and comparison with a group of people who have similar demographic data, we divide personal health index into four levels, healthy, good, sub-healthy and unhealthy, which are respectively displayed in dark green, light green, yellow and red colors (Fig. 4). The lower boundary (a) of the red level is the 5th percentile of the personal health index. The upper boundary (c) of the yellow level is the 50th percentile of the personal health index. The boundary between the red level and the yellow level is b as:

$$\mathrm{b} = (a + c)\big/2. \tag{22}$$

The upper boundary (e) of the dark green level is the 95th percentile of the personal health index. The boundary between the green level and the dark green level is d as:

$$d = (c + e)\Big/2. \tag{23}$$

The height of a color bar indicates the number of people within the level. The more people within a level corresponds to a higher color bar of the level. If the overall health status of an examinee is extremely good and belongs to the top 5% of the health index, he will fall above the dark green level. If the health status is extremely poor and belongs to the bottom 5% of the health index, he will fall below the red level.

According to the above method, the health index and corresponding health level of 133,085 samples are calculated after removing outliers. Among the obtained samples, 28,723 people belong to the healthy level, accounting for 21.6%; 31,165 people belong to the good health level, accounting for 23.4%; 18,284 people belong to the sub-healthy level, accounting for 13.7%; 41,603 people belong to the unhealthy level, accounting for 31.3%. In addition, 6,654 people are with extremely good health status, accounting for 5%; 6,654 people are extremely bad health status, accounting for 5%.

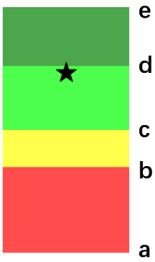


Fig. 4. Boundaries of various levels. (Color figure online)

In different health levels, six men aged 35 to 40 are randomly selected and respectively denoted as A, B, C, D, E, and F and their health indicators are calculated and compared (see Table 3).

Table 3. Calculation results of six persons with the same sex in the same age group.

Persons	A	B	C	D	E	F
Gender	Male	Male	Male	Male	Male	Male
Age	39	40	36	38	36	40
Systolic blood pressure	117	121	118	112	142	121
Diastolic blood pressure	78	69	70	60	92	69
Body mass index	19.67	21.8	25.98	25.55	28.09	25.67
Total serum cholesterol	4.12	4.68	4.63	3.94	5.38	4.65
Serum triglycerides	1.12	0.59	2.96	1.27	0.96	1.22
Serum high-density cholesterol	1.42	1.39	1.29	0.87	0.99	1.71
Serum low-density cholesterol	2.33	3.67	2.98	2.42	4.03	2.75
Fasting blood sugar	5.53	4.92	5.17	5.2	5.35	5.76
Serum thyrotropin	1.32	1.76	2.07	1.5	1.51	11.6
Health index	0.96	0.94	0.92	0.89	0.86	0.76
Percentile	97.95%	83.57%	62.18%	43.54%	24.49%	2.30%

Six people's health indexes are shown in Fig. 5.

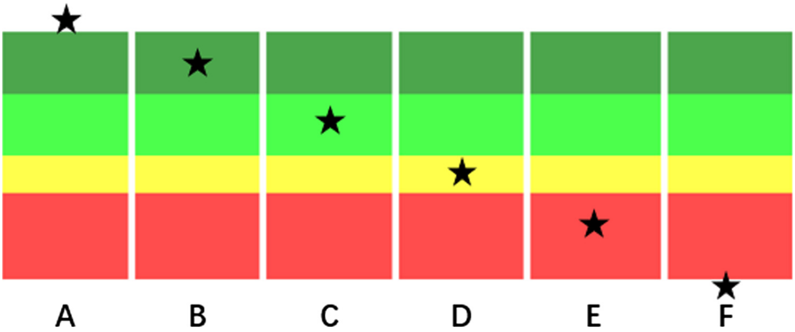


Fig. 5. Health index distribution of six men aged 35 to 40 of each health level.

Taking the examinees B and E as examples, the examinee B belongs to the healthy level and the examinee E belongs to the unhealthy level. The indicators of the examinees (B and E) and the data in Table 1 indicate that although the examinee B is aged 40, all his indicators are within the normal ranges. As for the examinee E, the values of five indicators of systolic blood pressure, diastolic blood pressure, body mass index, total cholesterol, and low-density cholesterol are all abnormally high, indicating the risks of hypertension and cardiovascular diseases. Although the examinee E is younger than the examinee B, the health index of the examinee E is not higher than that of the examinee B. The result also proves the feasibility of the health examination index.

The health index can be used to display the health changes of individuals for many years in a visualized way, as shown in Fig. 6.

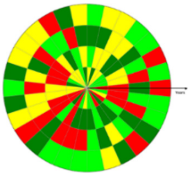


Fig. 6. Personal health status fingerprints.

In Fig. 6, the four colors indicate different health levels described above and the extension direction from the center of the circle to the exterior indicates the timeline. The line starting from the center to the outmost circumference indicates the period from the first year to the last year of the health examination. Each ring represents a health examination year. Each sector represents a certain health examination indicator or a certain type of health examination indicators. A color indicates a state of personal health index. In summary, the color of each grid indicates the health status of a certain health examination indicator of a person in a certain year.

An examinee G is randomly selected from the existing samples. According to the health examination data of the examinee G in 5 consecutive years (35 to 39 years old), the changes in the health index are shown in Fig. 7. When he is 35 years old, he is the healthiest. When he is 36 years old, the health index declines in the fastest pace. The health index slightly increases at the age of 37. Then the health index declines in the subsequent two years. Finally, at the age of 39, the health index is the worst, indicating the unhealthy level.

In the health index fingerprints of the examinee G (see Fig. 8), the center of the circle indicates the data in 2015 (35 years old) and the outermost ring indicates the data in 2019 (39 years old). Obviously, the index of diastolic blood pressure gradually becomes more and more unhealthy over time and changes from the good state to the unhealthy

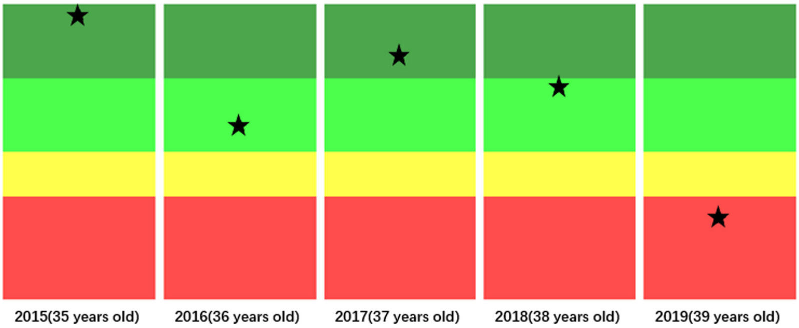


Fig. 7. Changes in the health index of examinee G.

state. The body mass index has been in a sub-health state in the 5 years. The index of total cholesterol is generally above the good state and in the sub-health state only in 2017. The index of LDL cholesterol is in a healthy state. Figure 8 shows the changes in the health status of the examinee G, intuitively and conveniently.

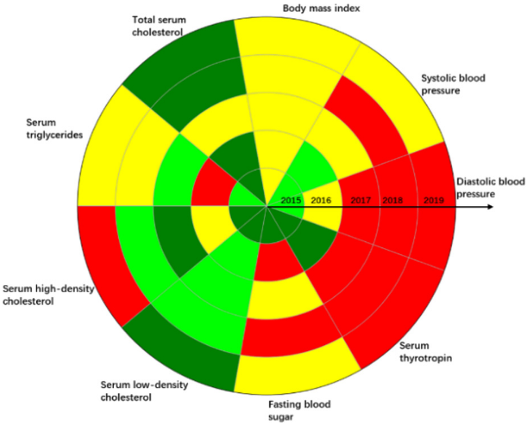


Fig. 8. Health index fingerprints of examinee G.

7 Conclusions

In this paper, we propose an approach of health examination health index and design diagrams of grade and fingerprint to visualize the health examination records effectively. It improves the interpretability of health examination records. The main conclusions are drawn as follows.

Firstly, based on the analysis of big data on health examination, the missing data of examinees are recovered with the tensor decomposition to provide a data basis for establishing the health examination health index. It's more sensible for usage of collaborative filter on big data than just recovering them with median or mean values.

Secondly, based on statistical analysis and pathology, through data mining and entropy, the weights of sensitive and key single indicators reflecting the health status are calculated to establish a resident health index. The index is more intuitively and easily understood and could comprehensively reflect the health status quantitatively for a population.

Thirdly, in the evaluation process of the health examination health index with TOPSIS method, this paper takes into account not only the medical knowledge of the health examination indicators, but also their data distribution on health examination. The TOPSIS method is suitable for the evaluation of the health index and can more scientifically evaluate the health state of residents.

Fourthly, the grade diagram and fingerprint diagram can intuitively display their ranks in the general population and similar demographic group and the changes in individual health index. Besides, they can track and judge the occurrence of potential diseases better, indicating its efficient applications.

Our research team will further improve the residential health examination health index and strengthen the verification of the health index. Aging, chronic diseases and disease prevention will become the three cores of the future health industry. We will also further expand the application scope of the health index in different professional or age groups and focus on the evaluation of the health levels of residents in various provinces and cities in China. The study will contribute to the transformation from disease treatment-centered management to health-centered management.

References

1. China Merchants Industrial Research Institute: China health examination industry market prospect research report of 2018 (Simple Version). Beijing, pp. 1–3 (2018)
2. Yang, Y., Jian, G.: The status quo and application prospects of health examination data sharing. Chin. J. Health Inf. Manag. **15**(6), 633–636 (2018)
3. Moon, Y.J., Choi, E., Hwang, Y.H.: Design and implementation of the expert system for health and medical treatment using integration of big data. J. Theoret. Appl. Inf. Tech. **96**(6), 1680–1689 (2018)
4. Fu, Q., Wang, Q.B., Jiang, Z.Q.: Theoretical study on health index. China Popul. Resour. Environ. **25**(5), 330–335 (2015)
5. Chen, L., Li, X., Quan, Z.S., et al.: Mining health examination records – a graph-based approach. IEEE Trans. Knowl. Data Eng. **28**(9), 2423–2437 (2016)
6. Xin, G., Yang, C., Yang, Q., Li, C., Wei, C.: Post-evaluation of well-facilitated capital farmland construction based on entropy weight method and improved TOPSIS model. Trans. Chin. Soc. Agric. Eng. **33**(1), 238–249 (2017)
7. Chen, L., Li, X., Yang, Y., Kurniawati, H., Sheng, Q.Z., Hu, H.Y., Huang, N.: Personal health indexing based on medical examinations: a data mining approach. Decis. Support Syst. **81**, 54–65 (2016)
8. Xia, M., Wei, Y., et al.: Human Body Health Evaluation Method based on Physical Health Indexes, CN 105962918A. (2016)
9. Wang, C.H., Chen, K.S.: New process yield index of asymmetric tolerances for bootstrap method and six sigma approach. Int. J. Prod. Econ. **219**, 216–223 (2020)
10. Genes, C., Esnaola, I., Perlaza, S.M., Ochoa, L.F., Coca, D.: Recovering missing data via matrix completion in electricity distribution systems. In: IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pp. 1–6. IEEE (2016)

Decision Support System for Acupuncture Treatment of Ischemic Stroke

Ying Shen[1](✉), Joël Colloc[2], and Armelle Jacquet-Andrieu[3]

[1] School of Electronics and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen, China
shenying@pkusz.edu.cn

[2] Université du Havre, 25 Rue Philippe Lebon, 76086 Le Havre Cedex, France
j.colloc@univ-lehavre.fr

[3] Université Paris Descartes, 45 rue des Saints-Pères, 75006 Paris, France
a.jacquet@u-paris10.fr

Abstract. In acupuncture research, the automatic selection of acupoints for treating stroke remains challenging. We design an ontology driven clinical decision support system to assist acupuncture treatment planning in the field of ischemic stroke and provide clinical decision-making services. This paper describes the development of a decision support system for the diagnosis and therapy of ischemic stroke based on clinical prior knowledge. An ontology is established in the field of acupuncture and ischemic stroke pertaining to clinical stages and their corresponding information components. Based on the ontology, the decision support system explores an acupuncture therapy for patient by considering therapy method, stimulation points and intensity, and manipulation ways with previous therapies. A performance analysis and assessment are proposed to evaluate the effectiveness of the decision support system. Experiments show that the proposed decision support system can aid ischemic stroke therapy decision-making by recommending the most relevant treatments suggestion through ontology inference. The visualization of diagnosis making and therapy planning also verifies the benefits of the combined exploration of decision support system and ontology, which promotes the development of acupuncture from individualized empirical knowledge to large-scale evidence-based medicine.

Keywords: Acupuncture · Ischemic stroke · Ontology construction · Decision-making support

1 Introduction

Acupuncture is a treatment approach that involves inserting thin needles into the body at acupoints. Acupuncture can improve the awareness, limb rigidity, and self-managing ability of stroke patients [1]. However, in acupuncture research, the study of acupoints

This work was financially supported by the Shenzhen General Research Project (No. JCYJ20190808182805919).

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 582–594, 2020.
https://doi.org/10.1007/978-3-030-65390-3_44

automatic selection remains challenging, since it must deal with unstructured clinical information and fuzzy descriptions of patients.

Stroke, also known as cerebrovascular accident, cerebrovascular insult or brain attack, refers to insufficient blood flow to the brain causing cell death. There are two main types of stroke: ischemic due to lack of blood flow and hemorrhagic due to bleeding [2]. In 2013, stroke was the second leading cause of death after coronary artery disease, accounting for 6.4 million deaths (12% of the total) [3]. Approximately 6.9 million people suffer from ischemic stroke and 3.4 million people have hemorrhagic stroke [4]. It should be noted that acupuncture treats the stroke's sequelae rather than its acute phase. When the acute crisis is over, acupuncture can reduce side effects and provides long-term beneficial effects during the phrase of rehabilitation.

The purpose of this study is to construct an acupuncture ontology and decision-making system. First, we design a decision support system (DSS) based on the generated acupuncture ontology and the knowledge of ischemic stroke. We then process user queries based on the structured ontology. OWL functions such as equivalent, inheritance and subsumption enhance the ability to search for related concepts from a built ontology. Finally, we design an optimized sorting algorithm based on link relationships to sort query results and display the most relevant results.

2 Related Work

Different from the booming study of DSS in the field of western medicine, the intelligent automatic diagnosis of acupuncture is still underexplored. Lee et al. [5] use the term frequency inverse document frequency (TF-IDF) method to analyze the association between acupoints and disease patterns. Based on the [5], Jung et al. [6] extract data on indications of each acupoint based on the frequency of the simultaneous occurrence of eight source points and eighteen disease sites. Feng et al. [7] discover acupoints and acupoint combinations based on data mining. Experience is performed through methods of statistics, apriori association rules and k-core approaches to analyze the acupoints combination to deal with the relationship between vascular dementia syndrome.

Some recent studies have used ontology driven clinical DSS to better learn semantic information and conduct knowledge inferences [8, 9]. For example, Gene Ontology (GO), UMLS, and SNOMED are widely used to aid practical diagnosis [10–12]. In Chinese medicine, there are two useful knowledge bases: the Traditional Chinese Medicine Language System (TCMLS) [13] and the knowledge-based Chinese Medical Diagnostic System (CMDS) [14]. TCMLS is a large-scale computerized language system that contains approximately 100,000 concepts, 300,000 terminologies, and 1.27 million semantic relationships. CMDS is based on a comprehensive medical ontology with domain knowledge of digestive system diseases. These ontologies are great achievements in medicine, but they lack or contain incomplete acupuncture data [15]. In this study, we build an acupuncture ontology to reduce the ambiguity of the clinical inference and improve the diagnosis and treatment of ischemic stroke.

3 Clinical Procedure

3.1 Diagnosis

From a Chinese medicine perspective, the purpose of diagnosis is to determine which meridians are unbalanced.

- **Patients' Condition.** The patient's specific state and physical condition determine his particular symptoms and degree of manifestation. The presentation of symptoms and their relative strength will vary from patient to patient. Guided by the pulses, one can determine the specific energetic disharmonies that cause illness and lead to other illnesses if not redressed [16].
- **Clinical Sign and Radial Pulse.** Acupuncturists identify disease through the study of clinical signs and events, which is reflected in medical and surgical semiology [17]. Pulse can give us the clue about the cause of disease. Its indications indicate the malfunctioning organ-meridian and even the point of the meridian [18]. This information allows the determination of meridian in which the equilibrium must be restored and the group of points advisable to use.
- **Points that needed to be Stimulated or be Deduced its Energy.** The condition of the pulse indicates abnormal organs and the meridians and pulses that need to be corrected [19]. Certain points have a special power for which the meridian connections are sometimes difficult to understand, which is worth to be considered by the DSS. For example, ST-36 for nerves; TW-10 for excess; BL-60 and CV-6 for pains.
- **Assistance of Western Medicine.** WM.1 of Fig. 1 is related to the **Positive Diagnosis** process, which includes two lists: a list of matching signs and component lists of disease syndromes, and a list of matching diseases that match the patient's medical history. In the step of positive diagnosis, acupuncturists take into account patients' current complaints, clinical states, treatment history, and genetic and social background [20].

  In **Differential Diagnosis**, the acupuncturist needs to search for the presence of specific signs to verify and confirm the previously stated hypotheses. As shown in Fig. 2, the differential diagnosis of ischemic stroke relies on the objects of each syndrome. This procedure can help to diagnose ischemic stroke with hemorrhagic cerebrovascular disease or ischemic cerebrovascular disease, its severity and possible location.

3.2 Therapy

To choose acupunctural therapy, some knowledge is needed, including treatment methods (needle insertion, moxibustion, and cupping therapy), stimulation points and intensity (acupuncture points and meridians structures), manipulation ways (spinning, flicking, or moving up and down relative to the skin), treatment cycle (a single complaint involves six to twelve treatments), treatment duration (a single complaint lasts over a few months), needle insertion number (approximately 5 to 20 needles), needle insertion duration (10 to 20 min), and needle insertion depth.

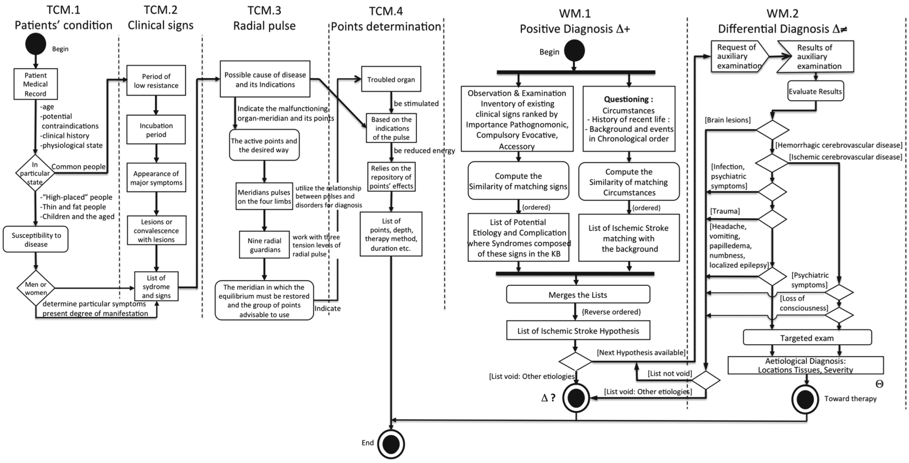

TCM.1
Patients' condition
TCM.2
Clinical signs
TCM.3
Radial pulse
TCM.4
Points determination
Begin
Patient Medical Record
-age
-potential contraindications
-clinical history
-physiological state
In particular state
Common people
-"High-placed" people
-Thin and fat people
-Children and the aged
Susceptibility to disease
Men or women
determine particular symptoms
present degree of manifestation
Period of low resistance
Incubation period
Appearance of major symptoms
Lesions or convalescence with lesions
List of sydrome and signs
Possible cause of disease and its Indications
Indicate the malfunctioning organ-meridian and its points
The active points and the desired way
Meridians pulses on the four limbs
utilize the relationship between pulses and disorders for diagnosis
Nine radial guardians
work with three tension levels of radial pulse
The meridian in which the equilibrium must be restored and the group of points advisable to use
Indicate
Troubled organ
be stimulated
Based on the indications of the pulse
be reduced energy
Relies on the repository of points' effects
List of points, depth, therapy method, duration etc.
End
WM.1
Positive Diagnosis Δ+
Begin
Observation & Examination Inventory of existing clinical signs ranked by Importance Pathognomonic, Compulsory Evocative, Accessory
Questioning : Circumstances - History of recent life : - Background and events in Chronological order
Compute the Similarity of matching signs
Compute the Similarity of matching Circumstances
{ordered}
{ordered}
List of Potential Etiology and Complication where Syndromes composed of these signs in the KB
List of Ischemic Stroke matching with the background
Merges the Lists
{Reverse ordered}
List of Ischemic Stroke Hypothesis
[Next Hypothesis available]
[List void: Other etiologies]
Δ ?
[List not void]
[List void: Other etiologies]
WM.2
Differential Diagnosis Δ≠
Request of auxiliary examination
Results of auxiliary examination
Evaluate Results
[Brain lesions]
[Hemorrhagic cerebrovascular disease]
[Ischemic cerebrovascular disease]
[Infection, psychiatric symptoms]
[Trauma]
[Headache, vomiting, papilledema, numbness, localized epilepsy]
[Psychiatric symptoms]
[Loss of consciousness]
Targeted exam
Aetiological Diagnosis: Locations Tissues, Severity
Θ
Toward therapy

Fig. 1. Clinical diagnosis.

For therapy, some rules needed to be considered. All actions on the meridian, point and pulse should be performed according to the husband-wife law (symmetrical right-left pulses) and the mother-son law (upstream-downstream in the flow of the energy except for the unequal coupled meridians), and effects on the same side of the body (ipsolateral) or on the opposite side (contralateral) [21].

4 Ontology Driven Decision Support System

4.1 Acupuncture Ontology Construction

Data Resource. Our study is based on 60,000 medical records related to ischemic stroke. In addition, for the literature on acupuncture treatment of ischemic stroke, several databases are extracted, including PubMed (1966–2016) [22], China Biology Medicine disc (SinoMed)[1] (1978–2016) [23], and China National Knowledge Infrastructure (CNKI)[2] (1999–2016) [24]. For acupuncture prescriptions, clinical cases that involves adult participants diagnosed with ischemic stroke are included. Language is restricted to Chinese and English.

Data Annotation. This paper uses lexical-semantic analysis annotation [25] to attach names, attributes, comments and descriptions to the text and provides metadata about existing data. Take the labeling results of Acupoint Lianquan as an example. Several types of classes are recognized and extracted, including acupoint name "Lianquan", acupoint code "RN23", reference "Huangdi Neijing-Suwen", function "Collecting and leading Yin liquid", indications "Tongue gall, tongue sharply contracted, longitudinal tongue saliva, stiff tongue, dry tongue and mouth, mouth sores, sudden aphonia, pharyngitis, deaf, cough, asthma, diabetes, poor appetite.", point compatibility "Curing aphasia with

[1] http://www.sinomed.ac.cn.
[2] https://www.cnki.net.

stiff tongue, tongue swelling, salivation and tongue sudden aphonia with EX-HN12, RN22 and LU11.", etc.

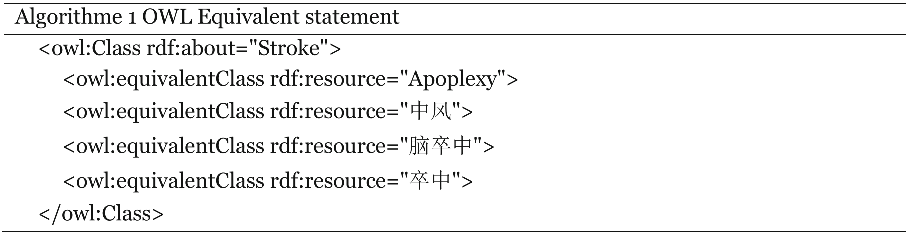
According to annotations, triples are used to record the medical entities and their corresponding information. For example, <Indications, Lianquan, tongue swelling> means that acupoint Lianquan is an indication for treating the tongue swelling. Some common relationships are shown in Table 1.

Table 1. Semantic relationships for the triple description

Terms	Semantic relationship	Related terms
Acupoint	Channel tropism	Channels
	Pass through	Channels
	Treatment	Diseases
	Treatment	Syndrome
	The effect of	Function
	Be included	Specific acupoint
	Locate	Physical orifices
	Adjacent to	Certain acupoints or landmars
	Prohibit the use of	Therapy

Ontology Development. We use Protégé as the ontology development tool and adopt Jena as the inference engine to infer the semantic relations in the ontology. The used functions of Jena include RDF reading and writing capacities, ontology processing capacities, and rule-based reasoning capacities. The following OWL functions are used to improve the triples reasoning:

- **Equivalent Syndromes.** There are some circumstances that exist multiple names for one identical thing, especially in medicine. The built-in OWL property owl:equivalentClass statements are often used for definition mapping between ontologies. The owl:equivalentClass links a class description to another class description, which indicates that two URI references actually refer to the same concept; namely, they have the same class extension [26]. For an individual syndrome class such as "Stroke", we can state that the following three URI references actually refer to the same kind of syndrome:



Algorithme 1 OWL Equivalent statement

```
<owl:Class rdf:about="Stroke">
   <owl:equivalentClass rdf:resource="Apoplexy">
   <owl:equivalentClass rdf:resource="中风">
   <owl:equivalentClass rdf:resource="脑卒中">
   <owl:equivalentClass rdf:resource="卒中">
</owl:Class>
```

- **Syndrome Inheritance.** In the ontology, the inheritance relationship between syndrome classes is defined in the categorical hierarchy of syndromes in the form of triple $\langle C_1, \mathit{rdfs} : \mathit{subClassOf}, C_2 \rangle$. This relationship shows a subordinate relationship between C_1 and C_2 and indicates the inheritance relationships between syndromes in the diagnosis of acupuncture.
- In the field of traditional Chinese medicine, the types of syndromes are mainly divided into four categories: heat and cold, zang-fu, deficiency and excess, and exterior and interior. According to the diagnostic principle, the symptoms related to subclass syndrome inherit the symptoms related to superclass syndrome.
- **Syndrome Subsumption.** Each acupuncture literature may contain many acupuncture points, and each condition may point to a different disease. Thereby, a labeled acupoint may appear in varies triples with several annotation combinations. The built-in property owl:unionOf links a class to a list of class descriptions. The owl:unionOf statement describes an anonymous class whose extension includes those individuals that appear in the class extension of at least one class description in the list.

Algorithme 2 OWL Subsumption statement

```
<owl:Class rdf:about= "hyperactivity of liver yang">
    <owl:equivalentClass>
      <owl:Restriction>
          <owl:onProperty rdf:resource= "syndrome"/>
          <owl:someValuesFrom rdf:resource= "Liver and kidney deficiency"/>
      </owl:Restriction>
    </owl:equivalentClass>
    <owl:equivalentClass>
      <owl:Restriction>
          <owl:onProperty rdf:resource= "syndrome">
          <owl:someValuesFrom rdf:resource= "Mouth askew"> </owl:Restriction>
    </owl:equivalentClass>
</owl:Class>
```

Graph Generation. Ontology describes the disease as a series of medical conditions, including syndromes composed of pathognomonic, compulsory, evocative, or accessary clinical signs related to pathology, that express the importance of the sign in a clinical picture [27]. The annotation graph is obtained from electronic medical records or literatures through the extraction of entities and concepts, including treatment time, symptom, acupoint etc. According to the annotation, triples are used to record the patient-related treatment information. For example, on February 15, 2016, the patient No. 20124567e2 with symptom stiff tongue was treated with Lianquan (RN23) and Tongli (HT5) acupoints. The corresponding triples (Patient, Symptom, Acupoint) can be recorded as (20124567e2, Stiff tongue, Lianquan (RN23)) and (20124567e2, Stiff tongue, Tongli (HT5)).

The initial ontologies of meridian and syndromes (Fig. 2) are the basis of ontology extension. The ontology enrichment is carried out through the matching approach by associating the concepts or entities that are identical between the built acupuncture ontology and the annotation graph of medical records. To achieve the purpose of the annotation, an additional "annotation" class is added to the ontology to store the associated relations between ontology class and medical record.

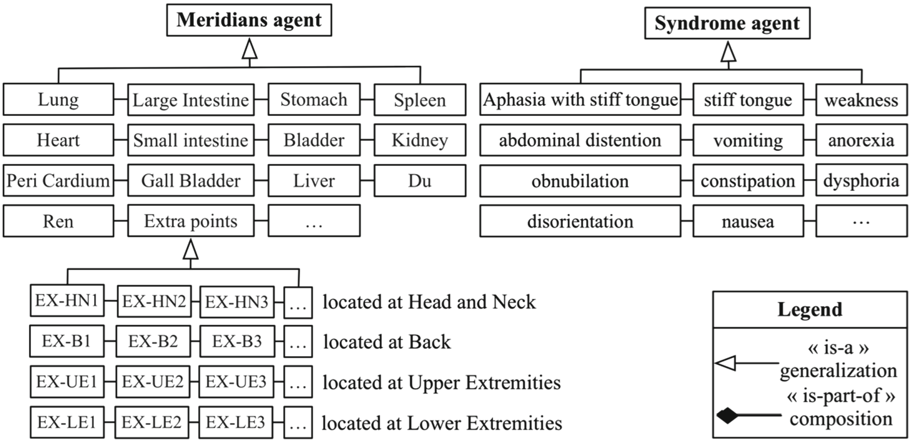


Fig. 2. Class diagram of "Meridian" and "Syndrome" ontology.

4.2 Decision Support System

The proposed DDS aims to match current cases with previously diagnosed cases, rank related cases, and then refer to their treatment planning.

Diagnosed Case Matching. The core idea of ontology-based text mining method is to determine whether the text keywords $Q(q_1, q_2, \ldots, q_n)$ match the existing ontology entities $C(c_1, c_2, \ldots, c_m)$. If yes, C will be added to the searching set $S(s_1, s_2, s_3 \ldots)$; otherwise, the texts containing text keywords will be extracted and match with ontology concepts.

Here, we consider how to assess the ontology entities $C(c_1, c_2, \ldots, c_m)$ and indicate its correlation degree with text keywords $Q(q_1, q_2, \ldots, q_n)$. The assessment function $A(c, Q|T)$ is adopted here, where T represents the text set to be retrieved. The co-occurrence frequency of q and c in a text is defined as $fqc(c, q|T) = tf(c|D) * tf(d|D)$, where $tf(*|D)$ indicates the number of concepts or words appear in the text T. The calculation formula of the assessment function is,

$$A(c, Q|T) = \sum\nolimits_{q \in Q} idf(q|T) idf(c|T) \log\big(f_{qc}(c, q|T) + 1.0\big) \tag{1}$$

Afterwards, we add the top K ontology concepts with highest co-occurrence rate to the searching set $S(s_1, s_2, s_3 \ldots)$, then consider both text keywords and searching set as the extension text words.

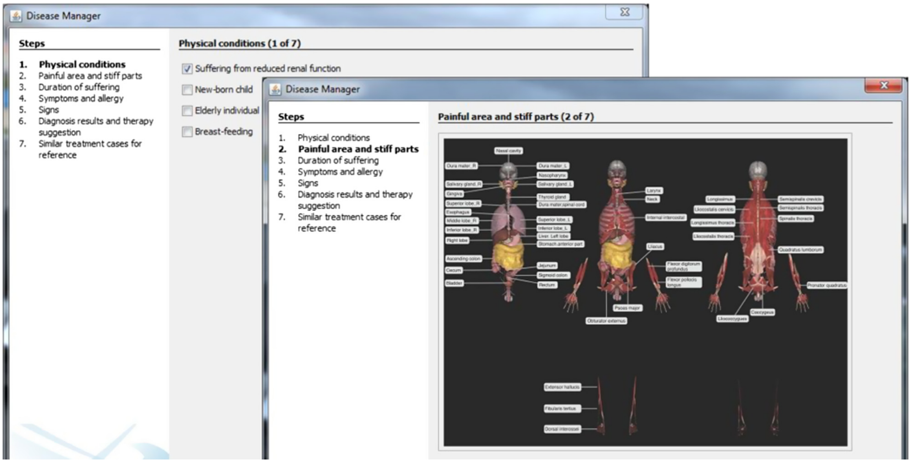
Diagnosed Case Ranking. The ranking weight in this paper is defined as $W = W_1 \cap W_2$. W_1 is the domain correlation degree, which refers to the relevance between target domain and diagnosed case concept/relation. The more diagnosed case concepts and relationships that belong to the target domain, the higher the domain correlation degree. TF-IDF method [28] is used to calculate its value in the field of ischemic stroke.

The calculation of W_2 is related to the semantic correlation length between two concepts. When calculating W_2, it is logically considered that ontology is a graph with a huge network structure, the nodes are entities, and the edge are relations. Since there may be multiple links between two given entities, we set all edge weights to 1 and adopt the Dijkstra's algorithm [29] to find the shortest paths between nodes in a graph. The core idea of this algorithm is to compare the distance from the starting node to other unlabeled nodes with the distance from the starting node to the labeled nodes. If the former is smaller, the correlation will be updated. This algorithm is an iterative process that calculates the correlations of all nodes. The ranking formula is therefore defined as $W = k_1 \times W_1 + k_2 \times W_2$, where $k_1 + k_2 = 1$, the user can assign values to variable k_1 and k_2 according to the actual needs of the application.

5 Experimental Results

5.1 Diagnosis and Therapy of Ischemic Stroke

Figure 3 illustrates the diagnosis and acupuncture treatment of ischemic stroke. As shown in Fig. 3, the diagnosis of ischemic stroke takes into account the patient's physical condition, painful area and stiff part, duration of suffering, syndromes and allergies, and clinical signs. In the selection of painful areas and stiff parts, we use human 2D maps to simplify user selection.



Disease Manager

Steps

1. **Physical conditions**
2. Painful area and stiff parts
3. Duration of suffering
4. Symptoms and allergy
5. Signs
6. Diagnosis results and therapy suggestion
7. Similar treatment cases for reference

Physical conditions (1 of 7)

☑ Suffering from reduced renal function

□ New-born child

□ Elderly individual

□ Breast-feeding

Disease Manager

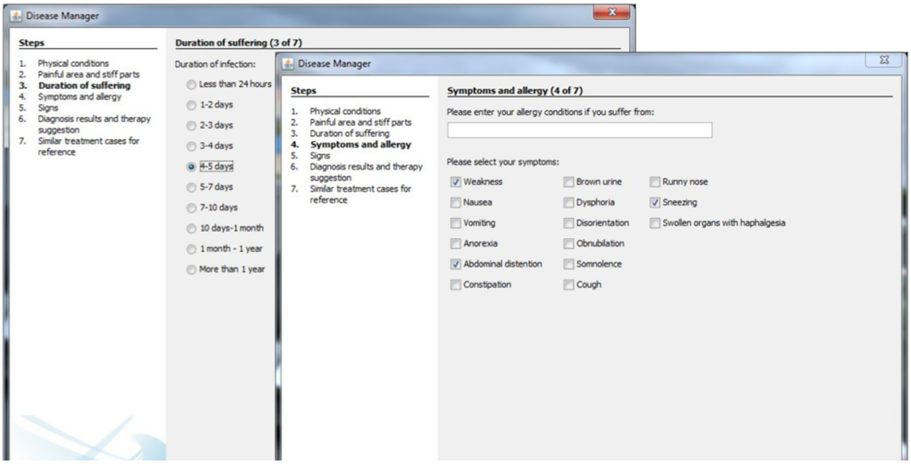
Steps

1. Physical conditions
2. **Painful area and stiff parts**
3. Duration of suffering
4. Symptoms and allergy
5. Signs
6. Diagnosis results and therapy suggestion
7. Similar treatment cases for reference

Painful area and stiff parts (2 of 7)

Fig. 3. Selection of physical condition, painful area and stiff part.

Because different diseases have different incubation and exacerbation periods, the therapy planning should be different according to the duration of suffering (see Fig. 4 left panel). Available options include initial treatment, prolonged course of disease, and improved conditions. All three options have units of measurement in days and can specify different time ranges. In the next step, patients can select symptoms according to their conditions, such as weakness, nausea, vomiting, anorexia, etc.



Disease Manager

Steps

1. Physical conditions
2. Painful area and stiff parts
3. **Duration of suffering**
4. Symptoms and allergy
5. Signs
6. Diagnosis results and therapy suggestion
7. Similar treatment cases for reference

Duration of suffering (3 of 7)

Duration of infection:

○ Less than 24 hours
○ 1-2 days
○ 2-3 days
○ 3-4 days
● 4-5 days
○ 5-7 days
○ 7-10 days
○ 10 days-1 month
○ 1 month - 1 year
○ More than 1 year

Disease Manager

Steps

1. Physical conditions
2. Painful area and stiff parts
3. Duration of suffering
4. **Symptoms and allergy**
5. Signs
6. Diagnosis results and therapy suggestion
7. Similar treatment cases for reference

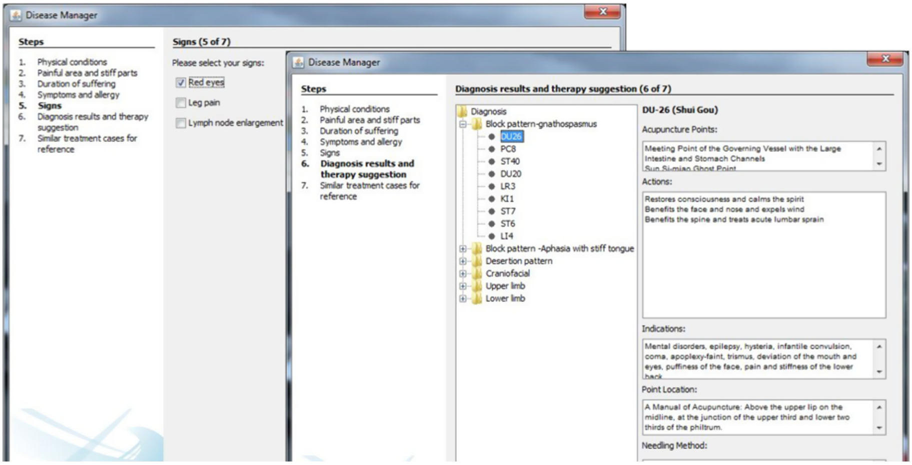
Symptoms and allergy (4 of 7)

Please enter your allergy conditions if you suffer from:

Please select your symptoms:

☑ Weakness
□ Nausea
□ Vomiting
□ Anorexia
☑ Abdominal distention
□ Constipation
□ Brown urine
□ Dysphoria
□ Disorientation
□ Obnubilation
□ Somnolence
□ Cough
□ Runny nose
☑ Sneezing
□ Swollen organs with haphalgesia

Fig. 4. Selection of duration of suffering, symptoms and allergies.



Disease Manager

Steps

1. Physical conditions
2. Painful area and stiff parts
3. Duration of suffering
4. Symptoms and allergy
5. **Signs**
6. Diagnosis results and therapy suggestion
7. Similar treatment cases for reference

Signs (5 of 7)

Please select your signs:

☑ Red eyes
□ Leg pain
□ Lymph node enlargement

Disease Manager

Steps

1. Physical conditions
2. Painful area and stiff parts
3. Duration of suffering
4. Symptoms and allergy
5. Signs
6. **Diagnosis results and therapy suggestion**
7. Similar treatment cases for reference

Diagnosis results and therapy suggestion (6 of 7)

Diagnosis
- Block pattern-gnathospasmus
  - DU26
  - PC8
  - ST40
  - DU20
  - LR3
  - KI1
  - ST7
  - ST6
  - LI4
- Block pattern -Aphasia with stiff tongue
- Desertion pattern
- Craniofacial
- Upper limb
- Lower limb

DU-26 (Shui Gou)

Acupuncture Points:

Meeting Point of the Governing Vessel with the Large Intestine and Stomach Channels
Sun Si-miao Ghost Point

Actions:

Restores consciousness and calms the spirit
Benefits the face and nose and expels wind
Benefits the spine and treats acute lumbar sprain

Indications:

Mental disorders, epilepsy, hysteria, infantile convulsion, coma, apoplexy-faint, trismus, deviation of the mouth and eyes, puffiness of the face, pain and stiffness of the lower back

Point Location:

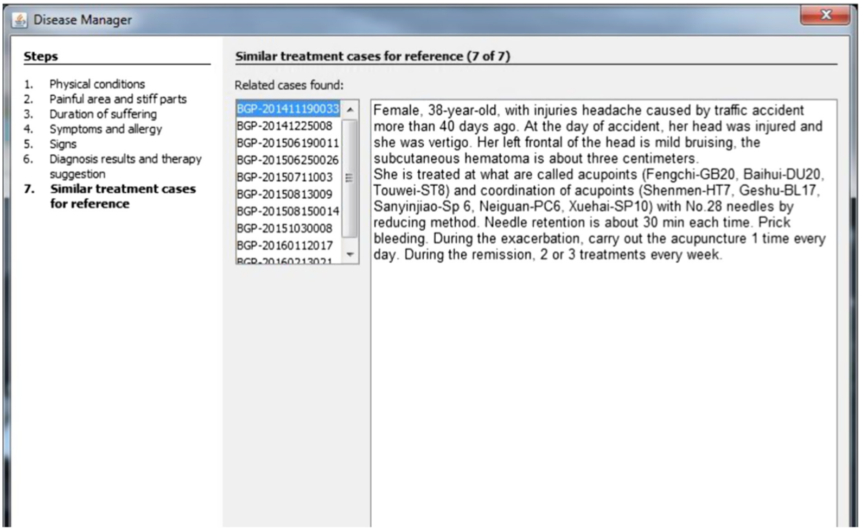
A Manual of Acupuncture: Above the upper lip on the midline, at the junction of the upper third and lower two thirds of the philtrum.

Needling Method:

Fig. 5. Diagnosis results and therapy suggestion for treating ischemic stroke.

In Fig. 5, the patient selects the signs that correspond to his condition, such as red eyes, leg pain and lymph node enlargement. With the help of the generated ontology and the information provided by the previous steps, the DSS makes a preliminary diagnosis of the disease and provides a series of acupoints for treating the disease. Detailed information about acupuncture points is displayed on the right panel of Fig. 5 for operation reference.

With the clinical signs and syndromes provided by the user in the previous steps, DSS can match previous relevant diagnosed cases. In Fig. 6, we show relevant clinical



Steps

1. Physical conditions
2. Painful area and stiff parts
3. Duration of suffering
4. Symptoms and allergy
5. Signs
6. Diagnosis results and therapy suggestion
7. **Similar treatment cases for reference**

Similar treatment cases for reference (7 of 7)

Related cases found:

BGP-201411190033
BGP-20141225008
BGP-201506190011
BGP-201506250026
BGP-20150711003
BGP-20150813009
BGP-201508150014
BGP-20151030008
BGP-20160112017
BGP-20160213021

Female, 38-year-old, with injuries headache caused by traffic accident more than 40 days ago. At the day of accident, her head was injured and she was vertigo. Her left frontal of the head is mild bruising, the subcutaneous hematoma is about three centimeters.
She is treated at what are called acupoints (Fengchi-GB20, Baihui-DU20, Touwei-ST8) and coordination of acupoints (Shenmen-HT7, Geshu-BL17, Sanyinjiao-Sp 6, Neiguan-PC6, Xuehai-SP10) with No.28 needles by reducing method. Needle retention is about 30 min each time. Prick bleeding. During the exacerbation, carry out the acupuncture 1 time every day. During the remission, 2 or 3 treatments every week.

Fig. 6. Presentation of similar diagnosed cases for treating ischemic stroke.

cases extracted by DSS. With reference to these similar cases, clinicians can more easily make treatment plans for ischemic stroke.

5.2 Assessment of DSS Performance

We use recall, accuracy and BLEU in 10 similar diagnosed cases as metrics. BLEU [30] measures the average n-gram precision on a set of reference sentences, with a penalty for overly short sentences. Since BLEU is based on the match of n-gram, failing to capture the semantic similarity, we also implement human evaluations to test the accuracy of the results. We ask the human raters to give a ranking of the similar diagnosed cases according to the overall quality. The higher these metrics, the better the results.

As shown in Table 2, we adopt Naïve Bayes as a baseline. We also report the results of our proposed DSS in diagnosis and similar treatment case recommendation through the ablation tests of discarding ontology (DSS w/o ontology) and diagnosed case matching (DSS w/o diagnosed case matching), respectively. We observe that our model substantially and consistently outperforms ablation models. For instance, our model improves the recall rate by 7.6% compared to the "DSS w/o ontology" model in diagnosis. This improvement results from the ontology-aware knowledge introduction and text matching implementation, thereby alleviating the limitations of knowledge sparseness and ambiguity.

Table 2. Performance results of diagnosis and similar treatment case recommendation

Diagnosis			
	Recall@10 (%)	Accuracy (%)	Human evaluation
Naïve Bayes	32.7	42.3	2.18
DSS w/o ontology	40.1	48.6	2.95
DSS w/o diagnosed case matching	47.3	55.7	3.36
DSS	47.7	54.1	3.91

Similar treatment case recommendation				
	Recall@10 (%)	Accuracy (%)	BLEU (%)	Human evaluation
Naïve Bayes	34.3	40.2	5.97	2.00
DSS w/o ontology	39.2	48.9	6.21	2.29
DSS w/o diagnosed case matching	42.7	51.2	6.59	1.94
DSS	49.6	56.2	18.64	3.68

6 Discussion and Conclusion

As an alternative source of healthcare, acupuncture is interpreted as an intangible connection between human and nature. In this study, a decision support system is developed to learn ontology-based diagnostic knowledge and provide clinical decision-making service for acupuncture treatment of ischemic stroke. With the help of the constructed ontology, we explore a method of interrogative reasoning for the diagnosis of ischemic stroke. The experimental results indicate that acupuncture ontology shows great potential in clinical reasoning by interpreting the correspondence between acupoints and syndromes. In the next phase of studies, we will create a virtual reality simulation of acupuncture therapy in a decision support system.

Acknowledgement. This work was financially supported by the National Natural Science Foundation of China (No. 61602013), the Shenzhen Key Fundamental Research Project (JCYJ 20190808182805919).

References

1. Denise, A., Florence, C., Hsing, J., et al.: The safety of pediatric acupuncture: a systematic review. Pediatrics **128**(6), e1575–e1587 (2011)
2. Brainin, M., Heiss, W.D. (eds.): Textbook of Stroke Medicine. Cambridge University Press, Cambridge (2019)
3. Naghavi, M., Wang, H., Lozano, R., et al.: Global, regional, and national age-sex specific all-cause and cause-specific mortality for 240 causes of death, 1990–2013: a systematic analysis for the Global Burden of Disease Study 2013. Lancet **385**(9963), 117–171 (2015)

4. Vos, T., et al.: Global, regional, and national incidence, prevalence, and years lived with disability for 301 acute and chronic diseases and injuries in 188 countries, 1990–2013: a systematic analysis for the Global Burden of Disease Study 2013. Lancet **386**(9995), 4244–4249 (2015)
5. Lee, T., Jung, W.M., Lee, I.S., et al.: Data mining of acupoint characteristics from the classical medical text: DongUiBoGam of Korean Medicine. Evid. Based Complementary Alternat. Med. **2014**, 329563 (2014)
6. Jung, W.M., Lee, T., Lee, I.S., et al.: Spatial patterns of the indications of acupoints using data mining in classic medical text: a possible visualization of the meridian system. Evid. Based Complementary Alternat. Med. eCAM **2015**(2) (2015)
7. Feng, S., Ren, Y., Fan, S., et al.: Discovery of acupoints and combinations with potential to treat vascular dementia: a data mining analysis. Evid. Based Complementary Alternat. Med. **2015**(9) (2015)
8. Dissanayake, P.I., Colicchio, T.K., Cimino, J.J.: Using clinical reasoning ontologies to make smarter clinical decision support systems: a systematic review and data synthesis. J. Am. Med. Inform. Assoc. **27**(1), 159–174 (2020)
9. Rodriguezgonzalez, A., Hernandezchan, G., Colomopalacios, R., et al.: Towards an ontology to support semantics enabled diagnostic decision support systems. Curr. Bioinform. **7**(3), 234–245 (2012)
10. Gene Ontology Consortium: Gene ontology consortium: going forward. Nucleic Acids Res. **43**(Databaseissue), 1049–1056 (2015)
11. Bodenreider, O.: The unified medical language system (UMLS): integrating biomedical terminology. Nucleic Acids Res. **32**(Databaseissue), D267–D270 (2004)
12. Agrawal, A., Gai, E.: Contrasting lexical similarity and formal definitions in SNOMED CT: consistency and implications. J. Biomed. Inform. **47**(2), 192–198 (2014)
13. Ma, J., Chen, H.: Complex network analysis on TCMLS sub-ontologies. In: Third International Conference on Semantics, Knowledge and Grid, pp. 551–553. IEEE (2007)
14. García-Crespo, Á., Rodríguez, A., Mencke, M., et al.: ODDIN: ontology-driven differential diagnosis based on logical inference and probabilistic refinements. Expert Syst. Appl. **37**(3), 2621–2628 (2010)
15. Ma, S.M., et al.: Gene-level regulation of acupuncture therapy in spontaneously hypertensive rats: a whole transcriptome analysis. Evid. Based Complementary Alternat. Med. **2019** (2019)
16. Witt, C.M., et al.: The effect of patient characteristics on acupuncture treatment outcomes. Clin. J. Pain **35**(5), 428–434 (2019)
17. Zhao, L., et al.: The long-term effect of acupuncture for migraine prophylaxis: a randomized clinical trial. JAMA Intern. Med. **177**(4), 508–515 (2017)
18. Ng, H.P., Huang, C.M., Ho, W.C., Lee, Y.C.: Acupuncture differentially affects the high-frequency spectral energy in radial pulse positions depending on type of lower back pain. Evid. Based Complementary Alternat. Med. **2019** (2019)
19. Shin, J.Y., Lee, J.H., Ku, B., Bae, J.H.: Effects of acupuncture stimulation on the radial artery's pressure pulse wave in healthy young participants: protocol for a prospective, single-arm, exploratory, clinical study. J. Pharmacopunct. **19**(3), 197 (2016)
20. Wen, J.J., Chou, H.: Integration of Chinese and Western medicine in fainting during acupuncture treatment. In: Smart Science, Design & Technology: Proceedings of the 5th International Conference on Applied System Innovation (ICASI 2019), Fukuoka, Japan, 12–18 April 2019, p. 109. CRC Press, 1 November 2019
21. Wu, D.: Human body meridian spatial decision support system for clinical treatment and teaching of acupuncture and moxibustion. Zhongguo zhen jiu = Chin. Acupunct. Moxib. **36**(1), 95 (2016)
22. Doms, A., Schroeder, M.: GoPubMed: exploring PubMed with the gene ontology. Nucleic Acids Res. **33**(suppl_2), W783–W786 (2005)

23. http://www.sinomed.ac.cn
24. https://www.cnki.net
25. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. Comput. Linguist. **31**(1), 71–106 (2005)
26. Wu, J.Z., Yu, X., Gao, W.: Distance computation of ontology vector for ontology similarity measuring and ontology mapping. J. Differ. Equ. Appl. **23**(1–2), 30–41 (2017)
27. Shen, Y., Colloc, J., Jacquet-Andrieu, A., Lei, K.: Emerging medical informatics with case-based reasoning for aiding clinical decision in multi-agent system. J. Biomed. Inform. **56**, 307–317 (2015)
28. Ramos, J.: Using TF-IDF to determine word relevance in document queries. In: Proceedings of the First Instructional Conference on Machine Learning, vol. 242, pp. 133–142, 3 December 2003
29. Jasika, N., Alispahic, N., Elma, A., Ilvana, K., Elma, L., Nosovic, N.: Dijkstra's shortest path algorithm serial and parallel execution performance analysis. In: 2012 Proceedings of the 35th International Convention MIPRO, pp. 1811–1815. IEEE, 21 May 2012
30. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics, 6 July 2002

Suicide Risk Assessment Model Based on Fuzzy Mathematics

Yuan Xu[1], Lijuan Shang[1], Jianfen Xu[1], Qunxia Gao[1,2](✉), and Dongmao Chen[1]

[1] Neusoft Institute Guangdong, Foshan 528225, Guangdong, China
shanglijuan@nuit.edu.cn, 604339203@qq.com
[2] South China Institute of Software Engineering.GU, Guangzhou 510990, China

Abstract. At present, most of the research on suicide stays in the traditional fields of sociology and psychology. In order to make suicide evaluation more objective, this paper uses fuzzy mathematics to analyze the reason of suicide. A suicide risk assessment model is established according to social factors and self-factors to assess whether individuals have suicidal tendencies. Firstly, it is important to determine the evaluation set of fuzzy comprehensive evaluation of suicide factors by fuzzy mathematics and membership function and the boundary value of the risk level by k-means clustering algorithm, and then pass the fuzzy evaluation the factor set and the comprehensive evaluation set calculate the single-factor fuzzy evaluation matrix, and normalize it, finally the mathematical model is established to get suicide risk score evaluation. The judgment matrix in the analytic hierarchy process is used to construct the judgment matrix of suicide factors to calculate the CR value to judge the validity of the matrix, and then judge the validity of the model. Through calculation, the CR value is 0.0227, which is far less than 0.1. Experiments show that the constructed suicide risk evaluation model is reasonable and can be used in the practical application of psychological evaluation.

Keywords: Suicide · Fuzzy mathematics · K-means · Analytic hierarchy process

1 Introduction

In recent years, the suicide crisis in society and universities has become more and more serious, and the incident has also attracted the attention of all sectors of society. According to a statistical data provided by the World Health Organization, there are about 800,000 suicide deaths worldwide every year, and one person dies every 40 s. Among them, about 250,000 suicides die in China each year, and the number of suicide attempts in China each year It is up to 2 million [1]. Therefore, to solve the suicide problem of modern people, it is necessary to focus on suicide research on the analysis of suicide factors, view and analyze suicide problems from different perspectives of society, find the factors that trigger suicide, and then draw the corresponding suicide Effective methods and strategies of behavior restrain the rising trend of suicide in modern society.

Suicide is also the main object of sociological research. Durkheim defined suicide as: any death directly or indirectly caused by the positive or negative actions taken by

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 595–609, 2020.
https://doi.org/10.1007/978-3-030-65390-3_45

the deceased himself is called suicide [2]. When suicide is taken as the research object of sociology, the factors leading to suicide are very complex, rather than the factors that can be analyzed in detail for a suicide case. For example, suicide may be caused by many factors, such as psychological problems, family, social life, interpersonal relationship, marriage and so on. Because suicide factors are very many and uncertain factors are very complicated, it is impossible to calculate the exact accurate value through mathematical quantitative calculation method [3]. In this paper, we need to use big data processing technology and establish fuzzy mathematics and other methods to study In order to reduce the incidence of suicide crisis in society, preventive intervention should be carried out for the groups with suicide crisis.

This paper makes the following contributions:

The data for this suicide preventive intervention comes from the kaggle data platform named Suicides in India 2001–2012 [4]. The data set has a total of 237,520 data. The main attributes of the data include the location (State), year (Year), type (Type_code), factor (Type) and many more.

(1) By analyzing the relationship between data attributes and establishing a fuzzy mathematical suicide risk evaluation model to analyze the relationship between suicide phenomena in society and many factors, combined with mathematical model design algorithms, and finally get the weight of suicide factors. Use different weight combinations of suicide factors to calculate the size of suicide risk.
(2) Combined with the factors caused by the above suicide problems, the questionnaire system is designed to intervene the members of the society through the system, so as to reduce the suicide rate in the society.

2 Related Work

2.1 Fuzzy Mathematics [5]

Fuzzy mathematics is to use mathematical methods to study fuzzy phenomena, taking uncertain things as its research objects. The emergence of fuzzy sets is the need for mathematics to adapt to the description of complex things. Its merit is to use the theory of fuzzy sets to find and confirm the solution of fuzzy objects, so that the mathematics of studying certain objects can be communicated with the mathematics of uncertain objects. In the past, precise mathematics and random mathematical descriptions were inadequate. Indeed, studying individual suicide is very difficult and difficult to control, so the comprehensive evaluation idea of fuzzy mathematics is introduced into suicide research to establish a suicide risk assessment model.

2.2 K-Mean Algorithm

The k-means clustering algorithm is an iterative clustering analysis algorithm. Its steps are to randomly select K objects as the initial cluster centers, and then calculate the difference between each object and each seed cluster center. The distance between each object is assigned to the nearest cluster center. K-mean algorithm is mainly used to calculate the boundary value of fuzzy mathematics membership function in this research. The main purpose is to classify data with similar causes of suicide.

3 Data Preprocessing

The process of data preprocessing is mainly to remove and merge the insignificant data and repeated data, and then visually analyze the data attributes after the data cleaning, so as to prepare for the future model analysis and establishment, and reduce the complexity of the algorithm to a certain extent.

3.1 Data Cleaning

After analyzing the relevant attributes of the data set, it is necessary to clean up the data that is not significant for suicide intervention, such as the data elimination of unknown suicide factors and the data cleaning of suicide mode in killing factors, and merge similar suicide factors.

Remove Missing and Unknown Suicide Factors
By analyzing the data set type (suicide factors), some data are not clear about the suicide factors, such as other causes, causes not known, and so on. Such data will produce certain noise to the study of suicide factors [6]. Therefore, such data should be eliminated in the process of data cleaning, so as to reduce the error of suicide factor analysis.

Remove the Data Segment of Suicide Mode Excluding Suicide Factors
Because the suicide factors in the data set contain a small number of suicide methods, such as by diving, by firearms, vehicles/trains, by jumping from building, etc. The suicide crisis in this study mainly analyzes the basic causes of the suicide crisis, as for which way the suicides should choose is of little significance to this study, so the data of suicide methods in the data set are removed.

Combining Similar Suicide Factors
By analyzing the set of suicide factors in the data set, we can find that some suicide causes are very similar, such as not having children bareness/impostancy and not having children in the data set The factors of barrenness/impostancy are very similar, so we can merge similar suicide factors to reduce unnecessary analysis of suicide factors and reduce the computational complexity and complexity of the model.

3.2 Data Analyzes

The following mainly visualizes the relationship between the year, age, reason, region and the number of suicides, and analyzes the relationship between each attribute and the number of suicides.

Figure 1 shows the total number of suicides per year from 2001 to 2012. It can be seen from the chart that the overall change in the number of suicides from 2001 to 2006 was relatively stable. From 2007 to 2010, the total number of suicides increased rapidly, and continued to decline after 2010. From the above analysis, we can see that the total number of suicides in China has a very close relationship with time, which may be related to the level of social and economic development or natural disasters [7] every

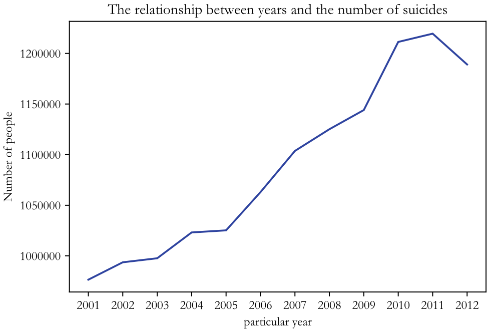


Fig. 1. The curve of total number of suicides per year

year. Therefore, in the future modeling process, it is necessary to use and analyze the year attribute reasonably.

Figure 2 shows the relationship between the total number of suicides and different age groups from 2001 to 2012. It can be seen from the figure that the number of suicides aged 0–14 is the lowest. The number of suicides over the age of 60 is also relatively low. This may be due to less social pressure on the elderly and children. The number of suicides aged 15–29 is the largest, the number of suicides aged 30–44 is second only to 15–29, and the number of suicides in 45–59 age group is close to half of that in 15–29 age group. That is to say, young people are the main suicide group in society, which may be caused by family pressure and social pressure [8].

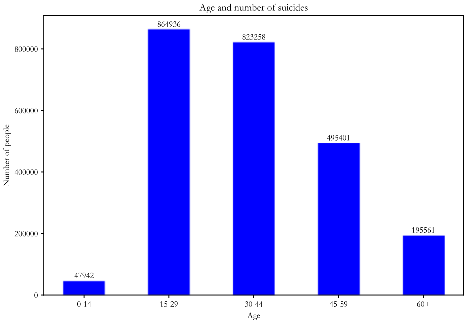


Fig. 2. The relationship between the age and the total number of suicides

Table 1 shows the relationship between suicide factors and the corresponding total number of suicides from 2001 to 2012. From the data in the table, it can be seen that the number of suicides by housewives is the largest, followed by the number of children killed by the unemployed. It can be seen that most suicide crisis events come from families and society, but there are also some psychological factors, such as mental the number of suicides in illness was 94 229. There are also a small number of people who commit suicide because they can’t get children or their ideology.

Table 1. The relationship between suicide factors and the number of suicides.

Factor	Number of people	Factor	Number of people
House Wife	341952	Not having children	766
Unemployed	114374	Ideological causes	2118
Mental Illness	94229	Illegitimate pregnancy	2494
Business activity	78112	Bankruptcy	2655
Student	74323	Physical abuse	3992
Love affairs	45039	Divorce	4133

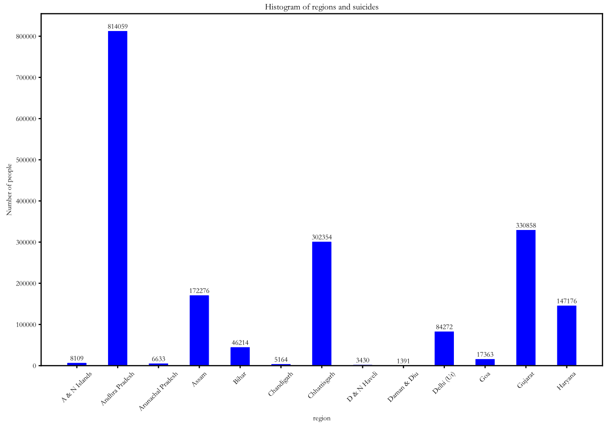


Fig. 3. The total number of suicides in each region

Figure 3 above shows the total number of suicides in each region in 2001 and 2012. It can be seen from the table that the difference in the number of suicides in different regions is also very large. The number of suicides in Nagaland state is 502, and that in Pradesh state is 205535. The above reasons may be that the number of people and the level of economic development of each state are different.

4 Model Analysis and Establishment

When building a model, we must first divide the data set, mainly based on factors such as age, gender, etc., to obtain data for different research subjects, and then find out related suicide factor sets, and at the same time, we need to establish a related comprehensive evaluation set. Combined with the K-mean algorithm to divide the boundary value of the risk degree of the factor set. Before the establishment of the fuzzy mathematical model, it is necessary to select the membership function that is most suitable for the suicide risk law, and combine the membership function to calculate the single-factor fuzzy evaluation matrix. At the same time, the single-factor fuzzy evaluation matrix value is normalized, and finally obtained by weighting The relevant risk level results.

The meaning of the specific symbols used in the paper is shown in Table 2.

Table 2. The description of the symbols in the paper

Symbol	Description
A_i	Represents the i-th research object
U	Represents the set of suicide factors
V	Represents a set of suicide evaluations
R	Express fuzzy evaluation matrix
B_i	Represents the i-th comprehensive risk evaluation vector
w_{ij}	Represents the risk value of the j-th suicide factor of the i-th research object
W_i	Represents the index value of the i-th research object
S	Represents the comprehensive value of risk degree of the research object

4.1 Model Data Set Classification

From the Sect. 3.2 data analysis, we can see that age, gender, reason and other factors have a strong [9, 10] correlation with the number of suicides. Since the reasons for suicide vary greatly under different genders and ages, in order to establish a reasonable and scientific model, the three attributes of age, gender and type are combined here. The combination number of existing data set a can be expressed as follows:

$$A = C_2^1 \times C_2^1 \times C_5^1 + C_2^1 \times C_2^1 \times C_1^1 \tag{1}$$

combinations for research, and the 24 different combinations are represented by the symbol E. Therefore, E_i represents the i-th research object of the study. The following

Table 3. Partial data of the subjects.

E_i		
Type_code	Gender	Age_group
Causes	Female	0–14
Causes	Female	15–29
Causes	Male	15–29
Education_Status	Female	0–100+
Professional_Profile	Male	30–44
Social_Status	Female	0–100+
…		

model establishment and analysis are based on each research object E_i (as shown in Table 3).

4.2 Determination of Fuzzy Evaluation Factor U

Before the establishment of the suicide fuzzy mathematical model, it is necessary to determine the fuzzy factor set. In this study, the fuzzy factor set mainly refers to the suicide reason of the suicide. Because the suicide reason of the suicide person is usually fuzzy, it is very important to have a reasonable and scientific evaluation factor set for the establishment of the model. In this study, the evaluation factors are composed of suicide factors, namely, the data attribute (type) field value after data cleaning. The evaluation set U of suicide factors can be recorded as:

$$U = \{u_1, u_2, u_3 \ldots u_m\} \tag{2}$$

Among them, $u_i(i = 1 \ldots 54)$ represents 54 suicide factors in the data set. The following is to make relevant treatment for the fuzziness of evaluation factors, that is, to design relevant membership functions.

4.3 Establishment of Comprehensive Evaluation Set

The evaluation set of this study is determined according to the degree of suicide risk. The suicide risk degree is divided into five levels, namely, very safe, safe, normal, dangerous and very dangerous. The risk assessment set V can be expressed as follows:

$$V = \{v_1, v_2, v_3, v_4, v_5\} \tag{3}$$

Among them, $v_i(i = 1 \ldots 5)$ stands for five risk levels. Very safe means that the suicide risk of personnel is close to 0 under certain factors, while very dangerous is that the suicide risk of personnel is very high under certain factors. In dangerous and very dangerous groups, suicidal phenomenon may occur at any time. If such people are found, they should be given mental counseling in time, and relevant suicide intervention should be carried out at the same time, so as to reduce the suicide rate of society and avoid huge losses to society and family.

For example, for multiple subjects, the suicide rate of adolescents aged 0–14 years was very low. If we use a clear average risk to divide, there will be many high-risk teenagers to be ignored, but for the 15–29-year-old, his suicide rate is the highest. If a clear mean risk is used, a large number of groups cannot be predicted. Therefore, for different research objects, it is necessary to determine different risk threshold, and separate people of different ages and genders to improve the accuracy of the mathematical model. Next, k-mean algorithm is used to divide the risk degree, and the process is introduced in detail.

4.4 Partition of K-Mean Boundary Value

The main steps of K-means algorithm are as follows: ① select k initial clustering centers. ② the distance between each object and the center of k is calculated, and the nearest cluster is assigned according to the principle of minimum distance. ③ the sample mean in each cluster is used as the new cluster center. ④ Repeat steps 2 and 3 until the cluster center does not change. ⑤ finally, k clusters are obtained.

According to the basic realization process of boundary value division step 1. In the clustering process, the Euclidean distance calculation method is used to calculate the distance from each point to the cluster center for 54 factor sets in each research object:

$$L_k = \sum_{i=1}^{48} (u_i - u_k)^2 \quad (k = 1..5) \tag{4}$$

Among them, u_i represents the number of suicides in the i-th factor of the total factors in the research object, and represents five cluster centers. The following is a brief introduction to the basic ideas of clustering process.

The value of k is mainly designed according to five comprehensive evaluation sets. The result of clustering is to select five centroids of cluster as the boundary value of division. The higher the cluster centroid value, the higher the risk level. Conversely, if the cluster centroid is small, the corresponding suicide risk will be lower. The calculation expression of centroid b is as follows:

$$b_i = \frac{1}{n}\sqrt{\sum_{j=1}^{n} (u_j - u_i)^2} \quad (i = 1..5) \tag{5}$$

The representative is the i-th cluster center. Through the above design, we can get the boundary value of different risk degree under each research object. The boundary value is mainly used for the calculation of the following membership functions (Table 4).

Table 4. The relationship between and.

	v_1	v_2	v_3	v_4	v_5
E_1	65	658	847	1322	2220
E_2	1677	9439	10898	20641	60299
E_3	0.75	7767	11009	20879	44238
E_4	381	1270	7197	14470	19924
E_5	32	280	540	5004	9369
...					

4.5 Establishment Model of Membership Function

Membership function is to calculate the membership degree of a single factor in each evaluation set through the relationship expression of function. Membership function mainly aims at the knowledge of probability, such as trapezoidal distribution, Gaussian distribution and triangular distribution. The establishment of membership function plays a very important role in fuzzy mathematical model. The following will analyze the establishment process of membership function model.

Establishment of Membership Function of Comprehensive [11, 12] Evaluation Set For the general situation, that is, $v_2,\ v_3,\ v_4$ the following membership interval distribution function can be established through the idea of triangular distribution to obtain the membership degree of single factor on a certain risk index.

$$u(x) = \begin{cases} 0, & x < a \; or \; x > c \\ \dfrac{x-a}{b-a}, & a < x \le b \\ \dfrac{c-x}{c-b}, & b < x < c \end{cases} \tag{6}$$

Among them, a, b and c are the boundary values of the membership function. Because their size relationship is a < b < c, and the number of suicides is not less than zero, the value of a is zero. The boundary value c is the maximum value of the independent variable of the membership function, so c is the maximum number of suicides in each research object. The value of b is replaced by the cluster centroid introduced in Sect. 4.4. It is easy to know that each research object has a fixed $b_i(i = 1..5)$ and v_i correspondence, b_i value is the centroid of each cluster, that is, b_i size is the average value of distance in each cluster.

Figure 4 shows the curve of membership function under the determined boundary value. It can be seen from the graph that the vertex is the place with the highest degree of membership. From the function curve of the above figure, it is a membership distribution completely in line with the actual situation. When x is equal to or close to b, the value of the membership function $u(x)$ is also close to 1, that is to say, x in the factor set completely conforms to the classification index b, the risk index can be set as b (Fig. 5).

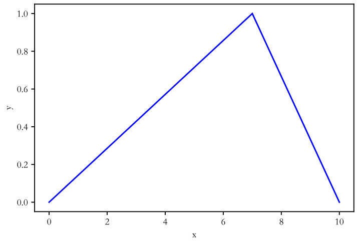


Fig. 4. The curve of membership function under the determined boundary value

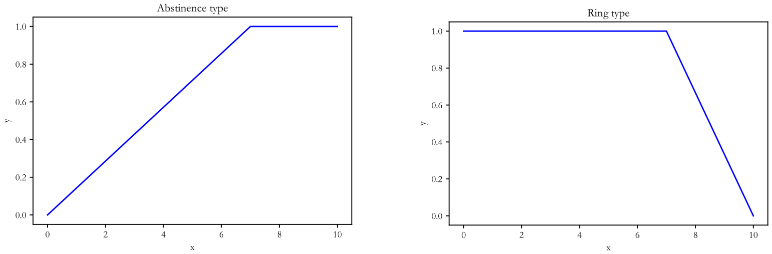


Fig. 5. Membership curve in two special cases

The above figure shows the ring up type and the ring down type of membership function, which belong to two forms of semi trapezoidal distribution membership function. They are used to evaluate the risk indicators of v_5 and v_1. For the ring type, when $x > b$, the attribute set x is larger than the cluster centroid, obviously the membership degree $u(x) > u(b)$. From the graph, when x is greater than b value, the evaluation set of attribute set can be judged as very dangerous. For the abstaining type, when x is less than b, there is $u(x) < u(b)$, so if the attribute set is less than b, the attribute set x can be classified as a very safe evaluation set.

4.6 Determination of Single Factor Fuzzy Evaluation Matrix R

Combining expression 5 and expression 6, the single factor fuzzy evaluation matrix R can be calculated, and the matrix element r_{ij} represents the values of the ith single factor and the j-th evaluation set:

$$R = \begin{vmatrix} r_{11} \; r_{12} \; r_{13}..r_{1m} \\ r_{21} \; r_{22} \; r_{23}..r_{2m} \\ r_{31} \; r_{32} \; r_{33}..r_{3m} \\ r_{n1} \; r_{n2} \; r_{n3}..r_{nm} \end{vmatrix} \tag{7}$$

1) Single factor risk value normalization

The setting of single factor risk value is the number of suicides under various suicide factors in each research object. Since the number of suicides is more, the risk value is higher. Because the risk value of each research object is relative, because the number of suicides in each research object is different, for example, the number of suicides of teenagers and children is much lower than that of young people, so the risk value is used to calculate the risk value It is more intuitive to see the degree of suicide risk in all kinds of research objects, and it is also ready for the calculation of comprehensive risk assessment value in the future.

$$A_i = \frac{x_{ij}}{\sum\limits_{j=1}^{48} x_{ij}} \quad (i = 1..24,\ j = 1..5) \tag{8}$$

Where x_{ij} is the risk value of the jth attribute of the ith research object, and A_i is the risk normalization vector of the i-th research object (Table 5).

Table 5. Partial data of single factor fuzzy evaluation matrix

	v_1	v_2	v_3	v_4	v_5
A_1	0.794958	0.079752	0.061956	0.039695	0.023638
A_2	0.765492	0.094561	0.081902	0.043242	0.014802
A_3	0.667206	0.147706	0.104208	0.054947	0.025933
A_4	0.480398	0.391218	0.069076	0.034357	0.024952
A_5	0.373482	0.383662	0.208345	0.022496	0.012015
A_6	0.805927	0.097229	0.050601	0.024108	0.022135
A_7	0.384222	0.354638	0.150107	0.076864	0.034169
A_8	0.27656	0.29801	0.266967	0.106561	0.051893
…					

2) Calculation of comprehensive risk degree of each factor in each research object

The comprehensive risk degree of each factor of the research object refers to the sum of the product of the factors under the research object in each risk evaluation level and the assigned weight. Here, the weight y of the five risk levels can be specified as follows:

$$y = \{1,\ 2,\ 3,\ 4,\ 5\} \tag{9}$$

Where the five indicators correspond to five comprehensive evaluation sets. The lower the weight, the smaller the risk index. For example, the very safe weight in the evaluation set is 1, and the very dangerous weight is 5. Therefore, the comprehensive risk degree of each factor in each object can be calculated according to the corresponding factor evaluation vector in the single factor risk evaluation index, combining the single factor comprehensive evaluation vector 1–3 and the weight y. The following expression for calculating the comprehensive risk degree of single factor can be obtained:

$$w_i = \sum_{j=1}^{5} (r_{ij} \times y_j) \quad (i = 1..48) \tag{10}$$

Among them, w_i is the comprehensive risk degree of the i-th factor in an object, and the value of r_{ij} is the j-th risk index value of the i-th factor.

3) Risk level of each research object

According to the comprehensive evaluation A_i of each factor in the I research object and the corresponding comprehensive evaluation fuzzy matrix R, the following comprehensive evaluation index B_i can be obtained by combining expression 6 and expression 7:

$$B_i = A_i \otimes R = A_i \otimes \begin{vmatrix} r_{11}\ r_{12}\ r_{13}..r_{1m} \\ r_{21}\ r_{22}\ r_{23}..r_{2m} \\ r_{31}\ r_{32}\ r_{33}..r_{3m} \\ r_{n1}\ r_{n2}\ r_{n3}..r_{nm} \end{vmatrix} = \{b1,\ b2,\ b3,\ b4,\ b5\} \tag{11}$$

In this paper, it is necessary to normalize the comprehensive evaluation set to facilitate the calculation of the comprehensive risk degree of the following research objects:

$$B_i = \frac{b_j}{\sum\limits_{j=1}^{5} b_j} (i = 1..24) \tag{12}$$

According to the normalized results of the total comprehensive risk assessment and the weight y, the following expression of the risk degree (W_i) of the i-th research object can be obtained:

$$W_i = \sum_{j=1}^{5} (B_{ij}^{'} \times y_j) \qquad (i = 1..48) \tag{13}$$

4) Establishment of comprehensive evaluation model of suicide

The comprehensive evaluation of suicide risk combines the risk value of suicide factors and the risk degree of each research object. The variable q_{ii} is introduced here. When the value of q_{ii} is 1, it means the attribute of the i-th research object, otherwise, the value of q_{ii} is 0. Therefore, the risk value Q_i of suicide risk factor set in each research object can be expressed as follows:

$$Q_i = \sum_{j=1}^{48} w_{ij} \times q_{ij} \quad (i = 1..24,\ q_{ij} \in \{0, 1\}) \tag{14}$$

The comprehensive risk index of suicide is the product of the suicide risk value of the research object and the risk value of the suicide factors in the object:

$$S_i = W_i \times Q_i \quad (i = 1..24) \tag{15}$$

5 Model Evaluation

The evaluation model of suicide risk score established by fuzzy mathematical analysis method can obtain the comprehensive risk degree of 24 research objects. In order to verify the accuracy of the model, the judgment matrix of 24 categories is constructed by using the judgment matrix of analytic hierarchy process, and the CR value is calculated to determine the validity of the judgment matrix, so as to achieve the purpose of testing the validity of the model.

5.1 Rules for Evaluating Judgment Matrix

The rules for evaluating the judgment matrix are as follows:

1 means that two elements have the same importance;
3 means that the former is slightly important than the latter;
5 means that the former is more important than the latter;
7 means that the former is strongly more important than the latter;
9 means that the former is extremely important than the latter;
2, 4, 6 and 8 means that the middle value of two adjacent judgments.

The detail proportional information is list in Table 6 and the Ri value reference can be seen in Table 7.

Table 6. Proportional scale

Factor i vs. factor j	Quantized value
Equally important	1
Slightly important	3
Stronger important	5
Strongly important	7
Extremely important	9
The middle value of two adjacent judgments	2, 4, 6, 8

Table 7. Reference table for the value of the average random consistency index Ri in the AHP

Order	1	2	3	4	…	21	22	23	24
Ri	0	0	0.52	0.89	…	1.6385	1.6403	1.6462	1.6497

According to the comprehensive risk degree of 24 categories, it is found that the comprehensive degree is distributed between −1.05–1.05. The judgment matrix is constructed by subtracting each category by two and constructing the judgment matrix according to the difference result grade.

5.2 Construct Judgment Matrix

For 24 Categories of factors, we can construct the following judgment matrix. For convenience, each category of reasons is used C_1 to C_{24} express the specific values. The following Table 8 is the obtained judgment matrix.

$$\bar{\omega}_i = \sqrt[n]{m_i} \tag{16}$$

$$\omega_i = \frac{\bar{\omega}_i}{\sum_{j=1}^{n} \omega_j} \tag{17}$$

Table 8. The judgment matrix obtained

C	C_1	C_2	…	C_{24}	$\bar{\omega}_i$	ω_i
C_1	1	5	…	9	1.2999	0.0541
C_2	1/5	1	…	5	0.3348	0.0140
…	…	…	…	…	…	…
C_{24}	1/9	1/5	…	1	0.1329	0.0055

According to the calculation formula (16) and (17), the weight set of C1–C6 is $B = [0.0541,\ 0.0140,\ 0.0164,\ 0.0366,\ \ldots,\ 0.0621,\ 0.0621,\ 0.0621,\ 0.0055,\ 0.0055]$ The next step is to check the consistency

$$\lambda\max = \frac{1}{24}\sum_{i=1}^{24}(AW)/\omega_i = 24.8605 \tag{18}$$

Where AW is

$$AW = \begin{bmatrix} 1 & 5 & \ldots 9 \\ 1/5 & 1 & \ldots 5 \\ \ldots & \ldots & \ldots\ldots \\ 1/9 & 1/5 & \ldots 1 \end{bmatrix} \begin{bmatrix} 1.2999 \\ 0.3348 \\ \ldots \\ 0.1329 \end{bmatrix} \begin{bmatrix} 0.0541 \\ 0.0140 \\ \ldots \\ 0.0055 \end{bmatrix}$$

$$CI = \frac{\lambda\max - n}{n-1} = \frac{24.8605 - 24}{24 - 1} = 0.0374 \tag{19}$$

$$CR = \frac{CI}{RI} = \frac{0.0374}{1.6497} = 0.0227 \leq 0.1 \tag{20}$$

Through the consistency test, the comparison matrix is effective.

6 Conclusions

This study mainly conducted detailed analysis of various suicide factors of different age groups and different genders and related suicide risk evaluation models. The related theories are derived from the fuzzy mathematical theory in social science research. First, conduct suicide factor analysis by collecting a large amount of suicide data, establish a suicide risk evaluation model, and use the analytic hierarchy process to test the accuracy of the model. Later, you can use the research conclusions of this paper to design relevant questionnaire interactive websites. Quickly understand the degree of personal suicide risk.

The main shortcoming of this study is that the data comes from India. According to the research on suicide in sociology, some of the reasons for suicide are closely related to society, such as social factors such as social economic development, religious beliefs, customs and culture, education level, etc. The ideas and methods involved in this research have great practical value and a certain degree of transplantation. It can make a more accurate qualitative and quantitative judgment for some multi-factor and vague evaluation subjects, such as judging the value of a person High and low, a user's satisfaction evaluation and other related similar questions.

Acknowledgement. This work was supported by the Foshan Self-financing Project of Foshan Science and Technology Bureau (1920001000636) and the Young Innovative Talents Project of Guangdong Province Ordinary colleges and universities (2019KQNCX230), (2019KQNCX229).

References

1. https://www.who.int/health-topics/suicide#tab=tab_1
2. Wei, N.: On Durkheim's sociological research methods——from the perspective of "suicide theory". J. Shanxi Youth Manag. Coll. **25**(03), 83–85 (2012)
3. Yang, X.H., Gu, J.: Application of fuzzy mathematics in sociological research. Sociol. Res. (01), 76–87 (2000)
4. https://www.kaggle.com/rajanand/suicides-in-india
5. Jun, X.: Research on the quality evaluation method of university innovation and entrepreneurship education based on fuzzy mathematics. China Bus. Theory **18**, 196–198 (2020)
6. Picard, E.H., Rosenfeld, B.: How clinicians incorporate suicide risk factors into suicide risk assessment. Crisis (2020)
7. Boggs, J.M., Beck, A., Ritzwoller, D.P., Battaglia, C., Anderson, H.D., Lindrooth, R.C.: A quasi-experimental analysis of lethal means assessment and risk for subsequent suicide attempts and deaths. J. Gen. Intern. Med. **35**(6), 1709–1714 (2020). https://doi.org/10.1007/s11606-020-05641-4
8. Ji, Y.S.D., Robertson, F., Patel, N.A., Peacock, Z.: Assessment of risk factors for suicide among US health care professionals. JAMA Surg. **155**, 713–721 (2020). https://doi.org/10.1001/jamasurg.1338
9. Jiang, P.R.: Several typical fuzzy distributions and their calculations. J. Shenyang Inst. Mech. Electr. Eng. **04**, 77–92 (1984)
10. Cwik, M.F., O'Keefe, V.M., Haroz, E.E.: Suicide in the pediatric population: screening, risk assessment and treatment. Int. Rev. Psychiatry **9**, 1–11 (2020)
11. Several typical fuzzy distributions and their calculation. Shenyang Inst. Mech. Electr. Eng. (1984)
12. Min, L., Shouyu, C.: Comparison of relative membership function considering interval value and traditional fuzzy distribution function. Math. Pract. Knowl. **43**(10), 201–205 (2013)

Detecting Topic and Sentiment Dynamics Due to COVID-19 Pandemic Using Social Media

Hui Yin[1], Shuiqiao Yang[2], and Jianxin Li[1(✉)]

[1] School of Information Technology, Deakin University, Geelong, Australia
jianxin.li@deakin.edu.au

[2] Data Science Institute, University of Technology Sydney, Sydeny, Australia

Abstract. The outbreak of the novel Coronavirus Disease (COVID-19) has greatly influenced people's daily lives across the globe. Emergent measures and policies (e.g., lockdown, social distancing) have been taken by governments to combat this highly infectious disease. However, people's mental health is also at risk due to the long-time strict social isolation rules. Hence, monitoring people's mental health across various events and topics will be extremely necessary for policy makers to make the appropriate decisions. On the other hand, social media have been widely used as an outlet for people to publish and share their personal opinions and feelings. The large scale social media posts (e.g., tweets) provide an ideal data source to infer the mental health for people during this pandemic period. In this work, we propose a novel framework to analyze the topic and sentiment dynamics due to COVID-19 from the massive social media posts. Based on a collection of 13 million tweets related to COVID-19 over two weeks, we found that the positive sentiment shows higher ratio than the negative sentiment during the study period. When zooming into the topic-level analysis, we find that different aspects of COVID-19 have been constantly discussed and show comparable sentiment polarities. Some topics like "stay safe home" are dominated with positive sentiment. The others such as "people death" are consistently showing negative sentiment. Overall, the proposed framework shows insightful findings based on the analysis of the topic-level sentiment dynamics.

Keywords: COVID-19 · Topic tracking · Sentiment analysis · Twitter

1 Introduction

The outbreak of the novel Coronavirus Disease 2019 (COVID-19) has influenced millions of people around the world [24]. It seriously threatens peoples lives due to its highly contagious from person to person. According to the reports, the globally confirmed cases of COVID-19 have surpassed 10 milloin and more than 500,000 people have lost their lives due to the infection until 29 June 2020[1]. Many

[1] https://news.google.com/covid19/map?hl=en-AU&gl=AU&ceid=AU%3Aen.

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 610–623, 2020.
https://doi.org/10.1007/978-3-030-65390-3_46

countries and regions have adopted a series of specific measures to help slow the spread of COVID-19, such as travel ban, lockdown, closure of public places (e.g., gym, restaurants, schools), requiring people to practice good personal hygiene, keep physical social distance of 1.5 m, and work or study at home to reduce contact with others. The above measures have changed people's daily lives, and people are required to follow policies to protect the safety of their communities. However, many mental symptoms like worry, fear, frustration, depression and anxiety could occur and cause serious mental heath issues to people due to the long-time social activity restriction during the pandemic period [28]. Therefore, understanding when and where people would experience mental issues in this special period is important for governments to make the right decisions.

Social media data has always been highly concerned by researchers [1,15,16, 22,30]. During this difficult period, people could spend more time on social media platforms like Twitter to get the latest news, communicate with friends and post their feelings and thoughts during the lockdown period. Such massive personal posts from social media can be a valuable data source for large-scale sentiment and topic mining for monitoring people's mental health across different events or topics [25]. Therefore, many recent work has focused on discovering peoples social sentiment due to COVID-19 [11,18,28,29]. For instance, Zhou et al. [28] adopted Twitter data for massive sentiment analysis due to COVID-19 for people living in New South Wales, Australia. Han et al. [11] studied the public opinion in the early stages of COVID-19 in China by analyzing text data from Sina Weibo. Rajesh et al. [18] exploited topic model to generate topics from tweets related to coronavirus and calculated the presence of different emotions. However, little work is found to analyze the topic and sentiment dynamics together, even though such dynamical analysis is important for authorities to understand when and where people would experience mental health issues.

In this work, we aim to dynamically identify the popular topics and their associated sentiment polarities due to the COVID-19 pandemic. Our research questions are: (1) How do people's emotions change? (2) What topics are people discussing? (3) How has the different topics evolved? (4) How is the sentiment dynamics of topics? To answer these questions, we propose a novel dynamic topic discovery and sentiment analysis framework which contains multiple modules include data crawling, data cleaning, topic discovery, sentiment analysis and result visualization. In the proposed framework, we employ the Dynamic Topic Model (DTM) [3] to generate accurate daily topics. To determine the sentiment polarity of each topic and tweet, we utilize a sentiment lexicon tool: VADER [13][2] to infer the sentiment polarity. We collect 13,746,822 tweets from 1 April 2020 to 14 April 2020 related to COVID-19 from Twitter across the world to test the effectiveness of the proposed framework. The experimental results show that the proposed framework can generate insightful findings such as the overall sentiment dynamics among people, topic evolutionary patterns, sentiment dynamics of different topics.

[2] https://github.com/cjhutto/vaderSentiment.

2 Related Work

With the spreading of COVID-19 across the world, researchers have proposed to use sentiment analysis based on social media as a tool to monitor people's mental health. In this section, we review the latest work related to COVID-19 analysis based on social media data.

Rajesh et al. [18] adopted a classic Latent Dirichlet Allocation (LDA) topic model method to generate 10 topics in a random sample of 18,000 tweets about coronavirus, then they used NRC sentiment dictionary to calculate the presence of eight different emotions, which were “anger”, “anticipation”, “disgust”, “fear”, “joy”, “sadness”, “surprise” and “trust”. Han et al. [11] explored public opinion in the early stages of COVID-19 in China by analyzing Sina-Weibo texts. The COVID-19 related microblogs were summarized into 7 topics and 13 more detailed sub-topics. However, they judged the polarity of the topics according to the polarity of the topic words and failed to consider the content of posts under this topic. Cinelli et al. [7] analyzed engagement and interest in the COVID-19 topic on different social media platforms such as Twitter, Instagram, Reddit, and provided a di?erential assessment of the global discourse evolution of each platform and their users. They found that reliable and suspicious information sources have similar spreading patterns. Depoux et al. [9] confirmed that the spread of social media panic is faster than that of COVID-19. Therefore, the public rumors, opinions, attitudes and behaviors surrounding COVID-19 need to be quickly detected and responded to. They suggested to create an interactive platform dashboard to provide real-time alerts of rumors and concerns about the spread of coronavirus worldwide, which would enable public health officials and relevant stakeholders to respond quickly with a proactive and engaging narrative, thereby reducing the impact of misinformation. Sharma et al. [21] designed a dashboard to track misinformation in Twitter conversations. The dashboard allows visualization of the social media discussions around coronavirus and the quality of information shared on the platform and updated over time. They evaluated sentiment polarity for each tweet based on its textual information and showed the distribution of sentiment in different countries over time.

More recently, Huang et al. [12] examined the public discussion about COVID-19 on Twitter and found that the most influential tweets are still written by regular users, such as news media, government officials, and individual news reporters. They also discovered that “fake news” sites are most likely to be retweeted within the source country and so are less likely to spread internationally. Zhou et al. [28] exploited tweets on Twitter to analyse the sentiment dynamics of people living in the state of New South Wales (NSW) in Australia during the pandemic period. They first summarized that the overall polarity of the community since the outbreak was positive, and then analyzed the sentiment dynamics of the NSW local government areas in terms of lockdown, social distance and JobKeeper.

Different from the above work that either performed static sentimental analysis or failed to analysis the detailed topic-level sentiment. We propose to analyse the dynamics of topic-level sentiment to monitor the evolution of people's mental states across different topics or events.

3 Proposed Framework

In this section, we introduce the proposed framework and the adopted techniques for detecting topic and sentiment dynamics. Our framework contains three steps for the task. Firstly, we divide the tweets into different topics generated by a dynamic topic model, then we determine the sentiment polarity of each tweet, and finally we summarize the sentiment polarity distribution of each topic. The proposed framework is shown in Fig. 1.

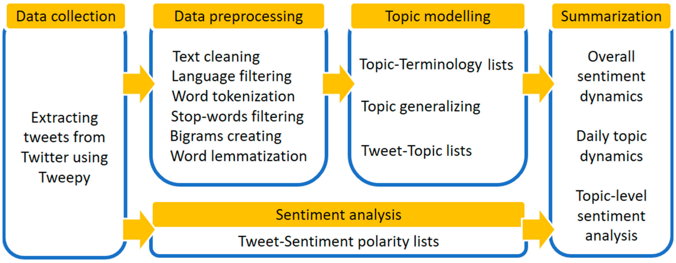


Fig. 1. Overall framework.

3.1 Topic Extraction

Topic modelling is the process of learning, recognizing, and extracting high-level semantic topics across a corpus of unstructured text. A popular method for this is Latent Dirichlet Allocation (LDA) proposed by Blei et al. [4], which is used to detect topics for static corpus. Later, Dynamic Topic Models (DTM) [3] is proposed to mine topic evolution over time by extending the idea of LDA to allow topic mining over fixed time intervals. Hence, DTM can handle sequential corpus and generate topics for different time slices of corpus. Specifically, the documents within each time slice are modeled with the static LDA, the topics associated with slice t evolve from the topics associated from the previous time slice $t-1$. The dynamics of the topic model is given by Eq. 1 and 2, Table 1 shows the meaning of the notations. DTM mines topics of each time slice with LDA and its parameters β and α are chained together in a state space model which evolve with Gaussian noise to get a smooth evolution of topics from time to time.

$$\beta_{t,k} \mid \beta_{t-1,k} \sim N(\beta_{t-1,k}, \sigma_2 I) \tag{1}$$

$$\alpha_t \mid \alpha_{t-1} \sim N(\alpha_{t-1}, \delta_2 I) \tag{2}$$

Table 1. Meaning of the notations.

Symbol	Description
α_t	As the per-document topic distribution at time t
$\beta_{t,k}$	As the word distribution of topic k at time t
$\eta_{t,d}$	As the topic distribution for document d in time t
$z_{t,d,n}$	As the topic for the n^{th} word in document d in time t
$\omega_{t,d,n}$	As the n^{th} word at time slice t, document d

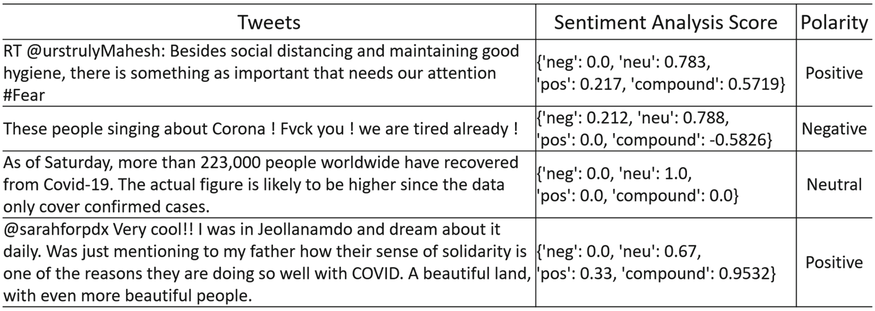
3.2 Tweets Sentiment Analysis

Sentiment Analysis (SA) also commonly referred as Opinion Mining (OM) that aims to find opinionated information and detect the sentiment polarity. Nowadays, SA techniques are quite mature and many tools are openly available, such as Stanford's CoreNLP [17], VADER [13], SentiStrength [23], SentiCircles [20], which are specifically designed for social media conversation. In this work, we employ VADER to identify the sentiment polarity of each tweet. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiment expressed in social media. It was introduced by Hutto et al. in 2014 and has been widely used in many social media text sentiment analysis tasks [6,8,10,21,27,28]. VADER can classify the sentiment into negative, neutral, and positive categories and employ compound score which is computed by summing the valence scores of each word in the lexicon and normalized in range (−1, 1), where "−1" represents most extreme negative and "1" represents most extreme positive. If compound score is greater than 0.05, the text is considered positive, if the score is less than −0.05, it is considered negative, if the score is between 0.05 and −0.05, the text polarity is neutral. The biggest advantage of VADER is that it does not require data preprocessing and model training, and can be used directly on the raw tweet to generate sentiment polarity. Some examples of tweet sentiment results from VADER are shown in Fig. 2. In this work, we use VADER to produce the sentiment polarity of each original tweet, namely positive, negative and neutral. And then, combine the topic mining result from DTM to analyze the topic-level sentiment.

3.3 Measuring Topic Sentiment

After the above two steps, all tweets will be clustered in the corresponding topics, and marked with sentiment polarity. The sentiment polarity is aggregated over tweets to estimate the overall sentiment distribution. For each topic per day, we sum up the number of positive, negative and neutral tweets in the topic, thus, each topic is associated with three sentiment counts.

$$DT_{i,j} = N_p + N_n + N_o \tag{3}$$



Tweets	Sentiment Analysis Score	Polarity
RT @urstrulyMahesh: Besides social distancing and maintaining good hygiene, there is something as important that needs our attention #Fear	{'neg': 0.0, 'neu': 0.783, 'pos': 0.217, 'compound': 0.5719}	Positive
These people singing about Corona ! Fvck you ! we are tired already !	{'neg': 0.212, 'neu': 0.788, 'pos': 0.0, 'compound': -0.5826}	Negative
As of Saturday, more than 223,000 people worldwide have recovered from Covid-19. The actual figure is likely to be higher since the data only cover confirmed cases.	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	Neutral
@sarahforpdx Very cool!! I was in Jeollanamdo and dream about it daily. Was just mentioning to my father how their sense of solidarity is one of the reasons they are doing so well with COVID. A beautiful land, with even more beautiful people.	{'neg': 0.0, 'neu': 0.67, 'pos': 0.33, 'compound': 0.9532}	Positive

Fig. 2. Example tweets with sentiment polarity inferred by VADER.

$$N = \sum_{i=1}^{|D|} \sum_{j=1}^{|T|} DT_{i,j}, \tag{4}$$

For each topic in the studied days, we define a sentiment distribution in Eq. 3 and 4. $|D|$ is the total studied days and $|T|$ is the total topic number. $DT_{i,j}$ represents the total number of tweets under topic j in day i, N_p, N_n and N_o respectively denote the positive, negative and neutral tweet counts. N represents the total number of tweets in our dataset. We observe the daily hot topics about COVID-19 on Twitter (April 1 to April 14), and analyze the sentiment polarity distribution of each topic. The details are presented in Sect. 4.3.

4 Experimental Study

In this section, we present the processes of data collection and data preprocessing, and how to determine the optimum topic number in topic modelling.

4.1 Data Collection and Preprocessing

The data collection process is described as follows. Firstly, we obtain tweet IDs from the public available coronavirus Twitter dataset[3] that collected by Chen et al. [5] from Twitter API[4] based on keywords such as Coronavirus, Covid, Covid19, Wuhanlockdown and account names such as CDCemergency, CDCgov, WHO, HHSGov to actively tracking tweets from Twitter. We collect totally 13,746,822 tweets with Tweepy (a python library for the Twitter API) based on the given tweet IDs from April 1 to April 14. After that, the non-English tweets in the tweet dataset are removed, and we get a total of 8,430,793 English tweets.

Data preprocessing or cleaning is the first step for text mining task [26]. It includes converting all letters into lowercase, removing stop words, non-English

[3] https://github.com/echen102/covid-19-tweetids.

[4] https://developer.twitter.com/en/docs/api-reference-index.

letters, URLs, etc. Then, phrase extraction is used to ensure that words such as “human_rights” could be one token instead of separating “human” and “rights”. Additionally, lemmatization is also adopted to remove inflectional endings and return the base or dictionary form of a word. After preprocessing, we remove very short tweets (less than 6 words) and retained a total of 4,919,471 tweets with 269,391 unique tokens. Figure 3 shows the ratio of raw tweets, English tweets, and finally adopted tweets. Table 2 shows the number of daily tweets used for the experiment.

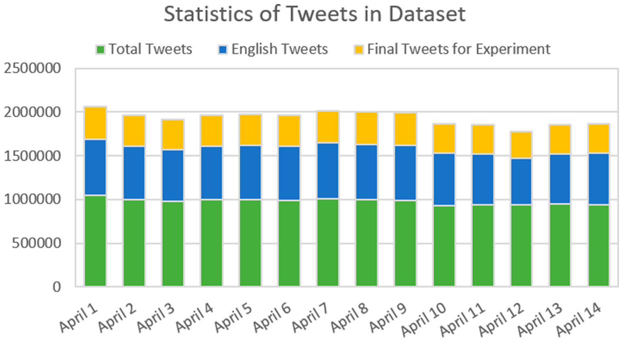


Fig. 3. Statistics of tweets in dataset. The green rectangle represents the total number of tweets per day, the blue rectangle represents the number of English tweets in all tweets, and the yellow rectangle represents the number of pre-processed tweets used in the experiment. (Color figure online)

Table 2. Size of daily tweets after preprocessing.

Date	April 1	April 2	April 3	April 4	April 5	April 6	April 7
Total	374,327	355,504	350,211	354,884	352,499	355,478	373,342
Date	April 8	April 9	April 10	April 11	April 12	April 13	April 14
Total	377,615	373,812	334,297	335,607	307,422	331,806	342,667

4.2 Topic Model Setup

The number of topics is a crucial parameter in topic modeling and capable of making these topics human interpretable. According to [3], for the first slice of Dynamic Topic Models (DTM) to get setup, we fit LDA with one day’s data (April 1) to get the best topic number. We employ Gensim package in Python to train LDA model for the selection of best topic number. Here, we use the coherence [19] by measuring the degree of semantic similarity between high scoring words of topics as an indicator to choose the best topic number. The coherence score helps distinguish between human understandable topics and artifacts of statistical inference.

$$Coherence = \sum_{i<j} score(w_i, w_j), \tag{5}$$

where select top n frequently occurring words in each topic, then aggregate all the pairwise scores of the top n words $w_i, \cdots, w_n$ of the topic.

Figure 4 shows the coherence scores of different topic numbers on the LDA model based on one day tweets. As we can see, when the topic number is 70, the coherence score gets the maximum value that is around 0.39. Therefore, we assume the total number of topics is stable and set the topic number for DTM to 70.

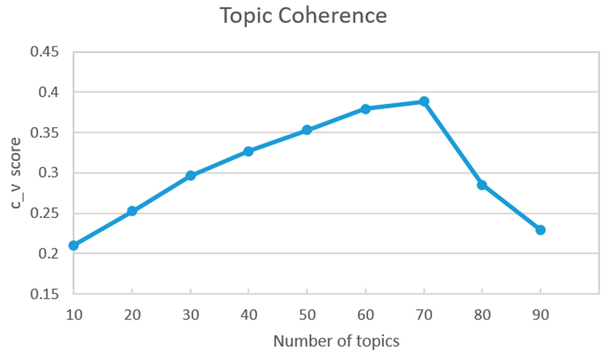


Fig. 4. Coherence score of different topic numbers on LDA.

4.3 Results

Figure 5 shows the overview of the number of deaths and confirmed cases, these datasets are collected from WHO[5]. According to reports, COVID-19 has gradually spread to all parts of the world and has aroused the attention of various countries.

Overall Sentiment Dynamics. Figure 6 presents the overall sentiment distribution on Twitter during the study period. The number of tweets about the COVID-19 is around 350,000 per day, and the daily number of positive and negative tweets is similar but all greater than the number of neutral tweets. In most days, the number of positive tweets is slightly higher than negative tweets. It shows that despite the spread of COVID-19, the community showed a dominant

[5] https://covid19.who.int/.

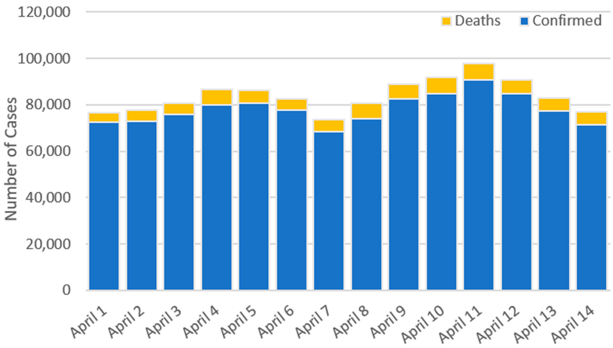


Fig. 5. Worldwide deaths and confirmed cases of COVID-19 during the study period.

positive sentiment during the study period. This observation is also consistent with the finding of other researchers who have reported the similar conclusions based on country-level sentiment analysis [2,14,28].

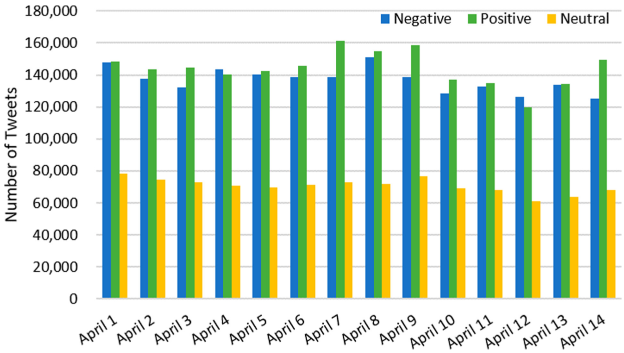
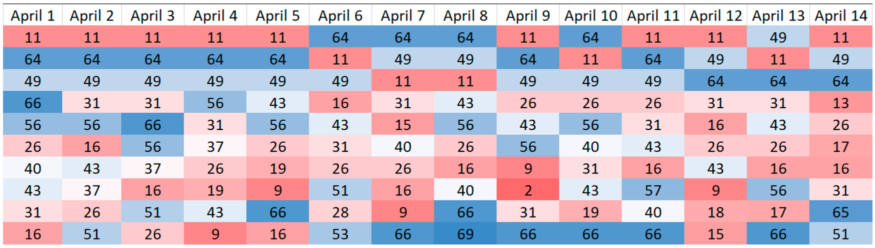


Fig. 6. The overall sentiment dynamics on Twitter during the study period.

Daily Hot Topics. For daily topics discussed by users, we use DTM to generate 70 topics and then observe the popularity of topics. Table 3 shows the extracted top 10 highest volume topics with the most relevant words associated with the topics. To find the hottest topic discussed by people in the studied period, we rank the topics by their corresponding tweets number. Figure 7 shows the index of the top 10 topics for each day, all topics are sorted in descending order. As we can see, topics 11, 49, and 64 are steadily ranked as the top 3 of daily hot topics.

Table 3. Top 10 topics about COVID-19 on Twitter during the study period. Below each topic, the most contributing words related to the topic are displayed.

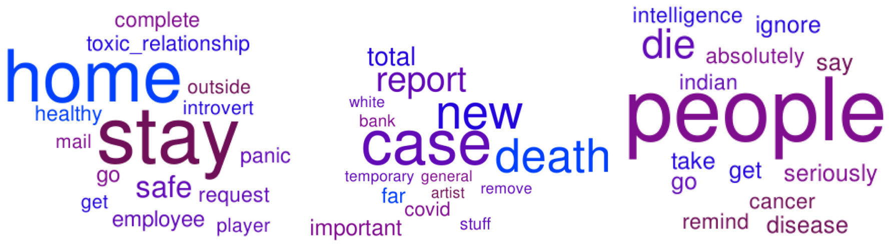
Topic 64 (disease)	Topic 49 (report)	Topic 11 (stay home)	Topic 26 (lockdown)	Topic 31 (life)
People	Case	Stay	Day	Time
Die	New	Home	Lockdown	Good
Ignore	Death	Safe	Fight	First
Seriously	Report	Go	Covid	Life
Get	Total	Employee	Road	Talk
Disease	Important	Toxic_relationship	Go	Hard
Take	Far	Healthy	Medical_supplie	Hour
Go	Covid	Request	Deliver	Go
Say	Bank	Complete	Benefit	Failure
Intelligence	Stuff	Panic	Stayhom	Pandemic
Topic 43 (work)	Topic 16 (social_distancing)	Topic 66 (stop spread)	Topic 56 (face mask)	Topic 40 (health care)
Know	Think	Virus	Make	Health
Work	Thing	Spread	Mask	Care
Medium	Month	Stop	Face	Right
Create	Next	Woman	Share	Company
Lead	Social_distancing	Leader	Wear	Risk
Little	Go	Move	Sell	Worker
Difficult	Vaccine	Citizen	Concern	Result
Street	Get	Act	Stayhome	Would
Need	Article	Deadly	Expect	Demand
Tip	Finally	Slow	Wonder	Resource



April 1	April 2	April 3	April 4	April 5	April 6	April 7	April 8	April 9	April 10	April 11	April 12	April 13	April 14
11	11	11	11	11	64	64	64	11	64	11	11	49	11
64	64	64	64	64	11	49	49	64	11	64	49	11	49
49	49	49	49	49	49	11	11	49	49	49	64	64	64
66	31	31	56	43	16	31	43	26	26	26	31	31	13
56	56	66	31	56	43	15	56	43	56	31	16	43	26
26	16	56	37	26	31	40	26	56	40	43	26	26	17
40	43	37	26	19	26	26	16	9	31	16	43	16	16
43	37	16	19	9	51	16	40	2	43	57	9	56	31
31	26	51	43	66	28	9	66	31	19	40	18	17	65
16	51	26	9	16	53	66	69	66	66	66	15	66	51

Fig. 7. The top ten topics per day. The number in the cell represents the index of the topic, and daily topics are sorted in descending order according to volume.

Figure 8 shows the most significant words that are associated with top three topics. Topics 11, 49, and 64 reflect the common concerns discussed by people, they are about staying at home to ensure safety, the latest case reports and people death due to the disease.



(a) Topic 11 (b) Topic 49 (c) Topic 64

Fig. 8. The most significant words for the three hot topics.

To further analyze these three topics, we hope to know people's opinions on these topics. We analyze the proportion of sentiment polarity of tweets under these topics to dynamically observe people's sentiment changes as the pandemic spreads, the results are shown in Figs. 9, 10 and 11. Overall, the topic's sentiment polarity various from topic to topic.

Topic 11 is related to staying at home, and our results show that the public kept a highly positive sentiment dominantly towards this measure to prevent infection. This may be because people work or study at home and enjoy more time with their families. They also had a positive belief to the combat against COVID-19 by the government and the society.

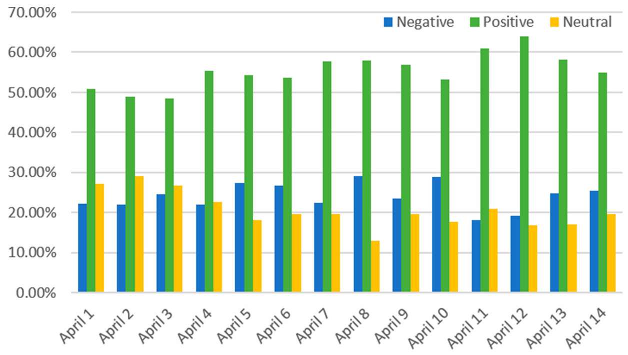


Fig. 9. The sentiment dynamics of topic 11 during the study period.

Topic 49 is about latest report about cases, the dominant sentiment around the topic of cases was almost negative despite positive sentiment existing. As the pandemic spread to more countries, the number of confirmed and deaths continues to rise, and people feel that the actual situation is worse than they expected.

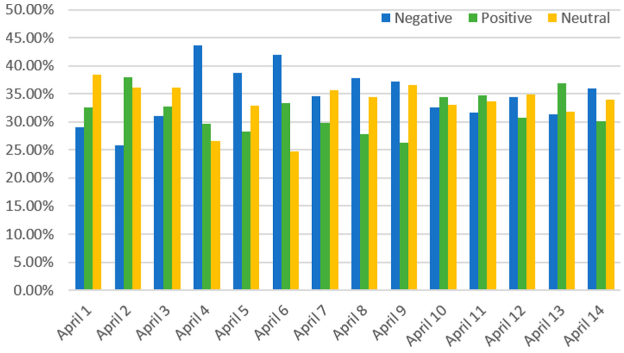


Fig. 10. The sentiment dynamics of topic 49 during the study period.

Topic 64 is about people's death due to COVID-19, it shows a diametrically opposite result to topic 11, tweets with negative emotions are much higher than others. It is believed that the outbreak cannot be effectively controlled completely since policy makers ignored the seriousness of the pandemic, so people expressed strong dissatisfaction with this consequence.

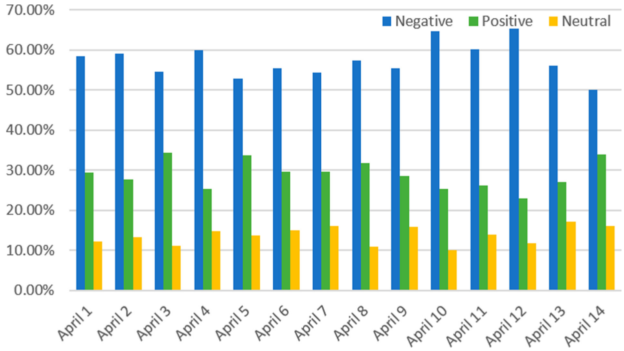


Fig. 11. The sentiment dynamics of topic 64 during the study period.

5 Conclusions and Future Work

This study conducted a comprehensive analysis about hot topics with associated sentiment polarity distribution during COVID-19 period. Instead of the country-level study, the sentiment in this work was analysed at global-level on more than 13 million tweets collected from Twitter for two weeks from 1 April 2020. The results of analysis showed that people concern about the latest coronavirus reports, measures to prevent infection, the attitudes and specific measures of governments towards the pandemic. The overall sentimental polarity was positive, but topic sentiment polarity various from topic to topic. More interesting topics can be explored based on the current study in the future. For example, more specific topics can be analyzed to help policy maker, government and local communities to formulate measures to prevent the spread of negative emotions on social network, such as food shortage, lose job, debt, study at home.

References

1. Alduaiji, N., Datta, A., Li, J.: Influence propagation model for clique-based community detection in social networks. IEEE Trans. Comput. Soc. Syst. **5**(2), 563–575 (2018)
2. Bhat, M., Qadri, M., Noor-ul Asrar Beg, M.K., Ahanger, N., Agarwal, B.: Sentiment analysis of social media response on the covid19 outbreak. Brain, Behav. Immunity **87**, 136–137 (2020)
3. Blei, D.M., Lafferty, J.D.: Dynamic topic models. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 113–120 (2006)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
5. Chen, E., Lerman, K., Ferrara, E.: COVID-19: the first public coronavirus twitter dataset. arXiv preprint arXiv:2003.07372 (2020)
6. Cheng, J., Bernstein, M., Danescu-Niculescu-Mizil, C., Leskovec, J.: Anyone can become a troll: causes of trolling behavior in online discussions. In: Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, pp. 1217–1230 (2017)
7. Cinelli, M., et al.: The COVID-19 social media infodemic. arXiv preprint arXiv:2003.05004 (2020)
8. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language. In: Eleventh International AAAI Conference on Web and Social Media (2017)
9. Depoux, A., Martin, S., Karafillakis, E., Preet, R., Wilder-Smith, A., Larson, H.: The pandemic of social media panic travels faster than the COVID-19 outbreak (2020)
10. Ferrara, E., Yang, Z.: Measuring emotional contagion in social media. PLoS ONE **10**(11), e0142390 (2015)
11. Han, X., Wang, J., Zhang, M., Wang, X.: Using social media to mine and analyze public opinion related to COVID-19 in China. Int. J. Environ. Res. Public Health **17**(8), 2788 (2020)
12. Huang, B., Carley, K.M.: Disinformation and misinformation on Twitter during the novel coronavirus outbreak. arXiv preprint arXiv:2006.04278 (2020)
13. Hutto, C.J., Gilbert, E.: Vader: a parsimonious rule-based model for sentiment analysis of social media text. In: Eighth International AAAI Conference on Weblogs and Social Media (2014)
14. Jaidka, K., Giorgi, S., Schwartz, H.A., Kern, M.L., Ungar, L.H., Eichstaedt, J.C.: Estimating geographic subjective well-being from Twitter: a comparison of dictionary and data-driven language methods. Proc. Natl. Acad. Sci. **117**(19), 10165–10171 (2020)
15. Jin, G., Yu, Z.: A Korean named entity recognition method using bi-LSTM-CRF and masked self-attention. Comput. Speech Lang. **65**, 101134 (2020)
16. Li, J., Cai, T., Deng, K., Wang, X., Sellis, T., Xia, F.: Community-diversified influence maximization in social networks. Inf. Syst. **26**, 101522 (2020)

17. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The stanford coreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55–60 (2014)
18. Prabhakar Kaila, D., Prasad, D.A., et al.: Informational flow on twitter-corona virus outbreak-topic modelling approach. Int. J. Adv. Res. Eng. Technol. (IJARET) **11**(3), 128–134 (2020)
19. Röder, M., Both, A., Hinneburg, A.: Exploring the space of topic coherence measures. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pp. 399–408 (2015)
20. Saif, H., Fernandez, M., He, Y., Alani, H.: SentiCircles for contextual and conceptual semantic sentiment analysis of Twitter. In: Presutti, V., d'Amato, C., Gandon, F., d'Aquin, M., Staab, S., Tordai, A. (eds.) ESWC 2014. LNCS, vol. 8465, pp. 83–98. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07443-6_7
21. Sharma, K., Seo, S., Meng, C., Rambhatla, S., Dua, A., Liu, Y.: Coronavirus on social media: analyzing misinformation in Twitter conversations. arXiv preprint arXiv:2003.12309 (2020)
22. Tang, N., Yu, J.X., Wong, K.F., Li, J., et al.: Fast xml structural join algorithms by partitioning. J. Res. Pract. Inf. Technol. **40**(1), 33 (2008)
23. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A.: Sentiment strength detection in short informal text. J. Am. Soc. Inform. Sci. Technol. **61**(12), 2544–2558 (2010)
24. World Health Organization: Coronavirus disease (COVID-19) pandemic (2020). https://www.who.int/emergencies/diseases/novel-coronavirus-2019. Accessed 15 May 2020
25. Yang, S., Huang, G., Cai, B.: Discovering topic representative terms for ShortText clustering. IEEE Access **7**, 92037–92047 (2019). https://doi.org/10.1109/ACCESS.2019.2927345. https://ieeexplore.ieee.org/document/8756216/
26. Yang, S., Huang, G., Ofoghi, B., Yearwood, J.: Short text similarity measurement using context-aware weighted biterms. In: Concurrency Computation. Wiley (2020). https://doi.org/10.1002/cpe.5765
27. You, Q., Luo, J., Jin, H., Yang, J.: Cross-modality consistent regression for joint visual-textual sentiment analysis of social multimedia. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, pp. 13–22 (2016)
28. Zhou, J., Yang, S., Xiao, C., Chen, F.: Examination of community sentiment dynamics due to COVID-19 pandemic: a case study from Australia. arXiv preprint arXiv:2006.12185 (2020)
29. Zhou, J., Zogan, H., Yang, S., Jameel, S., Xu, G., Chen, F.: Detecting community depression dynamics due to COVID-19 pandemic in Australia. arXiv preprint arXiv:2007.02325 (2020)
30. Zhou, R., Liu, C., Li, J., Yu, J.X.: ELCA evaluation for keyword search on probabilistic xml data. World Wide Web **16**(2), 171–193 (2013)

FabricGene: A Higher-Level Feature Representation of Fabric Patterns for Nationality Classification

Shuang Yu[1,2], Xiongfei Li[1,2], Hancheng Wang[2], Xiaoli Zhang[1,2](✉), and Shiping Chen[3]

[1] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, China
yushuang17@mails.jlu.edu.cn, {lxf,xiaolizhang}@jlu.edu.cn

[2] College of Computer Science and Technology, Jilin University, Changchun, China
wanghc5516@mails.jlu.edu.cn

[3] CSIRO Data61, Sydney, NSW 2016, Australia
shiping.chen@data61.csiro.au

Abstract. Nationality fabric classification is a significant work to promote the protection work of fabric patterns and further reveal its unique connotation and inheritance rules in big data era. Thus, how to ascertain the feature representation of fabric patterns becomes a primary problem. This paper presents a high-level feature representation for fabric patterns for nationality classification, called FabricGene, which improves the semantic expression ability of the fabric pattern features. In fabric patterns, each FabricGene represents a complete abstract concept including the external shape and connotation characteristics. We evaluate the performance of FabricGenes and basic geometric primitives to illustrate the effectiveness of FabricGenes in nationality classification. Five widely used classification algorithms are applied to classify the fabric patterns by learning from training data with 12 groups of FabricGenes and 11 groups of basic geometric primitives respectively. The results demonstrate that the FabricGenes perform more effectively and stably in nationality classification than the basic geometric primitives. Namely, the FabricGenes can express the fabric patterns' nationality features more accurately.

Keywords: Fabric pattern · Feature representation · FabricGene · Basic geometric primitive · Nationality classification

1 Introduction

Cultural heritage contains a nationality's unique spirit mechanism, the way of thinking and cultural awareness. The Chinese Fabric patterns are one of the important intangible cultural heritages, which not only reveals people's cognizance to the life, religion, regional environment and natural beauty, but also reflects distinctive national characteristics and long historic culture. They always

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 624–638, 2020.
https://doi.org/10.1007/978-3-030-65390-3_47

appear in varieties of fabrics such as carpets and clothing, which plays a captivating decorative effect and conveys aesthetic needs of human life. Some folk fabric patterns are illustrated in Fig. 1. It's a meaningful and challenging work to investigate the theory and methods of mining intangible cultural heritage with fabric patterns as an example.

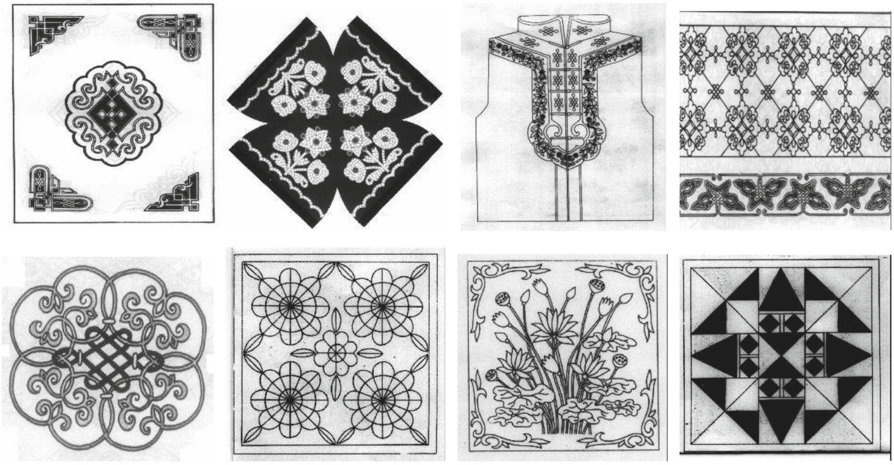


Fig. 1. Examples about folk fabric patterns

So far, there have been some studies on the analysis of cultural heritage characteristics based on varieties of patterns. For example, Foni et al. presented a general classification that might be employed to constitute a visual representation of a cultural heritage item [9]. Moorish geometric decorative pattern is a valuable artwork that combines the artistic style of Islam and Christianity creatively. Zarghilietet et al. described a new indexing method that can be used for indexing an Arabo-Moresque decor database [23,24]. Nasri and Benslimane proposed an original method that extracts the circle passing by the maximum pixels belonging to the binary rosette image, which was rather successful to describe the characterization of Islamic Rosette Patterns [16]. Waring introduced a global typology of kolam types and presented an extension to the square loop kolam (SLK) patterns [21]. For the fabric patterns, they are always collected from varieties of fabrics, and stored into computers in the form of image acquisition. Compared with the traditional way of storing in fabrics, the image acquisition method is more convenient, scientific and detailed to provide amounts of digital fabric patterns as the first-hand information for academia. Then the holographic database can be built. In a hierarchical representation of patterns, analysis always starts from the bottom feature. For varieties of patterns, the texture is one of their basic features. There have been many studies on the texture classification [1,12] and texture matching [3]. In digital fabric patterns, the texture information usually contains a variety of simple lines. Although the combination of some simple lines may compose a fabric pattern in external appearance, a single line has limited capacity in expressing the cultural connotation of fabric pattern features. Until now, there are few researches about exploring the cultural connotation of fabric patterns based on a higher-level feature representation.

In the process of mining cultural connotation, whether the culture can be analyzed quantitatively or not becomes a primary issue. Fortunately, Michel et al. proposed a new concept of "culturomics" for the linguistic and cultural phenomena, where the words in English language are considered to be the complete grammar units [13]. The survey indicated that the "culturomics" extended the boundaries of rigorous quantitative inquiry to a wide way of new phenomena spanning the social sciences and humanities. From Michel's research, the groundbreaking conclusion was drawn: *the culture can be calculated*. In this study, it can be observed that the words provides more information than the letters. Similarly, it might be better to consider the minimum meaningful unit of fabric patterns as the syntactic unit rather than the textures or simple lines.

In this paper, the "culturomics" is expanded to the field of fabric patterns and the concept of FabricGene is proposed as the syntactic units. Then we will concentrate on the feature representation of fabric patterns from the perspective of FabricGenes. It can not only improve the logic level of the fabric patterns in semantic expression, but also be conductive to mining its unique connotation and inheritance rules quantitatively especially in classification. Experiments have demonstrated that using the FabricGenes as the feature set can classify the fabric patterns more accurately than using the basic geometric primitives as the feature set. Then, the detailed process of constructing the gene pool is put forward to describe the patterns' main features with the help of humanists' suggestions. This makes it possible to incorporate more FabricGenes with the number of fabric patterns increasing. To the best of our knowledge, it is the 1^{st} time to use the "FabricGenes" to identify and predict the unknown fabric patterns, which is conductive to further exploring the cultural connotation and inheritance rules quantitatively.

2 Motivation

Basic geometric primitives can describe the fabric patterns' main characteristics to a certain extent. Two fabric patterns selected from different nationalities are presented in Fig. 2. The fabric pattern in Fig. 2(a) is from Mongolian, and the other is from Uighurs. The main geometric primitives forming the two fabric patterns are both the quadrilaterals, which are presented in Fig. 2(c). It's not difficult to understand that both of the fabric patterns would be classified into the same nationality if using the basic geometric primitives presented in Fig. 2(c) as their features. The main reason is that both of them consist of the same features in the level of basic geometric primitives that has limited capacity to express the features of fabric patterns.

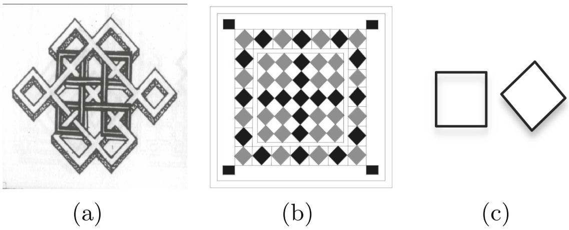


Fig. 2. Basic geometric primitives of two fabric patterns: (a) Mongolian fabric pattern; (b) Uighurs fabric pattern; and (c) basic geometric primitives forming the two fabric patterns.

In fact, the fabric pattern in Fig. 2(a) consists of two main components shown in Fig. 3(a) that are composed of some basic geometric primitives and can be explained with the cultural connotation of "Chinese knot". The main components of the fabric pattern in Fig. 2(b) are just quadrilaterals presented in Fig. 3(b). It's obvious that they have unique connotation characteristics respectively so that the two patterns may be classified into their corresponding nationalities.

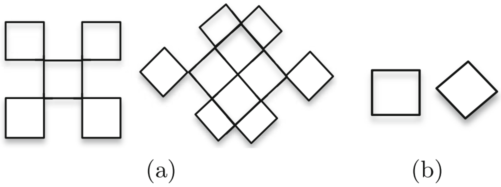


Fig. 3. Connotation characteristics of the two fabric patterns: (a) characteristics of Fig. 2(a); (b) characteristics of Fig. 2(b).

Nowadays, the cultural computability makes it possible to analyze the fabric patterns' characteristics quantitatively. The existing pattern features always consider putting the lower-level features such as simple lines, shapes and textures together according to certain rules. However, a variety of irregular lines, shapes, and textures make it impossible to quantitatively analyze a large number of fabric patterns that consists of complex structures and compositions. In addition, as mentioned above, the lower-level features have limited capacity to express the fabric patterns' connotation features. Thus, inspired by Michel's concept of "culturomics", it might be better to take the combination of external appearance and cultural connotation into account. In this paper, we present a higher-level fabric pattern feature named FabricGene as the syntactic unit, which is more representative, precise and effective in description of the fabric patterns' features than the basic geometric primitives. Furthermore, using the FabricGenes as the feature set can also assist researchers to conduct the data mining and analysis of fabric patterns more accurately.

It should be noted that the deep learning models [5,14,25] are not suitable to classify the fabric patterns. First, the deep learning models generally uses a very large amount of training data as support. Unfortunately, as one of the important intangible cultural heritages, the number of existing fabric patterns is limited, which cannot meet the training needs of the deep learning models. In addition, the deep learning models are black box models with uninterpretability, which is obviously not conducive to exploring the cultural connotation and inheritance rules of the fabric patterns.

3 The Feature Sets of Fabric Patterns

In the field of patterns, experts have obtained some research results. Auguste had authored some sourcebooks, featuring over 1,500 decorative elements and motifs from the major cultures in history through the 19th century, from Asia and Africa to Europe and the Americas [19,20]. The Chinese scholar Wu has authored a series of books to summarize and exhibit the Chinese dermatoglyphic patterns comprehensively [22]. In particular, what we learn are the core and essence of the Chinese folk fabric patterns integrated by the authors. However, the features of fabric patterns in these professional literatures have not been analyzed quantitatively. In this paper, we draw on the achievements and suggestions of these experts and propose more characteristic and meaningful FabricGenes to conduct the quantitative mining and analysis on fabric patterns.

3.1 FabricGene

With the development of times, the culture in each nationality or region has developed, evolved and merged with each other continuously. China is a typical multinational country with various fabric pattern styles. Each fabric pattern style not only contains the unique national characteristics, but also reflects the corresponding history culture. Therefore, there is a far-reaching significance for the continuation of traditional culture to inherit and mine the essence of fabric patterns. In addition, the nationality characteristic is one of important connotation characteristics of fabric patterns, so the nationality classification is a significant job to promote the protection of fabric patterns and further reveal its unique connotation and inheritance rules in big data era. Thus, it is vital for the data mining of fabric patterns to find the essence of fabric patterns and regard it as the feature representation. Humanists argue that the cultural genes can represent the certain cultural features. However, they can't be quantified according to the notion. In this paper, the FabricGenes are proposed to describe the main features of fabric patterns so that the concept hierarchy of features can be upgraded to a higher level. In this case, how to quantify and analyze the FabricGene becomes a primary issue. In external appearance, the FabricGenes are always comprised of the basic geometric primitives. In connotation characteristics, each FabricGene contains the unique cultural connotation, which can be used to mine the core and essense of cultural genes. First, the definitions of basic geometric primitive and cultural gene are proposed as follows,

Definition 1 ***Basic Geometric Primitive.*** *The simple geometric figures that contains little cultural connotation is regarded as the basic geometric primitives of fabric patterns, including straights, curves, polygons and some other simple shapes used frequently.*

Definition 2 ***Cultural Gene.*** *The geometric primitives that can be explained as the specific meanings by humanists is named as the cultural genes.*

The cultural genes play a leading role in such aspects as cultural connotation and inheritance. The FabricGenes in fabric patterns are considered as the basic units combined with linellaes and shapes that are used frequently or preferred by a certain ethnic group and contain certain meanings. Thus, the FabricGenes can be not only small to a straight, a curve, a circle, a polygon or other simple shapes, but also big to the combination and variation of many basic primitives. Like the culture genes, the FabricGenes are the smallest semantic units of the fabric patterns. For the fabric patterns, the logic integrity is mainly manifested from two aspects, namely the external appearance and the cultural connotation. Note that each FabricGene in this study not only contains the unique external appearance, but also expresses the corresponding cultural connotation information, which makes the logic integrity of fabric patterns be well maintained. Thus, it is meaningful to conduct the fabric pattern mining at the level of FabricGenes. According to the symmetry, we can obtain a meaningful non-repeating part of a fabric pattern (MNRP) of a fabric pattern. All the basic geometric primitives in the MNRP constitute the primitive set of the fabric pattern, which is denoted as P. The ith geometric primitive in P is denoted as p_i. For each geometric primitive $p_i \in P$, it contains another important feature namely the cultural connotation that can be obtained from the subjective suggestions of humanists. The cultural connotation feature of p_i is denoted as q_i. Then the connotation set Q of the fabric pattern can be constituted. For each pair of geometric primitives $p_i, p_k \in P$, we denote it as $p_i \wedge p_k$ if p_i and p_k is connective; otherwise $p_i \vee p_k$. The final FabricGene we obtain is denoted as FG.

Definition 3 ***FabricGene.*** *Let the primitive set of a fabric pattern be $P = \{p_1, p_2, \cdots, p_n\}$, and the corresponding connotation set be $Q = \{q_1, q_2, \cdots, q_n\}$, where n represents the number of primitives in P or Q. $\forall\ \widehat{P} \subset P$ where $\widehat{P} = \{p_1, p_2, \cdots, p_m\}$, $\widehat{P}$ will be incorporated into the candidate FabricGene set FGS if there is a rearrangement in $\widehat{P}$ that makes $p_{1'} \wedge p_{2'} \wedge \cdots \wedge p_{m'}$. At the same time, the connotation set corresponding to $\widehat{P}$ is denoted as $\widehat{Q} = \{q_1, q_2, \cdots, q_m\}$. According to subjective suggestions and experience of humanists, each candidate FabricGene $\widehat{P}$ in FGS can be regarded as a FG if the integrated connotation expressed by the combination of each q_i in $\widehat{Q}$ can be given an actual culture meaning by humanists.*

According to the combination of objective rules and subjective opinions mentioned above, the FabricGenes can be determined finally. It can be observed that each FabricGene is an interconnected and complete unit, which maintains the basic cultural characteristic of fabric patterns. The FabricGenes that have been discovered so far include varieties of flowers, geometric shapes, rattans, star patterns and so on. We illustrate a part of Mongolian and Uighurs FabricGenes with an example in Fig. 4. It can be found that a specific FabricGene always represents a complete abstract concept that contains more certain cultural connotation than the basic geometric primitive.

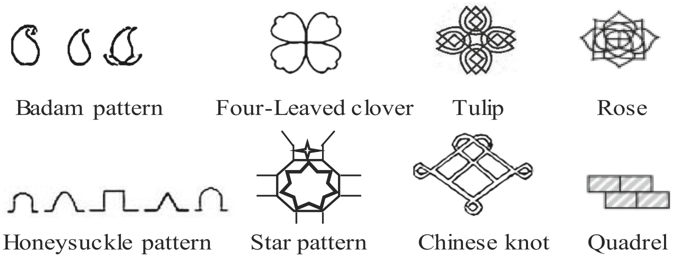
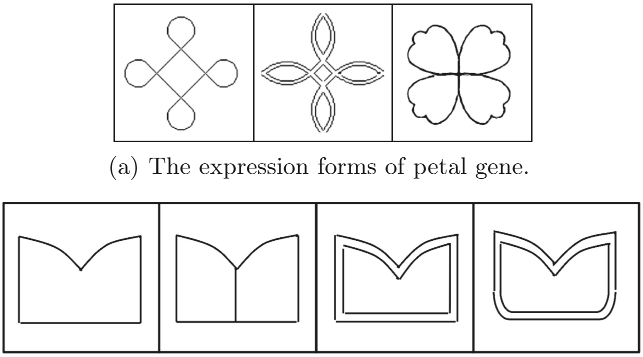


Fig. 4. A part of Mongolian and Uighurs FabricGenes.

3.2 Feature Representation with FabricGenes

A FabricGene may be expressed by a variety of forms due to the richness and complexity contained by the fabric patterns. An example in Fig. 5 shows that there are many variants that are similar in the external appearance but represent the same cultural connotation. These similar FabricGenes should be assigned into the same group. Through this way, we have sorted out 12 groups of FabricGenes for the fabric patterns of Mongolian and Uighurs. In Fig. 6, a part of four groups of FabricGenes are illustrated to provide a straight insight on the relationship among different groups of FabricGenes, including the similarity in the same group and the difference in different groups. The final feature set $\{Gene_1,Gene_2,...,Gene_{12},C\}$ is a set of FabricGenes, where C represents the nationality to which the fabric pattern belongs. For the two Mongolian fabric patterns in Fig. 7(a) and Fig. 8(a) whose C value are both 0, their vectorgraphs are presented in Fig. 7(b) and Fig. 8(b) respectively, and their MNRP is also given in Fig. 7(c) and Fig. 8(c) respectively. There are five kinds of FabricGenes in Fig. 7(c), namely $Gene_4$, $Gene_5$, $Gene_6$, $Gene_7$ and $Gene_8$ discovered in this study. Thus, the fabric pattern in Fig. 7(a) are mapped into a feature vector expressed as (0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0). Likewise, two FabricGenes presented in Fig. 8(c) are $Gene_2$ and $Gene_6$ respectively. Then the fabric pattern in Fig. 8(a) are mapped into a feature vector expressed as (0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0). Note that both of the two fabric patterns belong to the Mongolian and contain $Gene_6$, which is conductive to their nationality classification.



(a) The expression forms of petal gene.

(b) The expression forms of leaf gene.

Fig. 5. A variety of expression forms of FabricGenes.

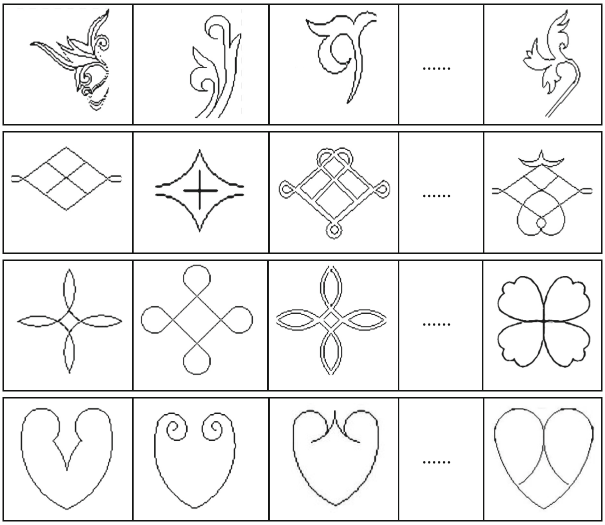


Fig. 6. An example to reveal the relationship among different groups of FabricGenes.

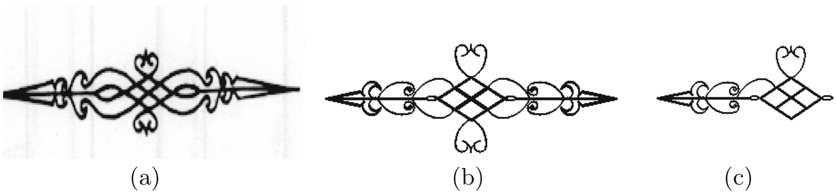


Fig. 7. The first example of constructing the fabric pattern's feature vector: (a) the Mongolian fabric pattern; (b) the vectorgraph; (c) the MNRP.

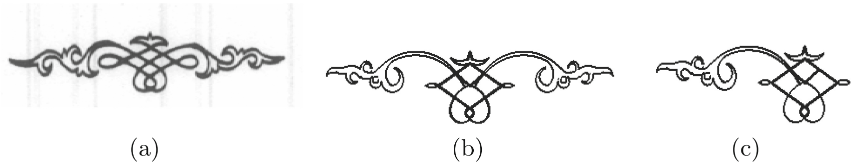

(a) (b) (c)

Fig. 8. The second example of constructing the fabric pattern's feature vector: (a) the Mongolian fabric pattern; (b) the vectorgraph; (c) the MNRP.

All the fabric patterns will constitute a vector space used for analyzing the fabric patterns quantitatively. In fact, this form of vector space has been widely used in various fields. For example, Mikolov et al. conducted the estimation of word representations in vector space [15]. Similarly, all quantitative analysis and mining of fabric patterns will also be performed in the feature vector space. At the same time, 11 groups of basic geometric primitives are also sorted out in the same way.

3.3 The Construction of FabricGene Pool

The fabric patterns usually consist of external characteristics and connotation characteristics. In this paper, the construction of FabricGene pool is similar to the construction of ontology and mapping knowledge domain that are dependent on human cognition in a certain degree. In the fabric patterns, the connotation characteristics are usually obtained from the domain experts' subjective interpretation and analysis, including the era, region, usage, nationality, religion and meanings. In this section, the gene pool has been sorted out through analyzing the unique appearance and cultural connotation of existing Mongolian and Uighurs fabric patterns. As shown in Fig. 9, the process of constructing the FabricGene pool can be summarized as follows,

Step 1: Vectorize the fabric patterns stored by the bitmap.
Step 2: Extract the FabricGenes by taking the unique appearance and connotation into account.
Step 3: Update the FabricGene pool and divide the FabricGenes into the corresponding groups according to their unique appearances and cultural connotation.

The vectorization of fabric patterns can not only reduce the storage space of fabric patterns greatly, but also be easy for us to edit the fabric patterns without distortion. In the process of modeling for the fabric patterns, each FabricGene contains its unique appearance and connotation that are useful to extract different FabricGenes in each fabric pattern. Through analyzing the fabric patterns on the FabricGene pool, the representation level of fabric patterns can be upgraded

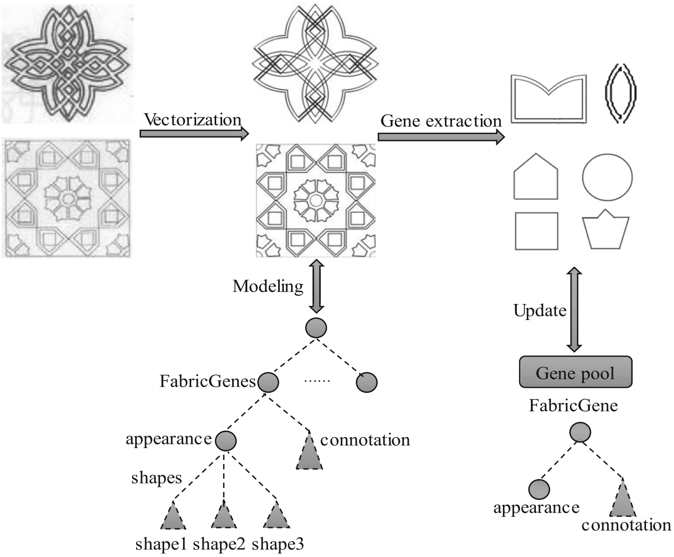


Fig. 9. The process of constructing the FabricGene pool.

to the level of cultural genes. In addition, the construction of FabricGene pool makes it possible to incorporate more and more new FabricGenes with the number of fabric patterns increasing, which helps us not only construct a more complete FabricGene database, but also conduct the sustained study and analysis on the fabric patterns.

4 Experiments Results and Analysis

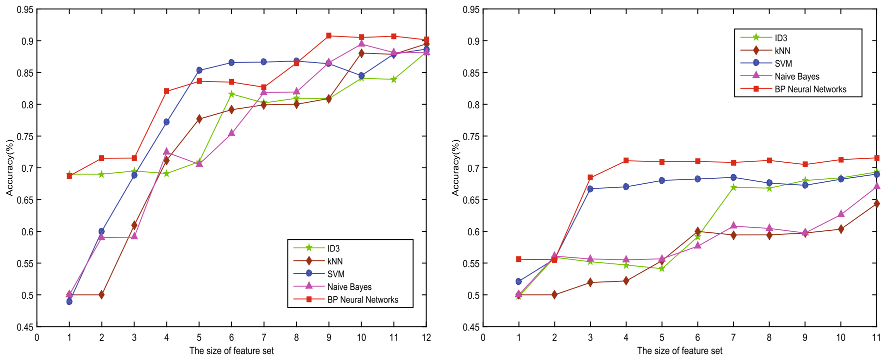
4.1 Experiment Setup

To evaluate the effectiveness of our proposed feature set for nationality classification, we perform experiments on a collection of 300 fabric patterns from two nationalities (Mongolian and Uighurs). Furthermore, two kinds of feature sets, including 12 groups of FabricGenes and 11 groups of basic geometric primitives, are used for the nationality classification. The fabric patterns are represented as the feature vectors, which are learned in this paper by several classical and widely used machine leaning methods, including the BP Neural Networks [18], the Support Vector Machine (SVM) [6,10], the k-Nearest Neighbour (kNN) [17], the Naive Bayes [2], and the ID3 [8]. Although the principles of these classifiers are different from each other, they can always obtain a quickly solution for current various of issues [4,7,11]. Then, two kinds of experiments based on different sizes of the feature set and different sizes of the training set are conducted on Windows 10 with a 3.2 GHz Intel Core i5 CPU and 8 GB RAM. In the first kind of the experiment, each feature set is constituted by adding a new feature to the previous feature set. Thus, 12 groups of experiments based on the FabricGenes and 11 groups of experiments based on the basic geometric primitives are conducted with the size of feature set increasing from 1 to 12 and 1 to 11 respectively. For each size of the feature set, the experiment is repeated 100 times with the random instance partitions and then the mean accuracy are calculated as the final results. At each repetition, the dataset is randomly divided into two subsets: one is the training set (two-thirds) and the other is the test set (one-third). In the second kind of the experiment, with the size of training set increasing from 10 to 200, 20 groups of experiments are conducted by respectively using 12 groups of FabricGenes and 11 groups of basic geometric primitives as the feature sets. Note that we conduct the experiment for each size of the training set in the same way as we did for each size of the feature set. Finally, we can also obtain the mean accuracy at each size of training set as the final results.

4.2 Experiment Results and Discussions

The classification mainly contains two steps: first, all the features are extracted in the preceding stage; then, two kinds of feature sets are learned by five classical and extensively used classifiers mentioned above, reporting the classification accuracy. Now, the mean accuracy on different numbers of FabricGenes and basic geometric primitives are respectively presented in Fig. 10(a) and 10(b), in order to intuitively observe the influence of different sizes of the feature set on classification accuracy. From Fig. 10(a) and 10(b), the classification accuracy obtained by the five classifiers present the upward trend overall as the size of the feature set increases. Furthermore, it can also be found that the overall classification accuracy of the five classifiers in Fig. 10(a) (using the FabricGenes as the feature set) are respectively much higher than that of the corresponding classifiers in Fig. 10(b) (using the basic geometric primitives as the feature set). The comparison between Fig. 10(a) and Fig. 10(b) demonstrates that the FabricGenes are more representative and effective in the description of fabric patterns' nationality features than the basic geometric primitives so that the nationality classification accuracy can be improved greatly. In other words, the basic geometric primitives have limited capacity in the expression of the fabric pattern features.

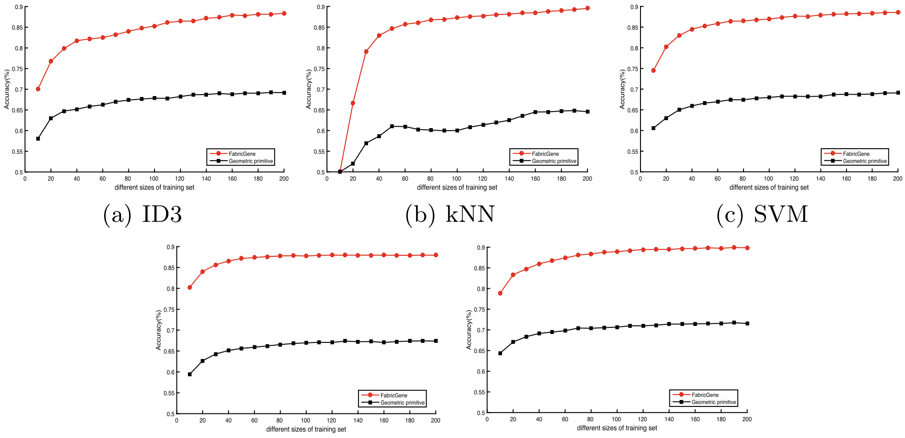
In addition, we also calculate the classification accuracy of the five classifiers at different sizes of the training set, based on 12 groups of FabricGenes and 11 groups of basic geometric primitives respectively. The results are shown in Fig. 11, where the influence of different sizes of the training set on the classification accuracy is intuitively observed. Not surprisingly, with the size of the training set increasing, the classification accuracy of five classifiers present stably upward trend overall. Moreover, at each size of the training set, the classification accuracy of these five classifiers based on the FabricGenes are respectively much higher than that of the corresponding classifiers based on the basic geometric primitives. The analysis above indicates that the FabricGenes indeed perform stably and more effectively in the nationality classification of fabric patterns.



(a) (b)

Fig. 10. Classification accuracy on different sizes of the feature sets: (a) FabricGenes; (b) basic geometric primitives.

The reason is that the FabricGenes contain richer national characteristics and cultural connotation than the basic geometric primitives.



(a) ID3 (b) kNN (c) SVM

(d) Naive Bayes (e) BP Neural Networks

Fig. 11. Classification accuracy of five classifiers at different sizes of the training set based on 12 groups of FabricGenes and 11 groups of basic geometric primitives.

In the previous experiments, it can be seen that with the growth of feature set and training set, all of the five classifiers have much higher classification accuracy if using the FabricGenes as the feature set. In order to more comprehensively verify the effectiveness of FabricGenes in nationality classification, we additionally utilize the receiver operating characteristic (ROC) curve to quantitatively measure the overall classification performance based on different feature sets. The

ROC is a commonly used evaluation metric due to its strong distinguish ability. A series of ROC analysis are conducted to evaluate the performance of the five classifiers based on the FabricGenes and basic geometric primitives respectively. Through this evaluation we can confirm whether the feature set of FabricGenes are more suitable for classifying the fabric patterns or not. The ROC analysis of the five classifiers based on different feature sets are shown in Table 1. It can be observed from Table 1 that the AUC values of the five classifiers based on the FabricGenes are respectively much higher than that of the corresponding classifiers based on the basic geometric primitives. Not surprisingly, the standard error values of the five classifiers based on the FabricGenes are also respectively much lower than that of the corresponding classifiers based on the basic geometric primitives. In summary, the FabricGenes do have stronger expression ability to the nationality features of fabric patterns than the basic geometric primitives.

Table 1. The AUC values of five classifiers based on the FabricGenes (FG) and basic geometric primitives (BGP).

Test result variable (s)	Feature set[a]	AUC	Standard error	Asymptotic 95%	Confidence interval
				Lower bound	Upper bound
Naive Bayes	FG	.967	.015	.939	.996
	BGP	.728	.051	.629	.827
KNN	FG	.964	.016	.932	.996
	BGP	.744	.050	.646	.843
BPNeuralNetwork	FG	.958	.020	.920	.997
	BGP	.745	.050	.648	.842
SVM	FG	.935	.025	.886	.984
	BGP	.645	.057	.533	.756
ID3	FG	.894	.035	.825	.963
	BGP	.728	.051	.629	.828

[a] FG represents the FabricGene, and BGP represents the basic geometric primitive.

Through the analysis above, a conclusion is drawn that the FabricGenes are truly more effective than the basic geometric primitives in describing the fabric patterns' main features and further nationality classification.

5 Conclusion and Future Work

In this paper, a higher-level feature representation of fabric patterns (named FabricGenes) is proposed as the cultural syntactic unit of fabric patterns. It focuses on the combination of external appearance and cultural connotation. Then the FabricGenes are used as the feature set for the nationality classification of fabric patterns. Five classical and widely used classifiers are utilized to conduct a series of experiments, aiming to verify the effectiveness and superiority of the FabricGenes on the nationality classification of fabric patterns. The results demonstrate that the FabricGenes perform more effectively and stably in nationality classification with the growth of the training set and feature set. The reason is that the FabricGene is a higher-level syntactic unit, which can reflect the corresponding nationality's unique cultural connotation in a certain degree. Moreover, the FabricGene pool is also constructed, which is conductive to not only constructing a more complete FabricGene database, but also conduct the sustained study and analysis on the fabric patterns.

Note that the fabric patterns' computability contributes significantly to the expert systems, similar to "culturomics". In this paper, using the FabricGenes for nationality classification is only a preliminary study of fabric patterns' computability. In the future, we will collect and investigate as many fabric patterns of different nationalities as possible although the number of existing fabric patterns is limited, so as to enrich the FabricGene database. Moreover, it is worthy of further study to extend the fabric patterns' computability into more fields such as era classification, region prediction, meaning investigation and so on. In addition, the construction of FabricGene pool is similar to the construction of the ontology and mapping knowledge domain. In the further research, we will be less dependent on the domain experts and mine more valuable information in fabric patterns based on the FabricGenes. Furthermore, we will also pay more attention to explore how to utilize the classifiers to mine more pattern evaluation rules and design a generation model of fabric patterns that is full of national characteristics.

Acknowledgements. The work was supported in part by the National Natural Science Foundation of China under Grant 61801190, in part by the Nature Science Foundation of Jilin Province under Grant 20180101055JC, in part by the Outstanding Young Talent Foundation of Jilin Province under Grant 20180520029JH, in part by the Industrial Technology Research and Development Funds of Jilin Province under Grant 2019C054-3, in part by the Thirteenth Five-Year Plan Scientific Research Planning Project of Education Department of Jilin Province (JJKH20200678KJ, JJKH20200997KJ), and in part by the Fundamental Research Funds for the Central Universities, JLU. We also acknowledge the support of the National Museum of China for their contribution to provide us with fabric patterns.

References

1. Ahmadvand, A., Daliri, M.R.: Rotation invariant texture classification using extended wavelet channel combining and LL channel filter bank. Knowl. Based Syst. **97**, 75–88 (2016)
2. alias Balamurugan, A., Rajaram, R., Pramala, S., Rajalakshmi, S., Jeyendran, C., Prakash, J.D.S.: Nb+: an improved Naive Bayesian algorithm. Knowl. Based Syst. **24**(5), 563–569 (2011)
3. Arifoglu, D., Sahin, E., Adiguzel, H., Duygulu, P., Kalpakli, M.: Matching Islamic patterns in Kufic images. Pattern Anal. Appl. **18**(3), 601–617 (2015). https://doi.org/10.1007/s10044-014-0437-z
4. Beucher, A., Møller, A.B., Greve, M.H.: Artificial neural networks and decision tree classification for predicting soil drainage classes in Denmark. Geoderma **352**, 351–359 (2019)
5. Bouwmans, T., Javed, S., Sultana, M., Jung, S.K.: Deep neural network concepts for background subtraction: a systematic review and comparative evaluation. Neural Netw. **117**, 8–66 (2019)

6. Chauhan, V.K., Dahiya, K., Sharma, A.: Problem formulations and solvers in linear SVM: a review. Artif. Intell. Rev. **52**(2), 803–855 (2018). https://doi.org/10.1007/s10462-018-9614-6
7. Deng, W., Yao, R., Zhao, H., Yang, X., Li, G.: A novel intelligent diagnosis method using optimal LS-SVM with improved PSO algorithm. Soft. Comput. **23**(7), 2445–2462 (2019). https://doi.org/10.1007/s00500-017-2940-9
8. Fletcher, S., Islam, M.Z.: Decision tree classification with differential privacy: a survey. ACM Comput. Surv. (CSUR) **52**(4), 1–33 (2019)
9. Foni, A.E., Papagiannakis, G., Magnenat-Thalmann, N.: A taxonomy of visualization strategies for cultural heritage applications. J. Comput. Cult. Heritage (JOCCH) **3**(1), 1–21 (2010)
10. Hamel, L.H.: Knowledge Discovery with Support Vector Machines, vol. 3. Wiley, Hoboken (2011)
11. Hasson, U., Nastase, S.A., Goldstein, A.: Direct fit to nature: an evolutionary perspective on biological and artificial neural networks. Neuron **105**(3), 416–434 (2020)
12. Liu, L., Fieguth, P.: Texture classification from random features. IEEE Trans. Pattern Anal. Mach. Intell. **34**(3), 574–586 (2012)
13. Michel, J.B., Shen, Y.K., et al.: Quantitative analysis of culture using millions of digitized books. Science **331**(6014), 176–182 (2011)
14. Miikkulainen, R., Liang, J., et al.: Chapter 15 - evolving deep neural networks. In: Kozma, R., Alippi, C., Choe, Y., Morabito, F.C. (eds.) Artificial Intelligence in the Age of Neural Networks and Brain Computing, pp. 293–312. Academic Press, Cambridge (2019)
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
16. Nasri, A., Benslimane, R., et al.: A rotation symmetry group detection technique for the characterization of Islamic rosette patterns. Pattern Recogn. Lett. **68**, 111–117 (2015)
17. Nowicki, R.K.: Rough Set–Based Classification Systems. SCI, vol. 802. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-03895-3
18. Prieto, A.: Neural networks: an overview of early research, current frameworks and new challenges. Neurocomputing **214**, 242–268 (2016)
19. Racinet, A.: Racinet's historic ornament in full color: all 100 plates from "l'ornementpolychrome". Courier Corporation (2012)
20. Racinet, A.: Full-color picture sourcebook of historic ornament: all 120 plates from "l'ornement polychrome". Courier Corporation (2013)
21. Waring, T.M.: Sequential encoding of Tamil Kolam patterns. J. Forma **27**, 83–92 (2012)
22. Shan, W.: The complete works of Chinese patterns. Shandong Fine Art Press, Shan Dong (2009)
23. Zarghili, A., Gadi, N., Benslimane, R., Bouatouch, K.: Arabo-Moresque Decor image retrieval system based on mosaic representations. J. Cult. Heritage **2**(2), 149–154 (2001)
24. Zarghili, A., Kharroubi, J., Benslimane, R.: Arabo-Moresque decor images retrieval system based on spatial relationships indexing. J. Cult. Heritage **9**(3), 317–325 (2008)
25. Zhou, D.-X.: Universality of deep convolutional neural networks. Appl. Comput. Harmonic Anal. **48**(2), 787–794 (2020)

Low-Light Image Enhancement with Color Transfer Based on Local Statistical Feature

Zhigao Zhang[1,2] and Bin Wang[1](✉)

[1] School of Computer Science and Engineering, Northeastern University,
Shenyang 110004, China
zzgtongxin@163.com, binwang@mail.neu.edu.cn

[2] College of Computer Science and Technology, Inner Mongolia University
for Nationalities, Tongliao 028000, China

Abstract. The image obtained under the condition of extremely low illumination has the characteristics of high noise, low contrast and the inability to display the detail information in the dark area. These features seriously reduce the visual quality of image and also degrade the performance of other computer vision algorithms that are designed for high quality input. In this paper, we propose a color transfer method based on the local color statistical feature to enhance the low-light image. In this method, an image with clear texture of normal exposure is used as a reference image. Our strategy is to cluster the reference image and the low-light image into several local areas according to the first-order statistical characteristics of color, and then implement color transfer between the locally optimal matching regions to achieve color enhancement of the dark image. In order to obtain an accurate and smooth category area, we propose a new GMM model to fit the color component of the image. In this model the spatial neighborhood relation of pixel coordinates is taken into account and coordinate space smoothing parameters are added. Meanwhile, in the parameter estimation of GMM model, we propose a new method for Gaussian component merging, which can effectively reduce the clustering error. We conducted a number of experiments on real-world low-light image data sets to reveal the effectiveness of our approach, and the results show that our approach is superior over several state-of-the-art methods in both objective and subjective evaluation.

Keywords: Low-light image enhancement · GMM · Image segmentation · Color transfer

1 Introduction

It is well known that low-light image enhancement is one of the most challenging tasks. The key of low-light image enhancement is how to improve the

The work is partially supported by the Fundamental Research Funds for the Central Universities (No. N171602003), and Liaoning Distinguished Professor (No. XLYC1902057).

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 639–653, 2020.
https://doi.org/10.1007/978-3-030-65390-3_48

visual effect and reveal the details hidden in the image so as to be more suitable for other image processing tasks, such as medical image analysis [16], video surveillance [23], face recognition [7], target object detection [6], etc.. Good light environment is an important condition to ensure image quality, however, in many real scenarios, the ideal lighting conditions are often difficult to meet.

In order to make the hidden details in low-light image visible again, several contrast enhancement techniques have been proposed. Among them, the most direct method is to linearly amplify the brightness component of the low-light image. However, when this method is used to enlarge the dark area, the relatively bright area in the image will be excessively enhanced, resulting the details of the bright area to be lost. Histogram equalization strategy can force the output image to fall within the range of [0, 1] to avoid the above problem [1]. However, in practical applications, they focus on contrast enhancement, rather than using real brightness information to adjust, there is a risk of over-enhancement and under-enhancement.

Inspired by the techniques of image segmentation and color transfer, in this paper, a new method based on image segmentation and color transfer is proposed for low-light image enhancement. In this method, we select a clear image (hereinafter referred to as the "reference image") as a reference image, and then cluster it into several independent regions according to the color statistics feature, we do the same work for the low-light image (referred to as the "input image"). Our strategy is to find the optimal match of each region of the input image in the reference image, and transfer the color components of the reference image region to the corresponding dark image region to achieve image enhancement. Therefore, how to effectively and accurately segment the image and find the optimal matching region between the two images are the key work. To solve the above problems, we propose a new Gaussian mixture model (GMM) that takes into account the spatial neighborhood pixel coordinate relations and adds coordinate space smoothing parameters. Meanwhile, a new Gaussian component merging method is also proposed to ensure the accurate smooth color region. Finally, a map function is proposed to set up the region mapping relation of input-reference image color categories, which can find the optimal matching region for each region of the input image. The overall framework is shown in Fig. 1. We also conducted a large number of experiments on the dark image dataset to verify its effectiveness. Compared to some baseline methods and the state-of-the-art methods, results show that our method achieves better performance in both objective evaluation and subjective evaluation.

In a word, we introduce a method for low light image enhancement using image segmentation and color transfer technology. Our main contributions are as follows:

- We propose a new model for low-light image enhancement. As to our best knowledge, this is the first attempt to combine image segmentation and color transfer technology to implement low-light image enhancement.

- In order to obtain the accurate and smooth color regions, a new Gaussian mixture model considering spatial neighborhood coordinate relationship is proposed, which adopts a new Gaussian component merging method.
- We propose a mapping method between image color categories, which can establish the optimal match between the input image color region and the reference image.

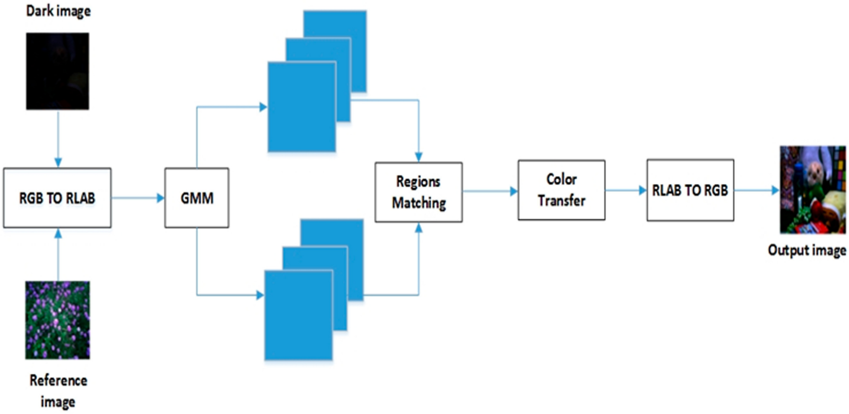


Fig. 1. The overall framework and flow of the proposed method

2 Related Work

For low-light image enhancement, many attempts have been made, among them histogram transformation is the basic processes, which is based on spatial operation, such as histogram equalization [13], dual-histogram equalization based on brightness retention [11], adaptive histogram equalization [15]. These algorithms directly or indirectly change the dynamic range of images through mapping functions to improve contrast. In recent years, many scholars have proposed improved algorithms based on histogram. Such as, Contrast limited adaptive histogram equalization algorithm [14], adaptive gamma correction with weighting distribution [9], adaptive contrast enhancement algorithm [27]. Although these methods improve performance to some extent, their effects are very limited. This is because in the large flat area of dark intensity, the low-light image contains severe noise, and the histogram based method will cause noise amplification while enhancing these areas. Recently, some model-based enhancement methods have been proposed, such as [5] [12], these methods can improve the visual quality of images to some extent, but their disadvantages are poor interpretation and easy to introduce artifacts.

Color transfer technique was first proposed by Reinhard et al. [17], which enables two images to have similar color distribution characteristics by transferring the color content of the reference image to another image. Color transfer algorithms are mainly divided into three categories: methods based on geometric features, user-assisted methods, and statistical features. Among them, the last method is simple and efficient, and achieves good results in many scenarios. On this basis, many subsequent methods based on color statistics have been proposed. Welsh et al. proposed a method of colorization for grayscale images [21]. Xiao et al. proposed a method for color transfer in arbitrary three-dimensional color space [22]. He et al. abstracted color combination into color emotion and used color emotion for color transfer [8]. Color transfer technology has been successfully applied in image style conversion [10], gray color conversion and scene segmentation, but it is seldom applied in image enhancement, especially in the field of low illumination.

At present, most of the existing low-light image enhancement methods use a single image for processing without considering the local statistical characteristics of the image, which fails to effectively suppress the noise and leads to serious noise amplification after contrast enhancement [24]. Although some methods have the capability of noise reduction, they sacrifice the texture and detail of low-light image. Significantly different from these methods, we introduce the reference image to realize the color enhancement of the low-light image through color transfer between the reference image and the local optimal matching region of the dark image. Our method can suppress the noise perceptually while retaining the details of the image. At the same time, compared with other model-based methods, it has better interpretability and less time consumption.

3 Methodology

3.1 Image Color Clustering Segmentation Based on GMM

In this paper, we use the GMM model to implement image segmentation and color transfer in RLAB color space, and then convert to RGB space to show the final enhancement. The conversion between them is described in detail in reference [2]. Due to the limitation of space, this paper omits the introduction of the method. GMM is a hybrid model with weighted combining multiple Gaussian model, Theoretically, any probability distribution can be approximated by increasing the number of models [4]. So, we used the model to simulate the distribution characteristics of color categories in the image, and use EM algorithm to determine the number of color categories. In order to obtain a more accurate category region with smooth boundary, we take into consider the spatial neighborhood relations of pixel coordinate in the GMM model, and added coordinate spatial smoothing parameters. At the same time, when the parameters of the model are established, the Gaussian component merging condition is changed to solve the problem of inaccurate clustering.

Image Color Distribution. Images can be regarded as scattered pixels in spatial coordinates, which do not conform to any distribution model in a strict sense. In this paper we assume that the image of the color distribution can be divided into several regions, each region are independent and the pixels in each region as well. So each color pixel in the region is subject to the same Gaussian distribution, multiple color area through weighted together is an image of the overall color distribution, so that it can take advantage of the GMM model to simulate the distribution characteristics of the image color. Assume that the number of initial color categories of the image is k. μ_i, Σ_i represent the mean value and covariance of the current category area respectively (the main diagonal elements are variances), $i = \{1, 2, 3, \ldots .k\}$. The initial values of μ_i and Σ_i can be calculated by k-means clustering method. The color distribution of each color category can be approximately represented as a Gaussian component $G(\varphi_i)$, and the probability density of pixel $I(p)$ belonging to the Gaussian component $G(\varphi_i)$ can be represented as:

$$P(I(p)|\varphi_i) = \frac{1}{(2\pi)^3|\Sigma_i|^{1/2}} \times exp(-\frac{1}{2}(I(p) - \mu_i)^T \Sigma^{-1}(I(p) - \mu_i)) \qquad (1)$$

where, $\varphi_i = \{\mu_i, \Sigma_i\}$ denotes the parameter set of the current Gaussian component. So, the Gaussian mixture model G() fitting the color distribution of the whole image can be represented by the weighted probability density combination of each color category, and its density distribution function is:

$$P(I(p)|\varphi) = \sum_i^k a_i P(I(p)|\varphi_i) \qquad (2)$$

Where, $a_i = Z_i/Z_t$ represents the weight of the current Gaussian component, Z_i represents the number of pixels in the current Gaussian component, Z_t represents the number of pixels in the entire image. a_i meets the condition $a_i \geq 0$ and $\sum_i^k a_i = 1$, its initial value can be obtained by k-means. Then, the EM algorithm can be used to iteratively solve the parameters of Eq. 2.

The Parameters. Although the GMM model can fit the image color distribution well, there are still several problems when the EM algorithm is used to determine the model parameters. Firstly, the model only considers pixels as independent variables, but fails to consider the neighborhood relationship in the coordinate space of pixels, which is closely related to the content semantics of images. Therefore, when clustering image colors, the spatial relation of pixel coordinates is not considered, which may lead to inappropriate clustering results and affect the effect of color transfer. Secondly, when classifying pixel points into a certain category, it is judged by the probability of membership degree, and then directly classified into a certain category. In this way, the color region boundary of the image will be relatively "stiff", which will lead to an unnatural phenomenon of color fusion between the resulting image regions during the color transfer process. Finally, when category regions are merged, as long as the mean value difference between regions is less than a given threshold, they will be merged. Such combination conditions have certain problems, which may cause the final color clustering error.

In order to solve the above problems, we improved EM to obtain more accurate GMM model parameters. Specifically, in the E-step of EM algorithm, the spatial domain relation of pixel coordinates is considered when calculating the probability of each data belonging to a certain Gaussian sub-model, and the coordinate space smoothing parameter is added to make the color category region divided by GMM model in the color space have continuity and the edge of the category region is relatively smooth; When maximizing the model parameters in m-step, the Gaussian component merging condition is changed, that is, the Gaussian component merging is carried out in each iteration process, and the parameter estimation is carried out again to solve the clustering error problem. The parameter estimation process of the improved EM algorithm is as follows:

E-Step: Add the Coordinate Space Smoothing Parameters
We define coordinate space smoothing parameters such as Eq. 3,

$$P^{i,\theta_\epsilon} = \frac{1}{W_i} \sum_{I_\epsilon \in \theta_\epsilon} D(I(p), I_\epsilon(p)) P(I(p)|\varphi_i)), \tag{3}$$

where, $I_\epsilon(p)$ is the pixel in neighborhood θ_ϵ, W_i is normalized, $D(I(p), I_\epsilon(p))$ is the smoothness parameter of neighborhood combining spatial coordinates and color coordinates. They are defined as follows:

$$D(I(p), I_\epsilon(p)) = exp(-\frac{(x - x_\epsilon)^2 + (y - y_\epsilon)^2}{\delta_d})exp(-\frac{|I(P) - I_\epsilon(P)|^2}{\delta_c}), \tag{4}$$

$$W_i = \sum_i \sum_{I_\epsilon(p) \in \theta_\epsilon} D(I(p), I_\epsilon(p)) P(I(p)|\varphi_i)), \tag{5}$$

In Eq. 4, (x, y) is the spatial position coordinate of pixel point $I(p)$, (x_ϵ, y_ϵ) is the spatial coordinate of the neighborhood pixel $I_\epsilon(P)$. θ_d, θ_c are the smoothness factor of coordinate space and color space.

Assume that the number of color category areas is K when GMM model is established, $I(p) \in \{I_k\}, k = 1, 2, \ldots K$. Then the probability that pixel point $I(p)$ belongs to the color category area represented by the current Gaussian component is:

$$P^{i,I(p)} = \frac{a_i P(I(p)|\varphi_i)}{\sum_{i=1}^{K} a_i P(I(p)|\varphi_i)} + P^{i,\theta_\epsilon}, \tag{6}$$

in Eq. (6), P^{i,θ_ϵ} is the probability that θ_ϵ of pixel point $I(p)$'s neighborhood belongs to color category i, when the pixel is located in the center of the region of color category i, its neighborhood pixels are most likely to have similar color characteristics with it. The probability of P^{i,θ_ϵ} will be higher, and the corresponding probability of $P^{i,I(p)}$ will also increase, that is, the probability that pixel $I(p)$ belongs to color category i will increase. On the contrary, if the pixel is located at the boundary of the color category area, there should be a relatively obvious difference between the neighboring pixel and it, and the corresponding probability will decrease. In other words, the probability that the pixel belongs to the color category i decreases, and the pixel can be evenly divided into multiple areas of different color categories. Therefore, P^{i,θ_ϵ} can ensure the uniformity and smoothness between different color categories.

M-Step: Change the Conditions for Merging the Gaussian Components

We use the $P^{i,I(p)}$ obtained from Eq. 6 in E-step to recalculate the parameters of each Gaussian component a_i, p_i, and Σ_i as follows:

$$a_i = \frac{\sum_i P^{i,I(p)}}{Z_t}, \tag{7}$$

$$u_i = \frac{\sum_i P^{i,I(p)} I(p)}{\sum_i P^{i,I(p)}}, \tag{8}$$

$$\Sigma_i = \frac{\sum_i P^{i,I(p)}(I(p) - \mu_i)(I(p) - \mu_i)^T}{\sum_i P^{i,I(p)}}. \tag{9}$$

In the classical EM algorithm, as long as two color category regions i and j satisfy $|u_i - u_j| < \delta$ (δ is a small enough value), the two regions are merged [25]. However, this merging method has some defects, and in some cases, the wrong merging processing may occur. For example, there are three category regions whose mean values are μ_1, μ_2 and μ_3 respectively. Given a threshold value of δ, and it satisfies $|\mu_1 - \mu_2| < \delta$, and $|\mu_1 - \mu_3| < \delta$, according to the algorithm, these three regions will be merged into a region. However, at this time, μ_2 and μ_3 do not necessarily satisfy $|\mu_2 - \mu_3| < \delta$, resulting in an error in clustering results. Therefore, the merged area should be re-estimated before the next merge operation. Specific improvement steps are as follows:

- Calculate the distance between each pair of color categories;
- Find the nearest color category region l to color category region i, that is, the distance between color category region i and l is the shortest;
- If the nearest color class l to the color class region i satisfies the condition $|u_i - u_l| < \delta$, then the two regions will be merged and the parameters will be re-estimated.

The termination condition of EM algorithm is that there is no merging of color category area in the current iteration, and all $P^{I,I(P)}$ in the current iteration is very little different from the last one.

In order to verify the effectiveness of the improved algorithm, this paper conducts a comparative experiment with k-means and traditional GMM color clustering algorithm. The experimental results are shown in Fig. 2. It can be clearly seen from the comparison that the region boundary obtained by k-means clustering is relatively "stiff", because only considering the color characteristics will separate the image content and easily produce wrong clustering results. For example, the white clouds in the picture cluster together with the sky, and

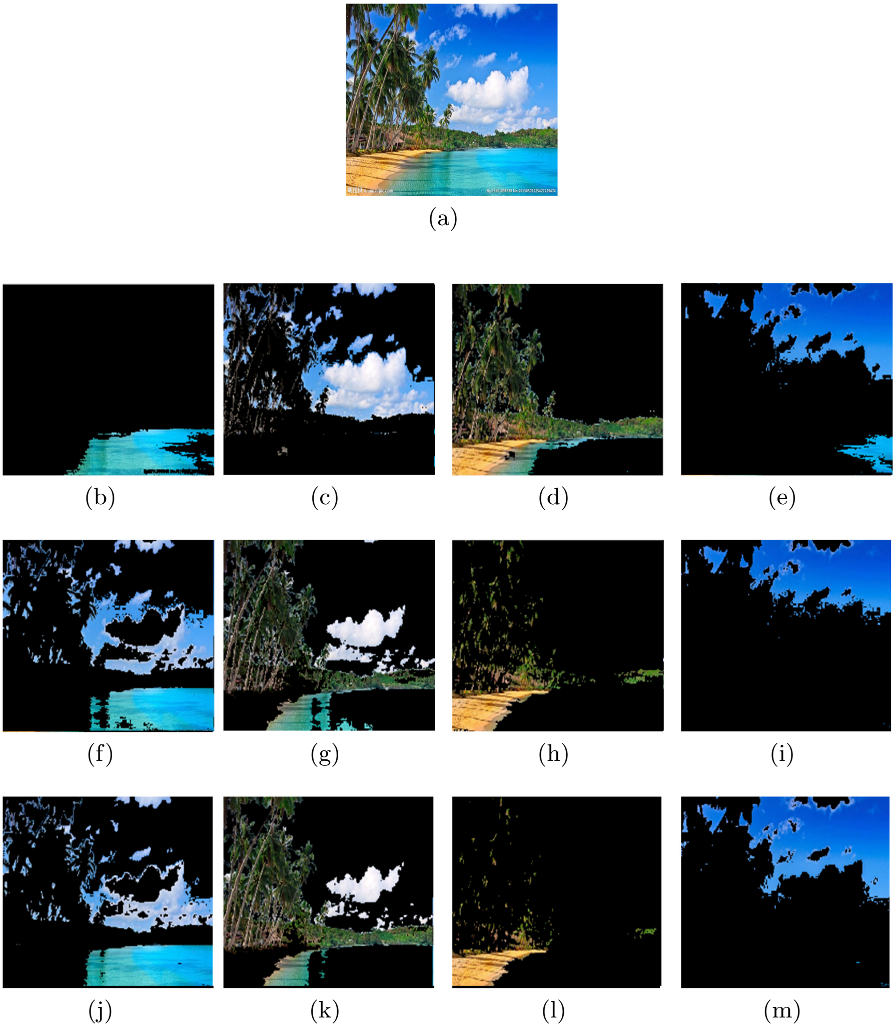


Fig. 2. Comparison of image color clustering results. (a) input image; (b)–(e) K-means clustering results; (f)–(i) Based on traditional GMM clustering results; (j)–(m) the clustering results of our proposed methods.

the sand and green trees cluster together. When the GMM model is used to cluster the image color areas, the regional boundary are relatively smooth. The clustering results of the improved algorithm in this paper are more accurate than those of the classical GMM model. For example, the sand and sky shown in the figure are better than those of the traditional GMM model.

3.2 Mapping and Color Transfer Between Color Category Regions

Mapping. After the image involved in the algorithm is segmented by clustering, each region of the input image will correspond to K candidate reference regions (assuming that the number of regions divided into reference images is K). Now, we will present how to find the best matching region from these candidate reference regions. We use the absolute value of the difference between the mean values of the two regions as the mapping function, which is expressed as follows:

$$f(in_i, ref_j) = |\mu_{in_i} - \mu_{ref_j}|, \tag{10}$$

where, in_i denote the ith input region, ref_j denote the jth candidate reference region. μ_{in_i} and μ_{ref_i} respectively represent the mean values of the input image color category region in_i and the reference image color category region $ref_j, j = 1, 2, \ldots k$. μ_{in_i} and μ_{ref_i} represent the color features of their respective regions to some extent, so we use $|\mu_{in_i} - \mu_{ref_j}|$ to represent the similarity between them. The smaller the difference, the higher the similarity between the two regions, then the optimal matching formula between regions can be expressed as follows:

$$(in_i, ref_j) \leftarrow min\{f(in_i, ref_j)|1 \leq i \leq K, 1 \leq j \leq K\}, \tag{11}$$

here, ref_j is the optimal matching region for the input image color category region in_i.

Color Transfer. In the traditional color transfer method proposed by Reinhard et al., only the global mean and standard deviation of two images are used for color transfer. Its equation is as follows:

$$O_i = \frac{\delta_i^T}{\delta_i^I}(I_i(x,y) - \mu_i^I) + \mu_i^T, \tag{12}$$

$\mu_i^I, \mu_i^T, \delta_i^I, \delta_i^T$ represent the mean value and standard deviation of the ith channel of the two images, and O_i represents the output image of the ith channel. The essence of this method is to modify the color content of the input image by using the first-order statistics of the reference image. It is mainly used for color transfer between two images under normal light conditions to change the image style.

There are significant differences in application scenarios and implementations between our approach and those proposed by Reinhard et al. Our method is to cluster and segment the image according to the color category and carry out color transfer between the best matched local regions to enhance the low-illumination image. As mentioned above, we used RLAB color perception space to achieve color transfer. Specifically, we first transferred the eigenvalue of the color components of the optimal matching reference color region corresponding to the input image region to the resulting image. Then, in order to maintain the continuous and uniform brightness component of the resulting image, the transfer of the brightness component is an overall migration. Equations 13–15 show the process of transition.

$$A_O^R = \sum_{i=1}^{k} P^{(i,I(p))} \times (\frac{\delta^{A^R}_{ref_i}}{\delta^{A^R}_{in_i}}(A^R_{in_i} - \mu^{A^R}_{in_i}) + \mu^{A^R}_{ref_i}), \tag{13}$$

$$B_O^R = \sum_{i=1}^{k} P^{(i,I(p))} \times (\frac{\delta^{B^R}_{ref_i}}{\delta^{B^R}_{in_i}}(B^R_{in_i} - \mu^{B^R}_{in_i}) + \mu^{B^R}_{ref_i}), \tag{14}$$

$$L_O^R = \frac{\delta^{L^R}_{ref}}{\delta^{L^R}_{in}} \times (L^R_{in} - \mu^{L^R}_{in}) + \mu^{L^R}_{ref}, \tag{15}$$

where, A_O^R, B_O^R, L_O^R are the three channel values of the RLAB color space of the resulting image. μ and δ are mean and standard deviation, Subscripts in_i and ref_i represent the ith region, Superscripts L^R, A^R, B^R represent the color components L, A and B of the image in RLAB space. $P^{(i,I(p))}$refers to the membership probability of pixel $I(p)$ belonging to the input image color category region i. it can be used to evenly distribute colors to all regions of the resulting image.

4 Experimental Results and Analyses

In order to evaluate the performance of our method subjectively and objectively, we used real low-light images for comparison experiments under different scenes. All the images were from low-light image dataset LOL, which contained 789 images of various low-light images and 789 corresponding ground truth images. Noting that LOL is collected by us with an off-the-shelf camera to accomplish this task. Limited by space, this paper only provides experimental results of 5 randomly selected low-light images. For convenience, we call them images (a) to image (e) as illustrated in Fig. 3. Additionally, we randomly chose a higher quality image from the data set LOL as a reference image, as shown in Fig. 4.

(a) (b) (c) (d) (e)

Fig. 3. Original low-light images: (a)–(e).

We compare the results of proposed method with some baseline methods and the state-of-the-art algorithms. These methods include histogram equalization (HE) [13], adaptive histogram equalization (CLAHE) [14], gamma correction (GC) [18], adaptive gamma correction contrast enhancement (AGCWD) [9], ying_2017 model-based image enhancement [26]. All the algorithm codes involved

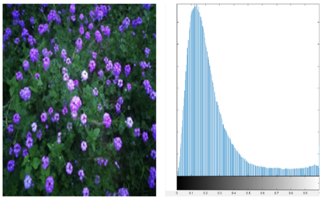


Fig. 4. The reference image used in our proposed method and its corresponding histogram

in our experiment were completed on the MATLAB 2018 simulation platform. As all existing low-illumination image enhancement algorithms inevitably present noise problems, to reduce the impact of noise on other algorithms, all enhanced images are processed by BM3D denoising technology. Figure 5 shows the enhancement and contrast effects of different algorithms for randomly selecting 5 low-illumination images (image (a)–image (e)) from the data set LOL.

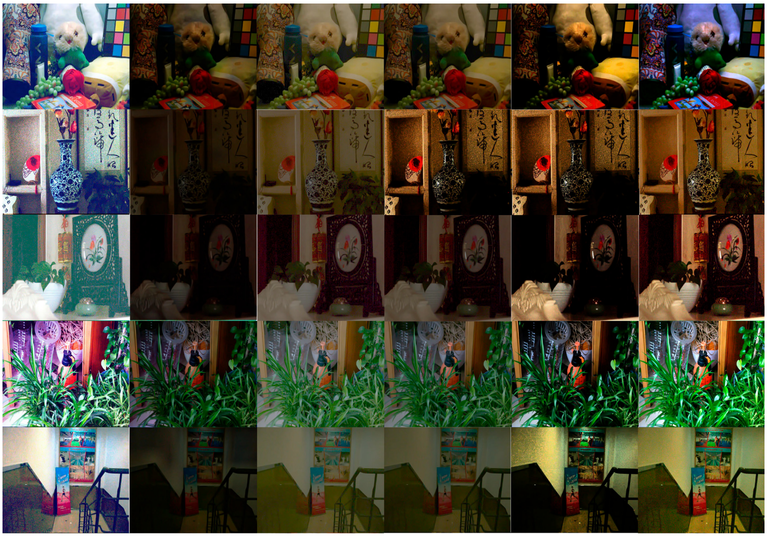


Fig. 5. Comparison of the results of different enhancement algorithms for five dark images. Each row represents all enhancement of a dark image, from the first column to the last column: HE,CLAHE,GC,Ying_2017, AGCWD and OURS.

In the subjective evaluation, we took the image enhancement results shown in Fig. 5 as a case for comparative analysis. As can be seen from the Fig. 5, HE algorithm can enlarge the dynamic range of image and improve the overall contrast. However, oversaturation is easy to occur, resulting in serious color deviation phenomenon, such as the overall color of the image white. GC ($\gamma = 0.4$)

adjusts the overall brightness of the image by exponential function. Although the brightness increases, the contrast effect does not improve significantly. The reason is that when GC enhances the image, it also inevitably amplifies the noise, thus reducing the visual quality of the image. The CLAHE enhancement algorithm can effectively reduce the excessive enhancement of brightness and highlight the target object, but it ignores the details of other areas, especially the overall visual effect of background image is dark. The Ying_2017 algorithm has a better effect on the darker areas of the image, making the color more vivid. But, this algorithm is prone to over-enhancement and will cause some color distortion. The contrast enhancement effect of AGCWD is significant, but it generally produces halo artifacts around the brighter areas. From the contrast effect, our method not only improves the overall brightness of the image, but also the visual quality of the image is better than other methods. The enhanced image has higher contrast, clearer dark area texture, and better retention of target edges and details.

In the objective evaluation of low-illumination images, due to the known Ground truth, the difference between images enhanced by different algorithms and Ground truth can be compared to verify the performance of the algorithm. In this experiment, we comparatively evaluate the images enhanced by four criteria: peak signal-to-noise ratio (PSNR) [3], structural similarity (SSIM) [20], mean square error (MSE) [3] and Entropy [19]. Tables 1, 2, 3 and 4 show the quantitative results of various evaluation criteria.

Table 1. PSNR value comparison of different methods

PSNR	HE	CLAHE	GC	Ying-2017	AGCWD	Proposed Method
Image (a)	15.4866	9.4859	14.1683	11.5077	10.8542	17.1476
Image (b)	11.9360	10.7571	15.8635	13.4173	14.0445	18.0050
Image (c)	11.9446	16.2274	24.9204	20.5176	16.8318	21.2795
Image (d)	10.5407	5.8268	8.5511	7.2020	7.5113	11.5951
Image (e)	13.7143	9.9674	17.0723	13.5277	13.4373	14.9689
Average	13.7359	10.4529	16.1151	13.2944	12.5340	16.5992

As can be seen from Table 1, the PSNR value obtained by our method is higher than that of other methods in most cases. This result indicates the quality of the enhanced images obtained by our method is better than other methods, which is consistent with the previous subjective evaluation results. As illustrated in Table 2 and Table 3, similar to PSNR, SSIM and ENTROPY also maintain high values relative to other algorithms. As can be seen from Table 2, the SSIM value of our method is significantly higher than that of other enhancement methods. In Table 3, we see that our ENTROPY values are the highest in most cases, this directly indicates that the method we proposed can extract more information from the dark image and recover more details and texture information.

Table 2. SSIM value comparison of different methods

SSIM	HE	CLAHE	GC	Ying-2017	AGCWD	Proposed Method
Image (a)	0.3434	0.3423	0.6605	0.5419	0.3212	0.6567
Image (b)	0.3169	0.2411	0.4942	0.3980	0.1975	0.7047
Image (c)	0.5225	0.6596	0.7048	0.8199	0.6146	0.7886
Image (d)	0.1843	0.1571	0.4459	0.3303	0.2179	0.6422
Image (e)	0.2897	0.2829	0.5710	0.6209	0.3274	0.6138
Average	0.3314	0.3366	0.5753	0.5368	0.3357	0.6812

Table 3. ENTROPY value comparison of different methods

Entropy	HE	CLAHE	GC	Ying-2017	AGCWD	Proposed Method
Image (a)	7.8481	6.2915	7.1345	6.7977	6.4974	7.9320
Image (b)	7.5656	5.3329	6.6157	6.2031	5.9904	7.8406
Image (c)	7.9644	6.7435	7.0384	6.8934	6.9913	8.8625
Image (d)	7.8901	5.5097	6.4931	6.2090	6.6937	7.4987
Image (e)	7.9209	5.6912	6.5408	6.3203	7.0234	7.9575
Average	7.8378	5.9138	6.7645	6.4847	6.6392	8.0183

Table 4. MSE value comparison of different methods

MSE	HE	CLAHE	GC	Ying-2017	AGCWD	Proposed Method
Image (a)	0.0283	0.1126	0.0383	0.0707	0.0823	0.0610
Image (b)	0.0623	0.0840	0.0259	0.0455	0.0394	0.0158
Image (c)	0.0639	0.0238	0.0320	0.0089	0.0207	0.0118
Image (d)	0.0279	0.2614	0.1396	0.1905	0.1774	0.0693
Image (e)	0.0425	0.1008	0.0196	0.0444	0.0453	0.0319
Average	0.0452	0.1165	0.0511	0.0720	0.0730	0.0380

Table 4 shows the metric value of MSE. Through observation, it is found that the MSE value obtained by our method is lower than others, which indicates that enhanced results obtained by our proposed method are very close to the ground truth. Through comparative analysis, it can be known that our method has excellent performance in PSNR, SSIM, MSE and ENTROPY.

5 Conclusion

In this paper, we propose a color transfer method based on local statistical features of images to enhance low-light images into recognizable images. In this method, the input image and reference image are clustered and segmented in

RALB color space by improved Gaussian mixture model. The optimal matching region of different color category space is found through matching mapping function adaptively to realize color transfer. Experimental results show that the enhancement method based on color transfer can not only improve the brightness of the image, but also keep the chromaticity and detail characteristics. Meanwhile it can reduce the color distortion and artifact, and achieve better enhancement effect compared with other algorithms. In the future, we will study the color transfer method of multi-reference graph pattern, which is used to solve the problem of optimal color region matching under the complex image content.

References

1. Arriaga-Garcia, E.F., Sánchez-Yáñez, R.E., Ruiz-Pinales, J., de Guadalupe García-Hernández, M.: Adaptive sigmoid function bihistogram equalization for image contrast enhancement. J. Electron. Imaging **24**(5), 053009 (2015)
2. Cepeda-Negrete, J., Sánchez-Yáñez, R.E., Correa-Tome, F.E., Lizarraga-Morales, R.A.: Dark image enhancement using perceptual color transfer. IEEE Access **6**, 14935–14945 (2018)
3. Chandler, D.M., Hemami, S.S.: VSNR: a wavelet-based visual signal-to-noise ratio for natural images. IEEE Trans. Image Process. **16**(9), 2284–2298 (2007)
4. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B-Methodol. **39**(1), 1–22 (1977)
5. Dong, X., et al.: Fast efficient algorithm for enhancement of low lighting video. In: Proceedings of the 2011 IEEE International Conference on Multimedia and Expo, ICME 2011, Barcelona, Catalonia, Spain, 11–15 July 2011, pp. 1–6. IEEE Computer Society (2011)
6. Ginesu, G., Giusto, D.D., Märgner, V., Meinlschmidt, P.: Detection of foreign bodies in food by thermal image processing. IEEE Trans. Ind. Electron. **51**(2), 480–490 (2004)
7. Havasi, L., Szlávik, Z., Szirányi, T.: Detection of gait characteristics for scene registration in video surveillance system. IEEE Trans. Image Process. **16**(2), 503–510 (2007)
8. He, L., Qi, H., Zaretzki, R.: Image color transfer to evoke different emotions based on color combinations. SIViP **9**(8), 1965–1973 (2014). https://doi.org/10.1007/s11760-014-0691-y
9. Huang, S., Cheng, F., Chiu, Y.: Efficient contrast enhancement using adaptive gamma correction with weighting distribution. IEEE Trans. Image Process. **22**(3), 1032–1041 (2013)
10. Hussein, A.A., Yang, X.: Colorization using edge-preserving smoothing filter. SIViP **8**(8), 1681–1689 (2012). https://doi.org/10.1007/s11760-012-0402-5
11. Kim, Y.T.: Contrast enhancement using brightness preserving bi-histogram equalization. IEEE Trans. Consum. Electron. **43**(1), 1–8 (1997)
12. Li, L., Wang, R., Wang, W., Gao, W.: A low-light image enhancement method for both denoising and contrast enlarging. In: 2015 IEEE International Conference on Image Processing, ICIP 2015, Quebec City, QC, Canada, 27–30 September 2015, pp. 3730–3734. IEEE (2015)
13. Narendra, P.M., Fitch, R.C.: Real-time adaptive contrast enhancement. IEEE Trans. Pattern Anal. Mach. Intell. **3**(6), 655–661 (1981)

14. Pisano, E.D., et al.: Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms. J. Digital Imaging **11**(4), 193–200 (1998)
15. Pizer, S.M., Amburn, E.P., Austin, J.D., Cromartie, R., Zuiderveld, K.: Adaptive histogram equalization and its variations. Comput. Vis. Graph. Image Process. **39**(3), 355–368 (1987)
16. Rasti, P., Daneshmand, M., Alisinanoglu, F., Ozcinar, C., Anbarjafari, G.: Medical image illumination enhancement and sharpening by using stationary wavelet transform. In: 24th Signal Processing and Communication Application Conference, SIU 2016, Zonguldak, Turkey, 16–19 May 2016, pp. 153–156. IEEE (2016)
17. Reinhard, E., Ashikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. IEEE Comput. Graph. Appl. **21**(5), 34–41 (2001)
18. Scott, J., Pusateri, M.A.: Towards real-time hardware gamma correction for dynamic contrast enhancement. In: 2009 IEEE Applied Imagery Pattern Recognition Workshop, AIPR 2009, Washington, DC, USA, 14–16 October 2009, pp. 1–5. IEEE Computer Society (2009)
19. Wang, P., Wang, J., Paranamana, P., Shafto, P.: A mathematical theory of cooperative communication. CoRR abs/1910.02822 (2019)
20. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
21. Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to greyscale images. ACM Trans. Graph. **21**(3), 277–280 (2002)
22. Xiao, X., Ma, L.: Color transfer in correlated color space. In: Sun, H. (ed.) Proceedings VRCIA 2006 ACM International Conference on Virtual Reality Continuum and its Applications, Chinese University of Hong Kong, Hong Kong, China, 14–17 June 2006, pp. 305–309. ACM (2006)
23. Xie, X., Lam, K.: Face recognition under varying illumination based on a 2D face shape model. Pattern Recogn. **38**(2), 221–230 (2005)
24. Yang, X., Wang, B., Yang, K., Liu, C., Zheng, B.: A novel representation and compression for queries on trajectories in road networks. IEEE Trans. Knowl. Data Eng. **30**(4), 613–629 (2018)
25. Yang, X., Wang, Y., Wang, B., Wang, W.: Local filtering: improving the performance of approximate queries on string collections. In: Sellis, T.K., Davidson, S.B., Ives, Z.G. (eds.) Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, 31 May–4 June 2015, pp. 377–392. ACM (2015)
26. Ying, Z., Li, G., Ren, Y., Wang, R., Wang, W.: A new image contrast enhancement algorithm using exposure fusion framework. In: Felsberg, M., Heyden, A., Krüger, N. (eds.) CAIP 2017. LNCS, vol. 10425, pp. 36–46. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64698-5_4
27. Yu, Z., Bajaj, C.L.: A fast and adaptive method for image contrast enhancement. In: Proceedings of the 2004 International Conference on Image Processing, ICIP 2004, Singapore, 24–27 October 2004, pp. 1001–1004. IEEE (2004)

Role-Aware Enhanced Matching Network for Multi-turn Response Selection in Customer Service Chatbots

Guangxuan Zhao, Ying Zhu, Shi Feng(✉), Daling Wang, Yifei Zhang, and Ge Yu

School of Computer Science and Engineering, Northeastern University, Shenyang, China
zhaoguangxuan_neu@163.com, zhuying_neu@163.com, {fengshi, wangdaling,zhangyifei,yuge}@cse.neu.edu.cn

Abstract. We study on the response selection problem for multi-turn conversation in retrieval-based customer service chatbots. Existing multi-turn context-response matching models do not consider the effect of speaker's role on utterance semantics. In this paper, we propose a **R**ole-aware **E**nhanced **M**atching network (REM) to distinguish utterances from the perspective of speakers' roles and enrich the semantic features of context with role-aware enhancement. First, the utterances are encoded by different GRUs according to speakers. Then an attention mechanism and an interaction function are employed between two speakers' utterances to enrich the semantics of context, followed by constructing matching matrices and aggregation. Extensive experiments are conducted on public available e-commerce dialogue dataset and the results show that our proposed model outperforms strong baseline methods by large margins.

Keywords: Multi-turn conversation · Role-aware Enhanced Matching · Customer service chatbots

1 Introduction

As an important kind of human-computer dialog systems, the chatbots aim to converse with humans naturally and meaningfully. Researchers have shown great interests in open domain multi-turn chatbots. In general, chatbots can be categorized into generation-based and retrieval-based according to their implementation methods. Retrieval-based chatbots select a proper response for the current conversation from a ready-made corpus, while generation-based ones generate the responses word by word through an encoder-decoder framework. Retrieval-based chatbots learn a matching model for a pair of a conversational context and a response candidate. Generation-based chatbots can incorporate rich backgrounds when mapping between successive turns of dialogue. In comparison, retrieval-based chatbots enjoy the advantage of informative and fluent responses, and they also have advantages in the syntactic correctness, the semantic diversity, and the length of the responses. As a consequence, retrieval-based chatbots are extensively applied in commerce such as XiaoIce (Shum et al. [1]) implemented by Microsoft and E-commerce assistant AliMe (Li et al. [2]) serving on Taobao.

X. Yang et al. (Eds.): ADMA 2020, LNAI 12447, pp. 654–668, 2020.
https://doi.org/10.1007/978-3-030-65390-3_49

Word and sentence embedding techniques are important for retrieval based response selection. Context and response must be projected properly into the vector space to capture the relationship between them, which is critical to the subsequent process. Another key technique for response selection tasks is context-to-response matching, which builds semantic matching model between two utterances.

Recently, retrieval-based multi-turn conversation response selection for customer service in e-commerce has received great attentions from both academic and industrial communities [3, 4]. Comparing with single-turn conversation which only considers the last utterance [5, 6], multi-turn conversation selects response using all the context utterances, which brings in more relevant information as well as noise. Early studies matched responses with context as a whole. In recent years, some methods have been explored for modeling the context at finer-grained level. Wu et al. [7] proposed the sequential matching network to match the response with each utterance in the thread respectively. Zhang et al. [8] employed the last context utterance to aggregate with all the other utterances for distilling the pivotal information. Yuan et al. [9] proposed a multi-hop selector network to alleviate the problem of the side effect of using too many context utterances. These models give a more comprehensive understanding of the context and can retain the information of each turn. Although the previous literature have achieved promising results, most of these methods treat the context utterances equally. However, the influence of different speakers' roles among the context, which is important especially in customer service domain, has not yet been considered in the existing multi-turn conversation models.

Table 1 gives a motivating dialogue example in e-commerce customer service. A customer consults customer service (agent for short) about the difference between two products firstly. After agent has answered, the customer want agent to give recommendations without necessary information (the usage of headphone), so agent makes an inquiry to customer. In the last context utterance, customer makes a supplement to the inquiry, so a proper response should be a final recommendation according to the whole dialogue context. We observe that the customer and agent have obviously different roles in customer service dialogues. The customer is usually well-motivated and leads the conversations. On the other hand, the agent attempts to reply appropriate information that the customer wants to know. For multi-turn conversations, the customer side contains more informative and critical contents, and thus tends to lead the conversation. This observation, however, is neglected by previous studies.

To tackle these challenges, in this paper, we propose a novel neural network model, called the **R**ole-aware **E**nhanced **M**atching network (dubbed as REM), for retrieval-based multi-turn customer service response selection. Different from the existing literature, we consider the roles of speakers in the conversation, and model the customer and agent historical utterances separately. Moreover, REM leverages attention mechanisms to enhance customer utterances' representations by all the agent utterances. The basic intuition behind this operation is that the customer should be paid more attentions for customer service response selection task. The learned enhanced representations are fed into a matching-aggregation framework to calculate the matching scores.

Table 1. Dialogue example in e-commerce customer service.

Speaker	Utterance
Customer	What's the main **difference** between the professional and competitive versions of your **headphones**?
Agent	The main **difference** is in the microphone unit and the audio unit
Customer	**Which is better?**
Agent	It depends on your **use**
Customer	I mainly use it to **watch dramas** and occasionally **play games**
Response	Competitive version is more suitable for you

To sum up, the main contributions of this paper are as follows:

- We propose a novel model called **R**ole-aware **E**nhanced **M**atching network (REM) to select response in customer service. REM follows representation-matching-aggregation framework and could match the candidate response with historical utterances at different level.
- We leverage attention mechanism to provide a role-aware enhanced representations of the historical utterances. To the best of our knowledge, we are the first to consider different roles of speakers in the e-commerce response matching task.
- We conduct extensive experiments on benchmark dataset, and the results show that our REM model reduces utterance matching errors by role-aware mechanism and outperforms strong baseline methods by large margins.

2 Related Work

Response selection is an important retrieval-based task for chatbots, whose purpose is to select the most suitable response from a group of candidates given the context of a conversation. Previous studies on retrieval-based chatbots focus on single-turn conversation's response selection, which takes the context of only the last utterance into consideration [10, 11]. Recently, multi-turn conversation's response selection has attracted more attentions due to its practical value. The existing work on multi-turn response selection can be categorized into two types that are representation-based and interaction-based models. The first type of methods concatenate all utterances of context as a single long sentence, and match candidate responses with the long sentence. Lowe et al. [12] spliced utterances into one sentence and then encode it with LSTM. Chen et al. [13] concatenated all utterances of context and conducted matching with enhanced sequential inference model (ESIM) [14].

The second type of existing methods employ a matching-aggregation framework that matches candidate response with each utterance in context, and then aggregates the matching features. Wu et al. proposed the sequential matching network (SMN) [7], which is also one of the important baseline of our model. SMN model presented each of the context and candidate responses as a word vector, and then, the word embedding containing contextual information was obtained through GRU. A series of matching matrices were obtained by calculating the inner product of context and responses. Next, the matching representation of each group was learned by conducting convolution, pooling and flattening operation on the matrices. Finally, SMN aggregated matching vectors through another GRU, and calculated the final matching score through MLP. Zhang et al. [8] considered the interaction between utterance and response and used the last context utterance to aggregate with all the other utterances for distilling the pivotal information from utterance embeddings. Yuan et al. [9] utilized a multi-hop selector to select the relevant utterances as context and matched the filtered context with the candidate response. Zhou et al. [15] adopted self-attention mechanism to obtain multi-grained representations, and aggregated features through 3D convolutions. Tao et al. [16] fused multiple types of representations for context-response matching, and utilized three strategies for representation fusion. Other studies have showed that the utterances in the context are of different levels of importance for response matching. Gu et al. [17] improved SMN model by enhancing sentence representations through an attentive hierarchical recurrent encoder and capturing bidirectional interactions between contexts and responses.

Although the existing multi-turn response selection methods have achieved promising results, these models fail to distinct different roles of the speakers in the conversations, which is important in customer service dialogues.

3 Role-Aware Enhanced Matching Network

3.1 Problem Formalization

Each conversation in multi-turn customer service response selection task can be described as a triple $\langle context, r, y\rangle$, where $context = \left\{c_1, s_1, \ldots, c_{N_c}\right\}$, each c_i denotes a customer utterance while each s_j denotes a service (agent for short) utterance. Moreover, r is a response candidate of agent depending on the context and $y \in \{0, 1\}$ is a binary label. We need to build a model to indicate whether r is a proper response for the *context*. Our goal is to learn a discriminator $g(\cdot, \cdot)$ with $\langle context, r, y\rangle$. For any context-response pair, the function $g(context, r)$ measures the matching degree of the pair.

3.2 Model Overview

Figure 1 and Fig. 2 give the two phases of REM architecture respectively, which generally follows the **representation-matching-aggregation** framework to match response with multi-turn context.

In representation phase shown in Fig. 1, for each utterance c_i, s_j in context and its response candidate r, REM first looks up a shared word embedding table and represents them as sequences of word embedding. Then the utterances and response are encoded with two encoders (such as recurrent neural networks) according to its speaker's role, namely customer or agent. Thus two types of contextualized word representation are obtained, one is for customer and the other is for agent. The two encoders use the same structure, but their parameters are not shared. Furthermore, REM utilizes all agent utterances (excluding response) to enhance customer utterances' representations through the

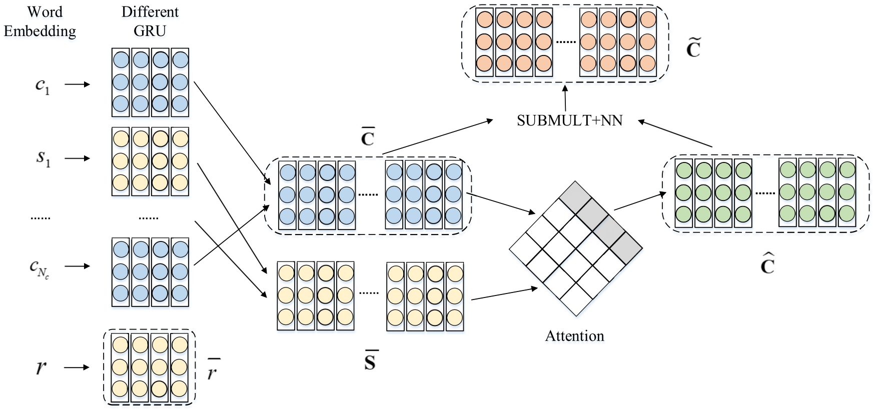


Fig. 1. Representation phase of role-aware enhanced matching network

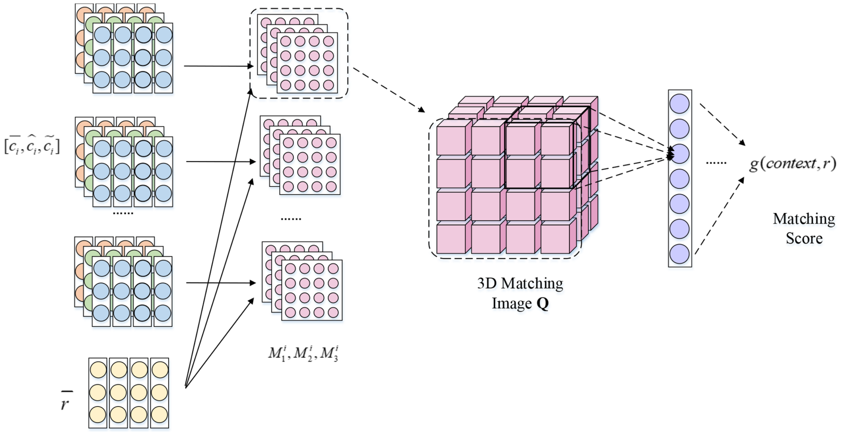


Fig. 2. Matching-aggregation phase of role-aware enhanced matching network

attention mechanism and an interaction function. After obtaining the new representation, each enhanced customer utterance representation is fed into the next phase and matched with the response.

In matching-aggregation phase of Fig. 2, we get n matching matrix (n is the number of utterances of the customer) by calculating the inner product of customer utterance and candidate response representation vector. The matching features are aggregated following the chronological order of the utterances in the context with convolution-pooling operation, and REM finally calculates the matching score of the context-response pair by MLP. We will elaborate the modules of REM in the following sections.

3.3 Utterance Representation

With a word embedding table, REM represents each utterances in context and response as $c_i = \left[e^c_{i,1}, e^c_{i,2}, \ldots, e^c_{i,n}\right]$, $s_j = \left[e^s_{j,1}, e^s_{j,2}, \ldots, e^s_{j,n}\right]$, $r = \left[e^r_1, e^r_2, \ldots, e^r_n\right]$, where $e^c_{i,k}, e^s_{j,k}, e^r_k \in \mathbb{R}^d$ are the embeddings of the k^{th} word of c_i, s_j and r respectively. Then all $c_i \in \mathbb{R}^{d\times n}$ are encoded with a gated recurrent unit (GRU) to transform word embeddings into contextual word representations, because GRU can propagate information along the word sequence. Suppose that $\overline{c_i} = \left[h^c_{i,1}, h^c_{i,2}, \ldots, h^c_{i,n}\right]$, where $h^c_{i,k}$ is the hidden vector of $e^c_{i,k}$, then $h^c_{i,k} \in \mathbb{R}^m$ is defined by

$$\begin{aligned} z_k &= \sigma\left(\mathbf{W_z} e^c_{i,k} + \mathbf{V_z} h^c_{i,k-1}\right) \\ r_k &= \sigma\left(\mathbf{W_r} e^c_{i,k} + \mathbf{V_z} h^c_{i,k-1}\right) \\ \tilde{h}^c_{i,k} &= tanh\left(\mathbf{W_h} e^c_{i,k} + \mathbf{V_h}\left(r_k \odot h^c_{i,k-1}\right)\right) \\ h^c_{i,k} &= z_k \odot \tilde{h}^c_{i,k} + (1 - z_k) \odot h^c_{i,k-1} \end{aligned} \tag{1}$$

where z_k and r_k are update gate and reset gate respectively, $\sigma(\cdot)$ is a sigmoid function, and $\mathbf{W}_z$, $\mathbf{W_r}$, $\mathbf{W_h}$, $\mathbf{V_z}$, $\mathbf{V_z}$, $\mathbf{V_h}$ are learned parameters. Similarly, for the agent's utterances and response, including all $s_j \in \mathbb{R}^{d\times n}$ and $r \in \mathbb{R}^{d\times n}$, we encode them with another GRU, whose architecture is same as Eq. (1), but uses different parameter values. We will show the reason why the parameter values of the two GRUs should be different and what is the effect if their parameter values are the same in the following experiment. In this way we will obtain $\overline{s_j} = \left[h^s_{j,1}, h^s_{j,2}, \ldots, h^s_{j,n}\right]$ and $\overline{r} = \left[h^r_1, h^r_2, \ldots, h^r_n\right]$.

3.4 Role-Aware Enhanced Representation

Based on our observation, the customer's intention leads the trend of dialogue in most common cases. Thus, we should pay more attention on the customer's utterances. REM uses all the agent context utterances' representation to enhance each customer utterance, enrich their semantic information and highlight the keywords for the matching task in conversation.

Firstly, we adopt an attention mechanism to distill pivotal information from agent utterance in context. Given the set of utterance representations $\bar{c}_l$ encoded by the sentence encoder, we concatenate them to form the customer words representation $\bar{C} = \{\bar{c}_l\}_{i=1}^{N_c}$. Also, the agent context representation $\bar{S} = \{\bar{s}_J\}_{j=1}^{N_s}$ is formed similarly by concatenating $\bar{s}_j$. After the work above, every customer utterance can be enhanced by the agent's utterances.

$$\bar{\mathbf{C}} = Concatenate(\{\overline{c_i}\}_{i=1}^{N_c}) \tag{2}$$

$$\bar{\mathbf{S}} = Concatenate(\{\overline{s_j}\}_{j=1}^{N_s}) \tag{3}$$

where N_c is the number of customer utterance. The cross attention representation $\widehat{c_i} = \left[o^c_{i,1}, o^c_{i,2}, \ldots, o^c_{i,n}\right]$, $o^c_{i,k} \in \mathbb{R}^m$ is defined by

$$Att\left(\bar{\mathbf{C}}, \bar{\mathbf{S}}\right) = [softmax\left(\bar{\mathbf{C}}[i] \cdot \bar{\mathbf{S}}^{\mathrm{T}}\right)]_{i=0}^{N_c \cdot n-1} \tag{4}$$

$$\hat{\mathbf{C}} = Att\left(\bar{\mathbf{C}}, \bar{\mathbf{S}}\right) \cdot \bar{\mathbf{S}} \tag{5}$$

$$\{\widehat{c_i}\}_{i=1}^{N_c} = Separate\left(\hat{\mathbf{C}}\right) \tag{6}$$

where $\bar{\mathbf{C}}[i]$ is the i^{th} word's hidden vector of concatenated customer utterances. Each row of $\hat{\mathbf{C}}$, namely $\hat{\mathbf{C}}[i]$, shows how closely customer utterances and agent utterances connect, and stores the fused semantic information of words in all the agent context utterances that may have dependencies to $\bar{\mathbf{C}}[i]$.

Finally, $\bar{\mathbf{C}}$ and $\hat{\mathbf{C}}$ are fed to an interaction function. We compared several interaction functions including NEURAL NET, NEURAL TENSOR NET, SUBTRACTION and *SUBMULT* + *NN* function [18]. After the comparison of the effect of the functions above, we find out that *SUBMULT* + *NN* function can get the best results. Thus, we choose to use *SUBMULT* + *NN* function, which has also been proven effective in various tasks, to form another feature $\widetilde{c_i} = \left[t^c_{i,1}, t^c_{i,2}, \ldots, t^c_{i,n}\right], t^c_{i,k} \in \mathbb{R}^m$ for each customer word. This function concatenates two vectors' element-wise multiplication and square of element-wise subtraction, then implicitly captures their distance and angle information through a one-layer perceptron with a ReLU activation.

$$\tilde{\mathbf{C}} = ReLU(\mathbf{W_p}\begin{bmatrix}\left(\bar{\mathbf{C}}[i] - \hat{\mathbf{C}}[i]\right) \odot \left(\bar{\mathbf{C}}[i] - \hat{\mathbf{C}}[i]\right) \\ \bar{\mathbf{C}}[i] \odot \hat{\mathbf{C}}[i]\end{bmatrix} + \mathbf{b_p})_{i=0}^{N_c \cdot n-1} \tag{7}$$

$$\{\widetilde{c_i}\}_{i=1}^{N_c} = Separate\left(\tilde{\mathbf{C}}\right) \tag{8}$$

where $\odot$ refers to element-wise multiplication, and $\mathbf{W_p} \in \mathbb{R}^{m\times 2m}$ and $\mathbf{b_p} \in \mathbb{R}^m$ are parameters.

Thus far, for k^{th} word in i^{th} customer utterance, we have 3 vectors $h^c_{i,k}, o^c_{i,k}, t^c_{i,k}$. Here $h^c_{i,k}$ represents contextual word information, $o^c_{i,k}$ contains agent context words' dependencies, and $t^c_{i,k}$ encodes their relationship. With these features, REM model could discriminate the important information for matching the response.

Notice that we obtain $o^c_{i,k}$ and $t^c_{i,k}$ features only for one role (customer), while discard another role's utterances (agent) in context, because their information has already been fused into the representations above-mentioned. Thus, we call these representations as role-aware enhanced representation.

3.5 Response Matching

With role-aware enhanced representation, a customer utterance can be denoted as $\left[\overline{c_i}, \widehat{c_i}, \widetilde{c_i}\right]$. Similar to the architecture of SMN model, we firstly match each utterance with response and then aggregate them. Remember that we get $\bar{r} = \left[h^r_1, h^r_2, \ldots, h^r_n\right]$ in Sect. 3.3 as response's representation. For each customer utterance-response pair, three matching matrices (denoted as M_1, M_2, M_3) are constructed. REM model performs best when all the three matching matrices are utilized. Any of the three matching matrices should not be removed, and we will validate the effectiveness of each matrix in the ablation experiments. For i^{th} pair, the $(j, k)^{th}$ element of M_1^i, M_2^i, M_3^i is respectively defined by

$$e_{1,j,k}^{(i)} = h_{i,j}^{c\ \ \mathrm{T}} \mathbf{A}_1 h_k^r \tag{9}$$

$$e_{2,j,k}^{(i)} = o_{i,j}^{c\ \ \mathrm{T}} \mathbf{A}_2 h_k^r \tag{10}$$

$$e_{3,j,k}^{(i)} = t_{i,j}^{c\ \ \mathrm{T}} \mathbf{A}_3 h_k^r \tag{11}$$

where $h_{i,j}^c$ denotes the j^{th} word of i^{th} customer utterance, h_k^r denotes the k^{th} word of response, and $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3 \in \mathbb{R}^{m \times m}$ are linear transformations. These three matching matrices implicitly contain semantic matching patterns of a customer utterance-response pair.

3.6 Aggregation

Finally, we aggregate all the M_1, M_2, M_3 into a 3D matching image $\mathbf{Q}$ [15], which is defined as

$$\mathbf{Q} = \{\mathbb{Q}_{i,j,k}\}_{N_c \times n \times n} \tag{12}$$

$$\mathbb{Q}_{i,j,k} = \left[e_{1,j,k}^{(i)}; e_{2,j,k}^{(i)}; e_{3,j,k}^{(i)} \right]$$

where each pixel $\mathbb{Q}_{i,j,k}$ is a concatenation of $e_{1,j,k}^{(i)}$, $e_{2,j,k}^{(i)}$, $e_{3,j,k}^{(i)}$, which means each pixel has three channels. Then we leverage a 2-layer 3D convolution with max-pooling operations to distill important matching features from the whole image. The operation of 3D convolution with max-pooling is the extension of typical 2D convolution, whose filters and strides are 3D cubes.

By flatten the second max-pooling operation's output, we can obtain a matching feature vector v. Finally, for the matching feature vector v, it is sent into a multi-layer perceptron (MLP) classifier. Here, the MLP is designed to predict whether a <context, r, y> triple match appropriately based on the derived matching feature vector and return a score denoting the matching degree. We compute the matching score $g(context, r)$ via a single-layer perceptron, which is formulated as:

$$g(context, r) = \sigma(\mathbf{W_v} v + \mathbf{b_v})$$

where $\mathbf{W_v}$ and $\mathbf{b_v}$ are parameters, and $\sigma(\cdot)$ is a sigmoid function that gives the probability if r is a proper response to *context*. Model parameters are updated according to a cross-entropy loss during training.

Table 2. Statistics of e-commerce dialogue corpus.

	Training	Validation	Testing
# context-response pairs	1 M	10 K	10 K
# candidates per context	2	2	10
Avg # turns per context	5.51	5.48	5.64
Avg # words per utterance	7.02	6.99	7.11

4 Experiments

4.1 Dataset

We evaluate our model on the E-commerce Dialogue Corpus released by [8]. The statistics of dataset is shown in Table 2.

E-commerce Dialogue Corpus is collected from Taobao[1], the largest e-commerce platform in China. The dataset contains real-world conversation between customers and customer service agent, including commodity consultation, logistics express, recommendation, negotiation, chitchat, and so on. Using the last utterance along with the top-5 key words in the context as a query, the similar utterances in corpus are searched using *Apache Lucene*. Then, the responses after the searched utterances are selected as negative responses. The ratio of the positive and the negative is 1:1 in training set and validation set, and 1:9 in test set. This setting ensures that the model will not be affected due to data imbalance during training, while it is consistent with the real application scenario during testing. We used the same retrieval evaluation metrics, Recall at position k in n candidates ($R_n@k$), as those used in previous work. Since there is only one proper response for each context, $R_n@k$ measures model's ability that rank the proper response in the top-k when there are n candidates.

4.2 Baselines

The baselines used in this paper can be divided into two categories: single-turn matching models and multi-turn matching models. Single-turn matching models concatenate the context utterance together to match a response, including basic models such as TF-IDF, RNN, CNN, LSTM and BiLSTM, as well as advanced models such as MV-LSTM [19], Match-LSTM [20] and ESIM, which is a very effective sentence matching model. We also reproduce SSE [21], which uses two layers of bidirectional LSTM with short connection and maximum pooling to encode two utterances respectively, and inputs the representation of the two sentences and their relationship characteristics into MLP for classification. The multi-turn matching models can be categorized into two types: representation-based and interaction-based models. Multi-View [22] models utterance relationships from both word sequence view and utterance sequence view; DL2R [23] reformulates the last context utterance with other utterances. These two representation-based models can obtain and classify the representation vectors of the whole context

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

and the response. On the other hand, SMN matches a response with each utterance in the context on multiple levels of granularity, and accumulate matching information through a recurrent neural network; DUA uses the last context utterance to refine each utterance through a self-matching attention. DAM utilizes self-attention mechanism instead of RNN to encode utterance, adds attention matching matrix based on interactive attention representation when constructing matching matrix, and uses 3D-convolution instead of RNN in aggregating matching feature. These models can be classified as interaction-based models.

4.3 Training Details

We consider at most 9 context utterances and 50 words for each utterance (response) in our experiments. We pad zeroes if the number of context utterances is less than 9, otherwise we keep the last 9 context utterances, and pad zeroes if the number of the words is less than 50, otherwise we keep the first 50 words. Word embeddings are initialized by the result of skip-gram algorithm which are pre-trained on the training data and updated during model training. The dimensionality of word vectors is 200. The hidden state dimensionality of two GRUs is 200. The first convolution layer has 12 [3, 3, 3] filters with [1, 1, 1] stride, and its max-pooling size is [3, 3, 3] with [3, 3, 3] stride. The second convolution layer has 6 [3, 3, 3] filters with [1, 1, 1] stride, and its max-pooling size is also [3, 3, 3] with [3, 3, 3] stride. Dropout with a rate of 0.2 is applied after word embeddings, before the last perceptron, and applied to all representations that participate in the calculation of M_1^i, M_2^i, M_3^i. REM updates the parameters with Adam [24] optimizer to minimize cross entropy. Learning rate is initialized as 0.001 and gradually decreased during training, and the batch-size is 256 and a larger batch-size in a certain range is more effective. We monitor the performance change of the model on validation set, and we make a early stop if no better performance is produced after multiple turns. Then we select the best performance model on validation set for testing. Our model achieves the best result when traversing approximately 4 epochs of the whole training samples in Ecommerce Dialogue Corpus.

4.4 Experimental Results

Table 3 presents the evaluation results of REM and baseline methods. All the results except ours are reported from the existing literature [4, 9] on the same dataset. As for REM, we repeat the experiment for 5 times and take the average of the results of the 5 runs, and as for SSE and RE2, we repeat each experiment for 3 times. As demonstrated, REM outperforms the other models dramatically at R_{10}@1 and R_{10}@2. REM outperforms the strong baseline DUA by a large improvement of 8.9% in terms of R_{10}@1, 8.4% in terms of R_{10}@2 and 2.6% in terms of R_{10}@5. All the previous multi-turn matching models do not concern different treatment to utterances according to their speakers' roles, which indicates the effectiveness of our role-aware enhancing approach on the multi-turn customer service response selection task. Notice that while REM outperforms the baseline model ESIM by a margin of 2% in terms of R_{10}@1 and 1.7% in terms of R_{10}@2, ESIM has a slightly higher R_{10}@5 than our model. In other words, while ESIM and REM can both rank the proper response into top-5, REM can give proper response a higher ranking, illustrating our REM model can better distinguish the challenging wrong candidates.

Table 3. Evaluation results of REM and baselines on the E-commerce dialogue corpus.

Model	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
TF-IDF	0.159	0.256	0.477
RNN	0.325	0.463	0.775
CNN	0.328	0.515	0.792
LSTM	0.365	0.536	0.828
BiLSTM	0.355	0.525	0.825
SSE	0.559	0.755	0.946
MV-LSTM	0.412	0.591	0.857
Match-LSTM	0.410	0.590	0.858
ESIM	0.570	0.767	**0.948**
Multi-View	0.421	0.601	0.861
DL2R	0.399	0.571	0.842
SMN	0.453	0.654	0.886
DUA	0.501	0.700	0.921
REM	**0.590**	**0.784**	0.947

Table 4. Evaluation results of model ablation.

	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
REM	0.590	0.784	0.947
− Different encoder	0.553	0.754	0.942
$-M_1$	0.552	0.748	0.935
$-M_2$	0.567	0.762	0.944
$-M_3$	0.560	0.764	0.947

4.5 Discussion

Model Ablation. To demonstrate the importance of each component in our proposed model, parts of the architecture are ablated. Table 4 presents the evaluation results of model ablation and for all the results, we repeat the experiment 5 times and adopt their average values. First, sharing parameters of customer utterance GRU encoder and agent utterance GRU encoder (denoted as - Different Encoder) makes the performance drop dramatically. This is because the importance of information is related to the role of the speaker, thus the information in words should be encoded into different semantic spaces.

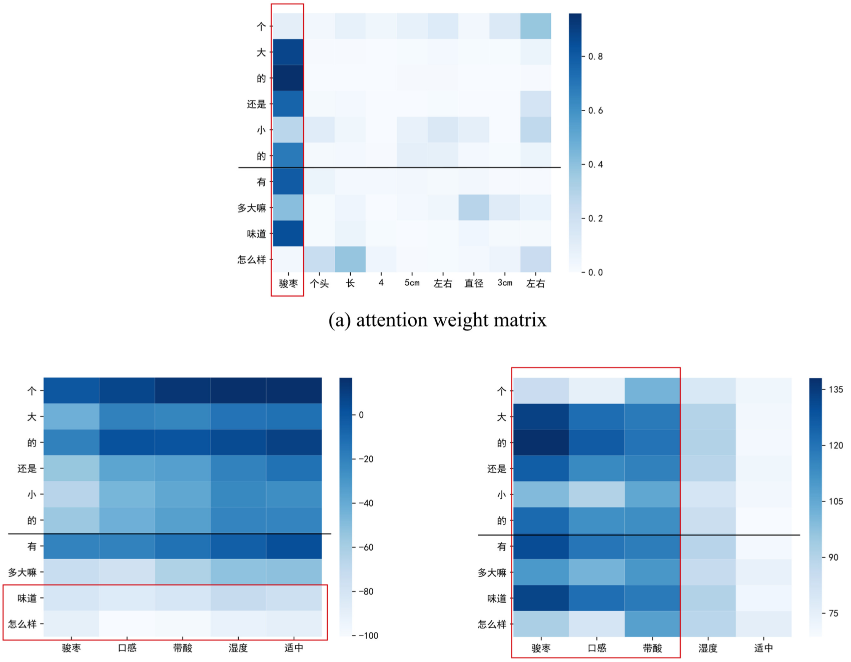
Then we remove one of the three matrices each time (denoted as $-M_1$, $-M_2$, $-M_3$). The biggest performance drop happens when removing M_1, because $\overline{c_i}$ is the foundation representation of words. The performance drop also happens when removing M_2 and M_3, proving that every component of our method is important, and our model is effective with all the components combined.

Table 5. Dialogue example for model visualization.

Speaker	Utterance
Customer (c_1)	Is it big or small?
Agent (s_1)	The Jun jujube is about 4 to 5 cm long and 3 cm in diameter
Customer (c_2)	How does it taste?
Agent (r)	The Jun jujube has an sour taste and moderate humidity

Visualization. We visualize matching matrix M_1, the attention weight matrix of $\bar{\mathbf{C}}$ and $\bar{\mathbf{S}}$, and M_2 of an example in Fig. 3, to show how the agent utterances be fused into customer utterances representations according to their dependency. We give an example by Table 5. The example is {c_1: Is it big or small? s_1: The Jun jujube is about 4 to 5 cm long and 3 cm in diameter; c_2: How does it taste? r: The Jun jujube has an sour taste and moderate humidity.}. We concatenate c_1 and c_2's M_1, and separate them by a black line in Fig. 3 (b). M_1 shows the matching matrix of c_1, c_2 and r's contextual word information. The part enclosed by a red box denote the relationship between utterance "How does it taste?" and response, which has larger absolute values (Lighter colors mean smaller negative values in this figure). This is because M_1 captures their semantic relation directly. The attention weight matrix reflects the dependency relationship between customer words and agent words illustrated by Fig. 3(a). In this example, word "Jun jujube" has larger weights to the majority of customer words, while there have no commodity's name in this context. This attention operation complements key information for customer utterances, and let M_2 highlight the matching values of words "Jun jujube", "taste" and "sour", because they are related to the commodity, illustrated by Fig. 3 (c).

Maximum Context Length. We investigate the influence of maximum context length for REM. Figure 4 shows the performance of REM on E-commerce Dialogue Corpus with respect to maximum context length. Notice that we always set maximum context length as an odd number, because we aim to select response of agent while it is always customer to express appeals first. We find that $R_{10}@1$ performance improves in general when the maximum context length increases, but $R_{10}@2$ decreases after length reaches 9, which means metrics may not improve stably. This can be explained for two reasons: firstly, the context length of 86% of the samples in the dataset is not more than 9, and secondly, there is hardly any relevant information in context utterance which has such long distance to the response. To balance effectiveness and efficiency, we set the maximum context length as 9.



(a) attention weight matrix

(b) M_1 of c_1, c_2 and r

(c) M_2 of c_1, c_2 and r

Fig. 3. Model visualization. The Chinese words are translated as 骏枣-Jun jujube, 口感-taste, 酸-sour, 湿度-humidity, 适中-moderate (Color figure online).

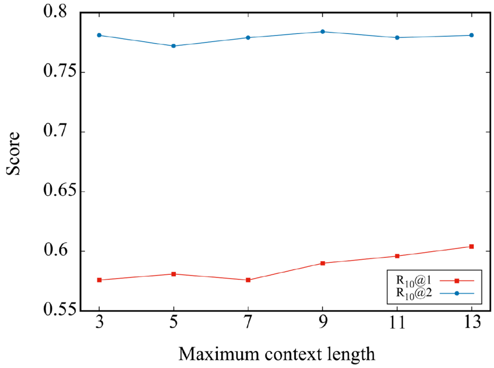


Fig. 4. Performance of REM across maximum context length

Error Analysis. Ignoring multiple suitable responses situation of the dataset, we investigate error cases and draw two conclusions. Firstly, when there is only one utterance in context, REM cannot conduct the role-aware enhancing, losing advantage over the other models. Secondly, some samples in the dataset are incomplete dialogues, losing customer's original intention. Although it is possible to select the response based on the agent context, our role-aware enhancing may cause information loss when customer's words have less semantic information. Here we give an example by Table 6. The example is $\{c_1$: What do you mean?; s_1: We'll intercept the goods and send to your new address; c_2: So what do I need to give you?; r: Your new address.$\}$. In this dialogue, the semantic information needed by response selecting is mainly in s_1, and cannot be effectively fused into c_1, c_2's representation, because c_1, c_2 have no words related to the topic.

Table 6. Dialogue example for error analysis

Speaker	Utterance
Customer (c_1)	What do you mean?
Agent (s_1)	We'll intercept the goods and send to your new address
Customer (c_2)	So what do I need to give you?
Agent (r)	Your new address

5 Conclusion

In this paper, we propose a retrieval-based multi-turn customer service response selection approach, which distinguishes utterances from the perspective of speaker's role differences and enriches the semantic features of context by using role-aware enhancing. Extensive experiments on an e-commerce dialogue dataset have confirmed the superiority of the proposed model over several strong baselines. Comprehensive analysis are conducted to further evaluate the model. In the future, we would like to explore other utterance representation, such as CNN or self-attention, and study more effective methods of role-aware enhancing.

Acknowledgements. The work was supported by the National Key R&D Program of China under grant 2018YFB1004700, National Natural Science Foundation of China (61872074, 61772122), and the Fundamental Research Funds for the Central Universities (N180716010).

References

1. Shum, H., He, X., Li, D.: From Eliza to XiaoIce: challenges and opportunities with social chatbots. Front. Inf. Technol. Electron. Eng. **19**(1), 10–26 (2018). https://doi.org/10.1631/FITEE.1700826
2. Li, F.-L., et al.: Alime assist: an intelligent assistant for creating an innovative e-commerce experience. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 2495–2498. ACM (2017)
3. Zhu, P., Zhang, Z., Li, J., Huang, Y., Zhao, H.: Lingke: a fine-grained multi-turn chatbot for customer service. In: COLING (Demos), pp. 108–112 (2018)

4. Qiu, M., et al.: AliMe Chat: a sequence to sequence and rerank based chatbot engine. In: ACL, pp. 498–503 (2017)
5. Wang, H., Lu, Z., Li, H., Enhong, C.: A dataset for research on short-text conversations. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 935–945 (2013)
6. Ji, Z., Lu, Z., Li, H.: An information retrieval approach to short text conversation. arXiv preprint arXiv:1408.6988 (2014)
7. Wu, Y., Wu, W., Xing, C., Zhou, M., Li, Z.: Sequential matching network: a new architecture for multi-turn response selection in retrieval-based chatbots. In: ACL, pp. 496–505 (2017)
8. Zhang, Z., Li, J., Zhu, P., Zhao, H., Liu, G.: Modeling multi-turn conversation with deep utterance aggregation. In: COLING, pp. 3740–3752 (2018)
9. Yuan, C., et al.: Multi-hop selector network for multi-turn response selection in retrieval-based chatbots. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 111–120 (2019)
10. Wang, H., Lu, Z., Li, H., Chen, E.: A dataset for research on short-text conversations. In: EMNLP, pp. 935–945 (2013)
11. Ji, Z., Lu, Z., Li, H.: An information retrieval approach to short text conversation. arXiv preprint arXiv:1408.6988 (2014)
12. Lowe, R., Pow, N., Serban, I., Pineau, J.: The ubuntu dialogue corpus: a large dataset for research in unstructured multi-turn dialogue systems. In: SIGDIAL Conference, pp. 285–294 (2015)
13. Chen, Q., Wang, W.: Sequential Attention-based Network for Noetic End-to-End Response Selection. arXiv preprint arXiv:1901.02609 (2019)
14. Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., Inkpen, D.: Enhanced LSTM for natural language inference. In: ACL, pp. 1657–1668 (2017)
15. Zhou, X., et al.: Multi-turn response selection for chatbots with deep attention matching network. In: ACL, pp. 1118–1127 (2018)
16. Tao, C., Wu, W., Xu, C., Hu, W., Zhao, D., Yan, R.: Multi-Representation fusion network for multi-turn response selection in retrieval-based chatbots. In: WSDM, pp. 267–275 (2019)
17. Gu, J., Ling, Z., Liu, Q.: Interactive matching network for multi-turn response selection in retrieval-based chatbots. arXiv preprint arXiv:1901.01824 (2019)
18. Wang, S., Jiang, J.: Compare-aggregate model for matching text sequences. In: ICLR (2017)
19. Wan, S., Lan, Y., Xu, J., Guo, J., Pang, L., Cheng, X.: Match-SRNN: modeling the recursive matching structure with spatial RNN. In: IJCAI, pp. 2922–2928 (2016)
20. Wang, S., Jiang, J.: Learning natural language inference with LSTM. In: HLT-NAACL, pp. 1442–1451 (2016)
21. Nie, Y., Bansal, M.: Shortcut-stacked sentence encoders for multi-domain inference. In: Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP (RepEval@EMNLP), pp. 41–45 (2017)
22. Zhou, X., et al.: Multi-view response selection for human-computer conversation. In: EMNLP, pp. 372–381 (2016)
23. Yan, R., Song, Y., Wu, H.: Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In: SIGIR, pp. 55–64 (2016)
24. Kingma, D.P., Ba, J.: 2014. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

Author Index