Chengfei Liu · Lei Zou
Jianxin Li (Eds.)

LNCS 10829

Database Systems for Advanced Applications

DASFAA 2018 International Workshops:
BDMS, BDQM, GDMA, and SeCoP
Gold Coast, QLD, Australia, May 21–24, 2018, Proceedings

Springer

Lecture Notes in Computer Science 10829

More information about this series at http://www.springer.com/series/7409

Chengfei Liu · Lei Zou · Jianxin Li (Eds.)

Database Systems for Advanced Applications

DASFAA 2018 International Workshops: BDMS, BDQM, GDMA, and SeCoP
Gold Coast, QLD, Australia, May 21–24, 2018
Proceedings

Springer

Editors
Chengfei Liu
Swinburne University of Technology
Hawthorn, VIC
Australia

Jianxin Li
University of Western Australia
Crawley, WA
Australia

Lei Zou
Peking University
Beijing
China

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-91454-1 ISBN 978-3-319-91455-8 (eBook)
https://doi.org/10.1007/978-3-319-91455-8

Library of Congress Control Number: 2018942340

LNCS Sublibrary: SL3 – Information Systems and Applications, incl. Internet/Web, and HCI

Printed on acid-free paper

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

Preface

Along with the main conference, the DASFAA 2018 workshops provided an international forum for researchers and practitioners to gather and discuss research results and open problems, aiming at more focused problem domains and settings. This year there were four workshops held in conjunction with DASFAA 2018:

- The 5th International Workshop on Big Data Management and Service (BDMS 2018)
- The Third Workshop on Big Data Quality Management (BDQM 2018)
- The Second International Workshop on Graph Data Management and Analysis (GDMA 2018)
- The 5th International Workshop on Semantic Computing and Personalization (SeCoP 2018)

All the workshops were selected after a public call-for-proposals process, and each of them focused on a specific area that contributes to, and complements, the main themes of DASFAA 2018. Each workshop proposal, in addition to the main topics of interest, provided a list of the Organizing Committee members and Program Committee. Once the selected proposals were accepted, each of the workshops proceeded with their own call for papers and reviews of the submissions. In total, 23 papers were accepted, including seven papers for BDMS 2018, five papers for BDQM 2018, five papers for GDMA 2018, and six papers for SeCoP 2018.

We would like to thank all of the members of the Organizing Committees of the respective workshops, along with their Program Committee members, for their tremendous effort in making the DASFAA 2018 workshops a success. In addition, we are grateful to the main conference organizers for their generous support as well as the efforts in including the papers from the workshops in the proceedings series.

March 2018

Chengfei Liu
Lei Zou

BDMS Workshop Organization

Workshop Co-chairs

Kai Zheng	University of Electronic Science and Technology of China, China
Xiaoling Wang	East China Normal University, China
An Liu	Soochow University, China

Program Committee Co-chairs

Muhammad Aamir Cheema	Monash University, Australia
Cheqing Jin	East China Normal University, China
Qizhi Liu	Nanjing University, China
Bin Mu	Tongji University, China
Xuequn Shang	Northwestern Polytechnical University, China
Yaqian Zhou	Fudan University, China
Xuanjing Huang	Fudan University, China
Yan Wang	Macquarie University, Australia
Lizhen Xu	Southeast University, China
Xiaochun Yang	Northeastern University, China
Kun Yue	Yunnan University, China
Dell Zhang	University of London, UK
Xiao Zhang	Renmin University of China, China
Nguyen Quoc Viet Hung	Griffith University, Australia
Bolong Zheng	Aalborg University, Denmark
Guanfeng Liu	Soochow University, China
Detian Zhang	Jiangnan University, China

BDQM Workshop Organization

Workshop Chair

Qun Chen	Northwestern Polytechnical University, China

Program Committee

Hongzhi Wang	Harbin Institute of Technology, China
Guoliang Li	Tsinghua University, China
Rui Zhang	The University of Melbourne, Australia
Zhifeng Bao	RMIT, Australia
Xiaochun Yang	Northeastern University, China
Yueguo Chen	Renmin University, China
Nan Tang	QCIR, Qatar
Rihan Hai	RWTH Aachen University, Germany
Laure Berti-Equille	Hamad Bin Khalifa University, Qatar
Yingyi Bu	Couchbase, USA
Jiannan Wang	Simon Fraser University, Canada
Xianmin Liu	Harbin Institute of Technology, China
Zhijing Qin	Pinterest, USA
Cheqing Jin	East China Normal University, China
Wenjie Zhang	University of New South Wales, Australia
Shuai Ma	Beihang University, China
Lingli Li	Heilongjiang University, China
Hailong Liu	Northwestern Polytechnical University, China

GDMA Workshop Organization

Workshop Co-chairs

Lei Zou	Peking University, China
Xiaowang Zhang	Tianjin University, China

Program Committee

Robert Brijder	Hasselt University, Belgium
George H. L. Fletcher	Technische Universiteit Eindhoven, The Netherlands
Liang Hong	Wuhan University, China
Xin Huang	Hong Kong Baptist University, SAR China
Egor V. Kostylev	University of Oxford, UK
Peng Peng	Hunan University, China
Sherif Sakr	University of New South Wales, Australia
Zechao Shang	The University of Chicago, USA
Hongzhi Wang	Harbin University of Industry, China
Junhu Wang	Griffith University, Australia
Kewen Wang	Griffith University, Australia
Zhe Wang	Griffith University, Australia
Guohui Xiao	Free University of Bozen-Bolzano, Italy
Jeffrey Xu Yu	Chinese University of Hong Kong, SAR China
Xiaowang Zhang	Tianjin University, China
Zhiwei Zhang	Hong Kong Baptist University, SAR China
Lei Zou	Peking University, China

SeCop Workshop Organization

Honorary Co-chairs

Reggie Kwan	The Open University of Hong Kong, SAR China
Fu Lee Wang	Caritas Institute of Higher Education, SAR China

General Co-chairs

Yi Cai	South China University of Technology, China
Tak-Lam Wong	Douglas College, Canada
Tianyong Hao	Guangdong University of Foreign Studies, China

Organizing Co-chairs

Zhaoqing Pan	Nanjing University of Information Science and Technology, China
Wei Chen	Agricultural Information Institute of CAAS, China
Haoran Xie	The Education University of Hong Kong, SAR China

Publicity Co-chairs

Xiaohui Tao	Southern Queensland University, Australia
Di Zou	The Education University of Hong Kong, SAR China
Zhenguo Yang	Guangdong University of Technology, China

Program Committee

Zhiwen Yu	South China University of Technology, China
Jian Chen	South China University of Technology, China
Raymong Y. K. Lau	City University of Hong Kong, SAR China
Rong Pan	Sun Yat-Sen University, China
Yunjun Gao	Zhejiang University, China
Shaojie Qiao	Southwest Jiaotong University, China
Jianke Zhu	Zhejiang University, China
Neil Y. Yen	University of Aizu, Japan
Derong Shen	Northeastern University, China
Jing Yang	Research Center on Fictitious Economy & Data Science CAS, China
Wen Wu	Hong Kong Baptist University, SAR China
Raymong Wong	Hong Kong University of Science and Technology, SAR China
Cui Wenjuan	China Academy of Sciences, China

Xiaodong Li	Hohai University, China
Xiangping Zhai	Nanjing University of Aeronautics and Astronautics, China
Xu Wang	Shenzhen University, China
Ran Wang	Shenzhen University, China
Debby Dan Wang	National University of Singapore, Singapore
Jianming Lv	South China University of Technology, China
Tao Wang	The University of Southampton, UK
Guangliang Chen	TU Delft, The Netherlands
Wenji Ma	Columbia University, USA
Kai Yang	South China University of Technology, China
Yun Ma	City University of Hong Kong, SAR China

Contents

The 5th International Workshop on Big Data Management and Service (BDMS 2018)

The 3rd Workshop on Big Data Quality Management (BDQM 2018)

The 2nd International Workshop on Graph Data Management and Analysis (GDMA 2018)

The 5th International Symposium on Semantic Computing and Personalization (SeCoP 2018)

The 5th International Workshop on Big Data Management and Service (BDMS 2018)

Convolutional Neural Networks for Text Classification with Multi-size Convolution and Multi-type Pooling

Tao Liang[1], Guowu Yang[1], Fengmao Lv[1](✉), Juliang Zhang[1,2], Zhantao Cao[1], and Qing Li[1]

[1] School of Computer Science and Engineering, Big Data Research Center,
University of Electronic Science and Technology of China,
Chengdu 611731, Sichuan, China
TaoLiang_uestc@126.com, {guowu,liqing}@uestc.edu.cn, fengmaolv@126.com,
caozhantao@163.com, zjlgj@163.com

[2] School of Computer Science and Engineering,
University of Xinjiang Finance and Economics, Urumqi 830000, China

Abstract. Text classification is a very important problem in Nature Language Processing (NLP). The text classification based on shallow machine-learning models takes too much time and energy to extract features of data, but only obtains poor performance. Recently, deep learning methods are widely used in text classification and result in good performance. In this paper, we propose a Convolutional Neural Network (CNN) with multi-size convolution and multi-type pooling for text classification. In our method, we adopt CNNs to extract features of the texts and then select the important information of these features through multi-type pooling. Experiments show that the CNN with multi-convolution and multi-type pooling (CNN-MCMP) obtains better performance on text classification compared with both the shallow machine-learning models and other CNN architectures.

Keywords: Convolutional Neural Networks (CNNs)
Nature Language Processing (NLP) · Text classification
Multi-size convolution · Multi-type pooling

1 Introduction

Text classification [12] is a very important problem in natural language processing (NLP). In the recent years, it has been widely adopted in information filtering, textual anomaly detection, semantic analysis, sentimental analysis, *et al.* Generally, the traditional text classification methods can be divided into two stages: artificial features engineering and classification with shallow machine leaning models such as Naive Bayes (NB), K-Nearest-Neighbors (KNN), Support Vector Machine (SVM), *et al.* In particular, feature engineering needs to construct the significant features that can be used for classification through

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 3–12, 2018.
https://doi.org/10.1007/978-3-319-91455-8_1

text preprocessing, feature extraction, and text representation. However, the feature engineering takes a large amount of time to obtain effective features since domain-specific knowledges are usually needed for a specific text classification task. Additionally, feature engineering is not possessed of strong generality, and a type of expression of textual features for a task may not be applicable for the other tasks.

We all know that the important reason why deep learning algorithms achieved great performance in the field of image recognition is that the image data is continuous and dense. But the text data is discrete and sparse. So if we want to introduce the deep learning methods into text classification, the essential problem is to solve the expression of text data. In other words, we should change the text data into continuous and dense data. Above all, deep learning itself has a strong property of data migration and lots of deep learning algorithms that are well suited to the field of the image recognition can also be used well in text classification.

In this paper, we propose a convolutional neural network with multi-size convolution and multi-type pooling (CNN-MCMP) for text classification. We exploit multiple size of convolutional windows to capture different combinations of information in the original text data. In addition, we use the multiple type pooling to select information of features. Shown in Table 1. The goal of pooling is to ensure the input of the full-connection layer is fixed and choose a variety of standard optimal feature of classification at the same time. The experiments that our proposed CNN-MCMP can obtain better performance on text classification compared with both the shallow machine-learning models and the previous CNN architectures.

Table 1. Difference between our works and existing works

	Our works	Existing works
1	Artificial features engineering, too much time and energy	End to end, little time and energy
2	Single type pooling	Multi-type pooling
3	Multi-size convolution	Multi-size convolution, add two special size: d = 1 and d = n

2 Convolutional Neural Network

CNN is a feedforward neural network, and it makes remarkable achievements in the field of image recognition. In general, the basic structure of the CNN includes four types of network layers: convolution layer, activation layer, pooling layer, fully-connection layer. Part of the networks may remove the pooling layer or fully-connection layer because of the special task. Shown in Fig. 1. Convolution layer is an essential network layer in CNN and each layer consists of several

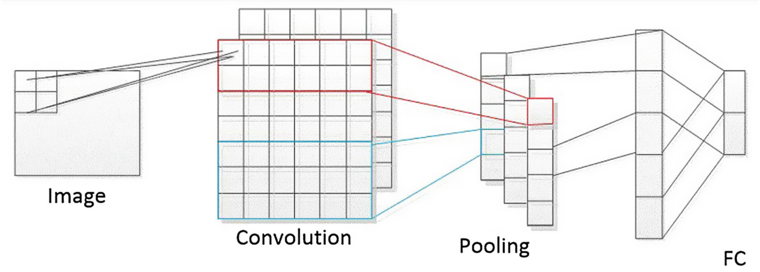


Fig. 1. The model structure of CNN

convolution kernels. The parameters of each convolution layer are optimized by BP (Back Propagation) algorithm [4]. The main purpose of the convolution operation is to extract different features of the input data and the complexity of the features gradually changed form shallow to deep.

The activation function layer can be combined with convolution layer and it can introduce non-linear factors into model because the linear model is not capable of dealing with many non-linear problems. And the activation function which commonly used are ReLU, Tanh, Sigmoid.

Pooling layer is often behind convolution layer. On the one hand, it can make feature map smaller to reduce the complexity of the network. On the other hand, it can select the important features. And the pooling which commonly used are max-pooling, average-pooling and min-pooling.

Fully-connection layer is generally the last layer of CNN. And the goal of the fully-connection layer is to combine local features into the global features which are used to calculate the confidence of each of the categories.

3 Methodology

This chapter mainly introduce the structure and implementation process of CNN-MCMP. First of all we introduce a brief introduction to the basic flow of model training and then focus on the model's word representation, multi-size convolution and multi-type pooling.

The basic flow of model training includes distributed representation and normalization of words, feature information extraction, feature information filtering and classification. When the model starts training, the original text data is changed into continuous and dense word vectors, and then extracted features from text data, choosing the important feature. Finally, we get the final model after training. Shown in Fig. 2. And the next section, we will introduce how to extract the features by multi-size convolution and how to select the feature information by multi-type pooling.

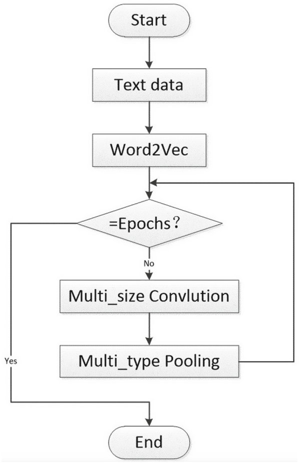


Fig. 2. The flow chart of the model training

3.1 Words Representation

The original data we get is multiple sentences made up of words. Obviously such data can not be used directly in model training, we should change it into real number. Traditionally, one-hot encoding [1] has been used to encode each word in sentence and it's so easy to represent. However, one-hot encoding can also leave the model facing some serious problems which are dimensionality disaster [13] and losing the important order of the sentences. The model will get the poor result in text classification by this way.

As mentioned above, an important operation for introducing the deep learning algorithms into NLP is to convert the discrete and sparse data into the continuous and dense data, shown in Fig. 3. We use two different conversion methods to change the original text data. The simplest way is to initialize the words using random real numbers. And the range of random real numbers is controlled from −0.5 to 0.5 in order to speed up the convergence of experiments and the quality of the word vectors [9,10]. The second method is using the pre-training word vectors. We use the word vectors proposed by Word2Vec in Google to initialize word vectors and the word vectors are trained based on Google news (about 30,000,000 words). The vectors' dimension of each word is 300 and represents the relationship between words. When change the words into word vectors, we directly find the corresponding word vectors of words in pre-trained word vectors.

3.2 Multi-size Convolution

We can use the model to classify the data after we changed the original text data into word vectors. We need convolution layer in model to extract the features of

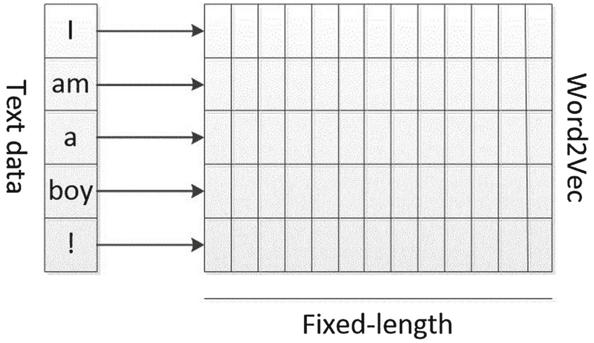


Fig. 3. Words representation

text as the main basis of classification. And we exploit multiple size convolutional windows to extract more different features.

In traditional convolutional neural network, the convolution kernel is fixed during the convolution process. However, the fixed size of the convolution kernel can not capture the semantic information as much as possible and the features extracted by model can not include enough information to classify data. Therefore, the introduction of multi-size convolution is necessary. It can capture the more textual information during the convolution process, because the different size of convolution kernel is different combination of n-gram in fact. The different combination of n-gram represent different combination of words in sentences. In addition, we introduce two special size of convolution: size = 1 and size = n (the length of sentence). Size = 1 makes model capture the information of words and size = n makes model capture the information of sentence. The multi-size convolution is shown in Fig. 4.

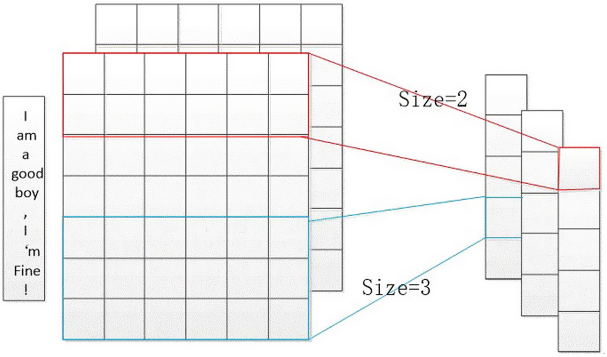


Fig. 4. Multi-size convolution

From the Fig. 4, given the sentence “I am a good boy, I’m Fine!”, we can get the a two-dimensional array through the word vectors, and the height of two-dimensional array is the length of sentence, the width of two-dimensional array is dimension of word vector where the dimension is 300. Given the two size of convolution kernel (size = 2 and size = 3) and each type of kernel extracted features on two-dimensional array to get the corresponding feature map.

3.3 Multi-type Pooling

We need to select the feature information extracted by convolution layer to get the maximum value of features or get the global feedback on these features. Therefore, we should exploit multi-type pooling to select the features, and different type pooling can get the more combinations of features to classify data.

In this paper, there are some functions of pooling: Fixed sentence length, because the multi-size convolutional kernel gets different size feature maps and we should ensure the input size is same before sending to fully-connection layer. And different size of feature map can be changed into same size after pooling. We mainly use two type pooling: max-pooling and average-pooling. Max-pooling can extract the maximum value of each feature map to splice into a new fixed vector. And the average-pooling can extract average information form feature map. The maximum value of each feature map and the average value of feature map include more complete information of sentence. The max-pooling can extract the maximum semantic information in the textual sentences and average-pooling can extract the average semantic information of the textual sentences. The operation of multi-type pooling is shown in Fig. 5.

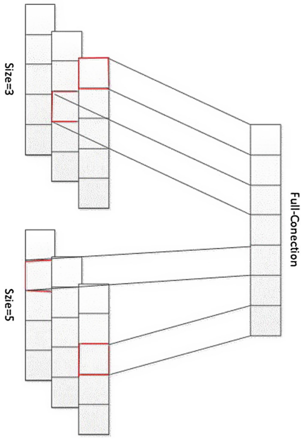


Fig. 5. Multi-type pooling

Figure 5 shows the operation of multi-type pooling and for the n feature maps obtained from the previous convolution, we can get two vectors which length is n through max-pooling and average-pooling. And then the two vectors are spliced into a vector as the input of fully-connection layer.

4 Experiments

We tested our network on two different datasets. Our experimental datasets involves binary classification and multi-class classification which involve sentiment analysis and theme recognition about NLP tasks.

We should control the learning rate and use a more flexible learning-rate setting method-exponential decay during the model training to more effectively train model and balance the speed and performance of the model. At the beginning, the learning-rate and the attenuation coefficient are set to 0.01 and 0.95, respectively. The value of learning-rate gradually decreases as the number of iterations increases to better approximate the optimal value.

4.1 MRS Data

MRS is a dataset about sentiment analysis [11] in NLP and each data belongs to a certain kind of emotion such as happy, sad, angry. MRS dataset is a binary classification dataset and each piece of data is a comment on the movie. The goal of the model is to dismiss the comment as a positive or negative comment. MRS dataset contains a total of 10662 data, which the training set contains pieces of 9600 review data and test set contains 1062 pieces of review data. In the experiment, two methods we used to initialize word vectors: random initialization and pre-trained initialization. The random initialization is to randomly initialize the word vectors into a certain range of real number and trained along with parameters of model. The pre-trained initialization is to initialize the word vectors with word vectors come from Word2Vec and trained along with parameters of model as well.

We compared our model with many existing network models to show the good performance of our model. The models include some machine learning models such as Sent-Parser model [3], NBSVM model [17], MNB model, G-Dropout model, F-Dropout model [16] and Tree-CRF model [11] and some convolution neural network models such as Fast-Text model [6], MV-RNN model [14], RAE model [15], CCAE model [5], CNN-rand model and CNN-non-static model [7]. As shown in Table 2, our model can obtain better performance than the compared methods.

4.2 TREC Data

TREC dataset is a dataset about QA in NLP and belongs multi-class classification. The TREC questions dataset involve six different question types, e.g. where the question is about a location, about a person or about some numeric information. The training dataset consists 5452 labelled questions whereas the test dataset consists of 500 questions.

Table 2. The accuracy on MRS-Data

Model	MRS
CNN-MCMP-rand	78.6
CNN-MCMP-non-static	**82.5**
Fast-Text	78.8
CNN-rand	76.1
CNN-non-static	81.5
RAE	77.7
MV-RNN	79.0
CCAE	77.8
Sent-Parse	79.5
NBSVM	79.4
MNB	79.0
G-Dropout	79.0
F-Dropout	79.1
Tree-CRF	77.3

We compared our model with three different model types: HIER model [8], MAX-TDNN model [2] and NBOW model. These network models include both non-neural network models and neural network models. We set the size of convolution kernel to be 2, 3 and 5 in multi-size convolution operation, respectively, the corresponding number of features is 200, 300 and 500. As shown in Table 3, our CNN-MCMP can obtain better results, compared with the other three models.

Table 3. The accuracy on TREC Data

Model	TREC
CNN-MCMP-non-static	**91.6**
CNN-MCMP-rand	90.4
HIRE	91.0
MAX-TDNN	84.4
NBOW	88.2

5 Conclusion

In this paper, we propose CNN-MCMP for text classification. Our method use multi-size convolution and multi-type pooling (including both max-pooling and average-pooling) in the CNN architecture. The multi-size convolution empowers the model to extract diverse n-gram semantic composition information. As for the multi-type pooling, the max-pooling can extract the most discriminative features for classification, while the average-pooling extracts averaged features to avoid the classification errors caused by accidental factors. Benefitting from the multi-size convolution and multi-type pooling, our method can achieve significant improvements over both the shallow machine learning models and the previous CNN architectures in text classification.

In our future work, we will focus on operating on the word vectors to further improve the performance. In particular, we can randomly disrupt the words in the original sentence to get different new sentences or randomly discard words in the original sentences to get new sentences as well. This operation can expand the scale of the dataset to improve the generalization ability of the model to a certain degree. In addition, the experimental dataset can be incorporated into the corpus to train the word vectors, because the word vectors trained by this way are more suitable for a specific experiment task and more conducive to model training.

References

1. Cassel, M., Lima, F.: Evaluating one-hot encoding finite state machines for SEU reliability in SRAM-based FPGAs. In: 12th IEEE International On-Line Testing Symposium, 2006, IOLTS 2006, 6 pp. IEEE (2006)
2. Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning, pp. 160–167. ACM (2008)
3. Dong, L., Wei, F., Liu, S., Zhou, M., Xu, K.: A statistical parsing framework for sentiment classification. Comput. Linguist. **41**(2), 293–336 (2015)
4. Hecht-Nielsen, R.: Theory of the backpropagation neural network. In: Neural Networks for Perception, pp. 65–93. Elsevier (1992)
5. Hermann, K.M., Blunsom, P.: The role of syntax in vector space models of compositional semantics. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 894–904 (2013)
6. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
7. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
8. Li, X., Roth, D.: Learning question classifiers. In: Proceedings of the 19th International Conference on Computational Linguistics, vol. 1, pp. 1–7. Association for Computational Linguistics (2002)
9. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
10. 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)
11. Nakagawa, T., Inui, K., Kurohashi, S.: Dependency tree-based sentiment classification using CRFs with hidden variables. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 786–794. Association for Computational Linguistics (2010)

12. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Mach. Learn. **39**(2–3), 103–134 (2000)
13. Sapirstein, G.: Social resilience: the forgotten dimension of disaster risk reduction. Jàmbá J. Disaster Risk Stud. **1**(1), 54–63 (2006)
14. Socher, R., Huval, B., Manning, C.D., Ng, A.Y.: Semantic compositionality through recursive matrix-vector spaces. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 1201–1211. Association for Computational Linguistics (2012)
15. Socher, R., Pennington, J., Huang, E.H., Ng, A.Y., Manning, C.D.: Semi-supervised recursive autoencoders for predicting sentiment distributions. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 151–161. Association for Computational Linguistics (2011)
16. Wang, S., Manning, C.: Fast dropout training. In: International Conference on Machine Learning, pp. 118–126 (2013)
17. Wang, S., Manning, C.D.: Baselines and bigrams: simple, good sentiment and topic classification. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, pp. 90–94. Association for Computational Linguistics (2012)

Schema-Driven Performance Evaluation for Highly Concurrent Scenarios

Jingwei Zhang[1,2], Li Feng[2], Qing Yang[3(✉)], and Yuming Lin[2]

[1] Guangxi Cooperative Innovation Center of Cloud Computing and Big Data,
Guilin University of Electronic Technology, Guilin 541004, China
gtzjw@hotmail.com

[2] Guangxi Key Laboratory of Trusted Software,
Guilin University of Electronic Technology, Guilin 541004, China
xintu_li@163.com, ymlinbh@163.com

[3] Guangxi Key Laboratory of Automatic Measurement Technology and Instrument,
Guilin University of Electronic Technology, Guilin 541004, China
gtyqing@hotmail.com

Abstract. Online applications needs to support highly concurrent access and to response to users as soon as possible. Two primary factors make the above requirements to be a technical challenge, one is the large user base, the other is the sharp rise in traffic caused by some specific activities, such as ticketing on 12306.cn during Spring Festival season, shopping on taobao.com during its dual 11 promotions, etc. For the latter, a core focus is how to expand the performance of those existing hardware and software and then to ensure the quality of services when a sharp rise on access happened. Since database schemas have a direct link with data access granularity, etc., this paper considers database schemas as an important factor for performance optimization on highly concurrent access and also covers other elements affecting access performance, such as cache, concurrency, etc., to analyze the performance factors for databases. Extensive experiments are designed to conduct both performance testing and analyzing under different schemas. The experimental results show that a reasonable configuration can contribute a good database performance, which provides factual basis for optimizing highly concurrent applications.

Keywords: Performance optimization · High concurrency
Database schema

1 Introduction

The ecosystem for online applications has presented a remarkable progress in China, which is not only attracting a large number of online users, but also is bringing technical challenges to provide available services. Especially, some special activities can gather large-scale users and cause performance pressure in a specific time range, it is very difficult to provide a regular service with those

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 13–23, 2018.
https://doi.org/10.1007/978-3-319-91455-8_2

existing resources, including hardware and software. For example, the visits and transactions on 12306.cn rises sharply when ticketing during Spring Festival season, taobao.com also presents the same situation during its dual 11 promotions. A prominent characteristic for those applications is that they have a steady performance requirement in most of the time, but will also present a sharp rise on service requirements in a specific time range triggered by some extra events, which will lead to slow response or even unsatisfied services. Usually, the above phenomena is temporary but very critical. It is very necessary to consider some factors to improve the performance and to ensure regular services.

For the above applications, one obvious challenge is highly concurrent access requirements. For dealing with high concurrency, there are two general strategies, one is to extend the hardware devices, such as more computing and storage resources, the other is to design a new software stack. Since the performance requirements for the above applications are temporary, it is not cost-effective to expand hardware, especially, some applications can not be solved easily by simple scaling out, such as ticketing on 12306.cn. Though a new software stack maybe brings an obvious performance improvement, it is still a huge project. Some special factors should be considered to give full play to the advantages of both hardware and software. In this paper, we will focus on the database schemas for close-coupled highly concurrent access performance since they have a direct effect on data granularity. Here, the close coupled highly concurrent access means that those access are not suitable to be realized on a distributed environment, such as ticketing on 12306.cn. The concrete contributions of this paper are as following.

- summarizing the characteristics of highly concurrent access for some online applications, such as ticketing;
- designing two specific database schemas and discussing their related factors for highly concurrent optimization;
- carrying out extensive experiments to provide factual basis for further query optimization in highly concurrent scenarios.

This paper is organized into five parts. Section 2 presents the existing related work on highly concurrent access optimization. Section 3 analyzes the concrete problem and discusses the optimization requirements. Section 4 designs two primary database schemas and analyzes their links for highly concurrent access. Section 5 designs the testing cases and carries experiments to provide analysis basis for further query optimization in highly concurrent scenarios.

2 Related Work

Performance optimization on high concurrency has been a challenge and a popular focus. The first strategy for improving high concurrency is optimization on new hardware and framework. Considering that CPU and GPU are integrated into a single chip, [3] investigated the collaborative performance between

CPU and GPU and their advantages on concurrency. [12] introduced the storage strategies for wechat system, which integrated PaxosStore with combinatorial storage layers to provide maintainability and expansibility for the storage engine. [4,8] was aware that resource competition, transaction interaction etc. have non-linear influence on system performance, and proposed DBSeer, a framework for resource usage analysis and performance prediction, which applied statistical models to measure some performance indexes accurately on highly concurrent OLTP workload. Considering the requirements of verifying the outsourcing data integrity, [7] proposed Concerto, a key-value storage, to substitute online verification by deferring verifications for batch processing and to improve concurrent performance. [6] proposed a new mechanism for concurrent control to guarantee steady system performance of multi-core platform even facing high-competitive workload, which discovered the dependency between those operations of each transaction to avoid killing transactions by restoring nonserializable operations when meeting transaction failures. [2] discussed a new database framework, which separated query processing from transaction management and data storage and then provided data sharing between query processing nodes. This framework is enhanced by flexible transaction processing to support efficient data access.

The second kind of optimization strategy for high concurrency is to consider some specific factors and models. [11] introduced Slalom, a query engine, which monitored users' access schemas to make decisions on dynamic partitions and indexes, and then to improve query performance. [10] proposed a novel multi-query join strategy, which permitted to discover those shared parts for multiple queries and to improve the performance of concurrent queries. Since those data-intensive scientific applications are heavily dependent on file systems with high-speed I/O, [9] put forwards an analytical model for evaluating the performance of highly concurrent data access and provided basis for deciding the stripe size of files to improve I/O performance. Comparing to tuple queries, [1] designed PaSQL to support package query and provided corresponding optimization strategies. For Optimization on distributed environments, [5] put forward a concrete optimization mechanism on Cassandra by make a detailed consideration of the close connection between distributed applications and business scenarios.

3 Problem Analysis

The high concurrency in a short time is triggered by some special application scenarios, which usually cause a sharp rise on the number of user requests, and bring serious performance pressure for daily operational systems, even can not provide normal services, but all above are not the normal state of the applications. These applications have enough hardware and software to support their daily operating, it is not cost-effective to extend hardware for the solution of high concurrency in a short time. In addition, the data objects in those applications are intensive and highly relevant, the performance improvement contributed by scaling out is not obvious. But optimization space can be discovered between the software system and the hardware platform.

Usually, system performance are constrained by the following factors, the first is data, for example, a conflict will happen when updating the same data item simultaneously. The second is communication and the hardware platform, such as network bandwidth, I/O speed of disks, etc. The third is some soft-configurable factors affecting data access performance, such as cache, index, etc. The second kind of factors are stable since they are related with hardware resources. The first kind of factors are decided by the sequence of operations, the concrete implementation of DBMS, etc., which can not be predicted and changed easily. But database schemas have a direct influence on them since schemas have a tight link with the data granularity, for example, the same access requirements will cause different locking range under different schemas.

In order to carry out an effective performance evaluation for highly concurrent access based on database schemas and their relevant factors, this paper will take the ticketing application on 12306.cn as a specific example, and design two schemas, namely station sequence schema and station pair schema, to evaluate the performance of two kinds of queries, which are to query a specific train information by the train no. and to query a specific routine by the designated station names. This evaluation aims at discovering those optimization factors for high concurrency, which can help to exploit the potential ability of both those existing hardware and software to ensure the available services for high concurrency.

4 Database Schema Designing

Database schemas have a great influence on database performance since they are often related with data access granularity, locking size, the times of I/O, and so on. In this section, we will consider the popular application scenario, ticketing, and design two primary database schemas to organize data for further performance evaluation and analysis on high concurrency.

4.1 A Database Schema on Station Sequence

A specific train route consists of a set of concrete train stations and can be represented by a unique ID(train no.), which can be denoted as an ordered n-tuple, $TR = <ID, s_1, s_2, \cdots, s_n>$. Here, s_i corresponds to a tangible train station. For all train routes $\{TR_1, TR_2, \cdots, TR_m\}$, the triple $<ID_i, s_{ij}, j> \in TR_i$ is unique, $1 \leq i \leq m, 1 \leq j \leq n$. We can organize all those tuples $<ID_i$, $s_{ij}, j>$ into databases to form a primary database schema. Table 1 illustrates a part of data conforming to this schema.

For the above schema, a train route with n stations are represented by n records in the database. Each record corresponds to a specific station, which tells the detailed information from the its previous station to the current station. For this schema, it needs an extra computation when you order a ticket between two stations. Assuming your itinerary is from **SongJiang** to **HangZhouDong**, if you want to order a ticket from the train **K149**, you will have to judge whether your itinerary is covered by a specific train, such as collecting those related records in Table 1 to provide the details.

Table 1. Station sequence schema

No.	TrainNo	StationName	StationNo	Duration(mins)	Price(RMB)	Num of tickets
1	K149	ShangHaiNan	1	0	0	200
2	K149	SongJiang	2	19	9	200
3	K149	JiaXing	3	34	11	200
4	K149	HangZhouDong	4	6 0	14.5	200
5	...	...	...	...	...	...

4.2 A Database Schema on Station Pair

Since a ticket is composed of two stations, we can also organize the train route into a series of triples $<ID, s_i, s_j>$, which represents that the train with No.ID can start from the departure station s_i to the arrival station s_j. All these above triples constitute a new schema. Table 2 illustrates a part of data conforming to this schema. Station pair schema provide a direct and detailed representation for train routines.

Table 2. Station pair schema

No.	TrainNo	Start_station	End_station	Duration(mins)	Price(RMB)	Num of tickets
1	K149	ShangHaiNan	SongJiang	19	9	200
2	K149	ShangHaiNan	JiaXing	53	12.5	200
3	K149	ShangHaiNan	HangZhouDong	113	24.5	200
4	K149	SongJiang	JiaXing	34	11	200
5	K149	SongJiang	HangZhouDong	94	23.5	200
6	...	...	...	...	...	...

Compared to the station sequence schema, the station pair schema can provide more convenient query from the departure start to the destination. This schema needs more storage space since it enumerates all reachable routines between any pair of stations. A train route with n stations are represented by $\frac{n*(n-1)}{2}$ records in the database.

Focusing on the above two different schemas, this paper will consider some factors related with the performance optimization for high concurrency brought by large amount of users, such as CPU utilization, query cache, etc., to test query performance.

4.3 Other Optimization Factors

Optimization by Index. Index is a primary mechanism for query optimization on databases since it is helpful to establish more efficient execution plans. At the same time, index should also be given a reasonable consideration since extra cost will be paid to maintain indexes when updating data. Since those attributes included in query predicates are helpful to improve query performance, we will create indexes for station sequence schema on **StationName** and **TrainNo**, and for station pair schema on **Start_station**, **End_station** and **TrainNo**.

Optimization on Query Cache and Connections. Each server has an optimal concurrency capability, which is decided by its hardware and software. When growing closer to its optimal concurrency capability, the server will achieve its maximum throughout, but more workloads will cause a sharp decrease on performance. It is very important to decide the optimal concurrency capability for servers. Since both query cache and database connections have a direct impact on the optimal concurrency capability, we will set query cache and database connections by experiments for improving high concurrency.

5 Experiments and Evaluations

5.1 Experimental Setup and Dataset

In order to evaluate the concurrency performance on the above two schemas, we use a server with 3.3 GHZ 4-core CPU, 8 GB memory and 1 TB hard disk as the experimental platform, which is configured with the open source Database MySQL 5.5.

The dataset is collected from 12306.cn, which includes 3030 train stations and 3114 train routines. In our database, there are 38087 records for station sequence schema and there are 345311 records for station pair schema.

5.2 Testing Cases and Evaluation Metrics

In order to test the performance influence on different schemas, we will design testing cases for highly concurrent query on the above two schemas. The related testing cases are as following,

- **Q1:** to query all related records for a specific train
- **Q2:** to query all available trains for two specific stations

The above two queries are expressed as SQL expressions, which are listed in Table 3.

In order to submit a large number of queries simultaneously for testing highly concurrent performance, we apply multiple threads to submit query requirements. The number of users will vary from 10,000 to 100,000, each user is responsible for submitting one query. The number of database connections will also vary for combination test. The completion time of queries, memory usage and CPU utilization are considered as the evaluation metrics. In order to simplify the test, we will not consider the case of train transit since concurrency performance is our focus and a train transit can be transformed into multiple operations on single train routine.

Table 3. Testing cases

Schema	Q1	Q2

Station sequence schema	SET @routine = 'K149'
	SELECT * FROM station_sequence_schema WHERE TrainNo = @routine
		SET @start = 'ShangHaiNan'
		SET @end = 'GuiLinBei'
		SELECT * FROM station_sequence_schema a WHERE EXISTS (SELECT * FROM station_sequence_schema b WHERE a.TrainNo = b.TrainNo AND a.StationName = @start AND b.StationName = @end AND a.StationNo < b.StationNo)

Station pair schema	SET @routine = 'K149'
	SELECT * FROM station_pair_schema WHERE TrainNo = @routine
		SET @start = 'ShangHaiNan'
		SET @end = 'GuiLinBei'
		SELECT * FROM station_pair_schema WHERE start_station = @start AND end_station =@end

5.3 Experimental Results and Analysis

In this section, we will execute Q1 and Q2 on the designed schemas by different configurations to test concurrency performance.

Experiment 1: Queries on Station Sequence Schema. In this group of experiments, we organized data by the station sequence schema and executed queries with different numbers of database connections and users. Firstly, we simulated a fixed number of users for query cases, Q1 and Q2, and executed queries under different database connections. The number of users is fixed at 10,000 and the number of database connections varies from 10 to 1000. Figure 1 presents the total completion time of queries, in which x-axis is the number of database connections and y-axis corresponds to the completion time of queries. At first, the completion time of both Q1 and Q2 presented a sharp decline when increasing the number of database connections since these query pressure are shared by more connections. But when continuing to increase the number of database connections, the completion time of queries only presents slight changes. This experiments show that the number of database connections have a reasonable range for different query workload. For example, 50 is a reasonable number of connections for the current experiments. Since a database connection needs extra network and memory cost, the number of database connections should be reasonably set for different query workloads.

Secondly, we fixed the number of database connections and varied the number of users to provide different query workload for observing concurrency performance. The number of users is changed from 10,000 to 100,000 and the number of database connections is fixed at 100. Each user is responsible for submitting one query. Figure 2 presents the completion time of queries, which show that the completion time of queries is positively related with the query workload. The number of database connections have a great influence on query performance.

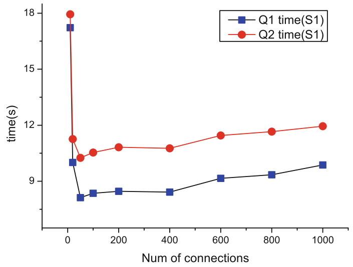


Fig. 1. Query performance on station sequence schema(S1) with different database connections

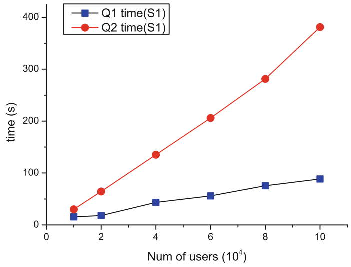


Fig. 2. Query performance on station sequence schema(S1) with different number of users

Experiment 2: Queries on Station Pair Schema. In this group of experiments, we focused on the query performance contributed by the station pair schema. Both different number of database connections and different number of users are considered to observe concurrency performance by two groups of experiments. One is that the number of users is initially set to be 10,000 for query cases, Q1 and Q2, and database connections varies from 10 to 100, the other is that the number of database connections is fixed at 100 and the number of users varies from 10,000 to 100,000. Figures 3 and 4 presents the completion time of queries, which also show that a good number of database connections can contribute a better query performance and can avoid extra network and memory cost.

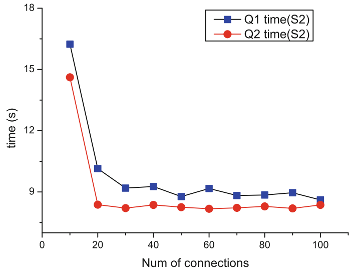


Fig. 3. Query performance on station pair schema(S2) with different database connections

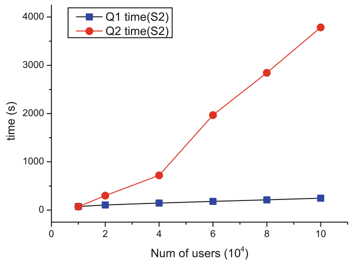


Fig. 4. Query performance on station pair schema(S2) with different number of users

Figures 5 and 6 presents the performance comparison on Q1 and Q2 respectively. The queries on station sequence schema won in all cases, which is attributed to two reasons, one is that station sequence schema uses less records than station pair schema when representing the same information, the other is that station sequence schema touches less number of records than station pair schema for same queries. Station sequence schema is more suitable for query scenarios than station pair schema, but station pair schema can provide a fine-grained data controlling, such as locking when dealing with updating.

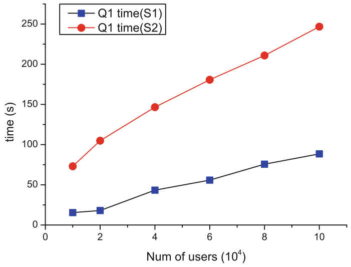


Fig. 5. Query performance comparison of Q1

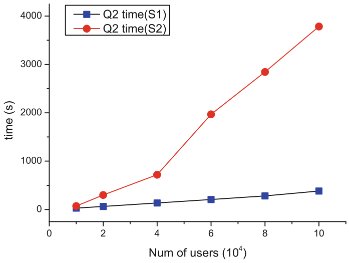


Fig. 6. Query performance comparison of Q2

When the number of database connections is fixed at 100, we observe the usage of both CPU and memory for Q1 and Q2 with different number of users, whose results are listed in Table 4. When a large number of queries arrived, CPU utilization reaches to 100% quickly, which causes a query delay and also indicates that CPU utilization is a primary factor for optimization.

Table 4. CPU and memory usage before optimization

Num of users (10^4)	1	2	4	6	8	10
CPU(%)	82.4	100.0	100.0	100.0	100.0	100.0
Memory (a total of 8,134MB)	5217	5003	5000	5046	5092	5044

Experiment 3: Query Optimization. The group of experiments are responsible for observing the query performance after optimization. The number of database connections are set to be 100 and the number of users varies from 10,000 to 100,000, index and query cache are considered for further query optimization. Firstly, indexes are set on both station sequence schema and station pair schema, those attributes existing in **where** clause, such as TrainNo, Start_station, End_station, etc., are used to establish indexes. Secondly, the query cache is enlarged as big as possible. Figure 7 presents the experimental results, both query cache and index made contributions for query performance improvement. Q2 on station pair schema presents a noticeable improvement, which is

because that the query can directly locate the objectives from a large number of records with the aid of index and that the query cache permits more queries to work simultaneously. Table 5 also presents the optimization results, in which CPU utilization shows a great improvement, the less memory requirements also confirm a shorter query queue. Index and query cache provide a direct improvement for concurrent queries.

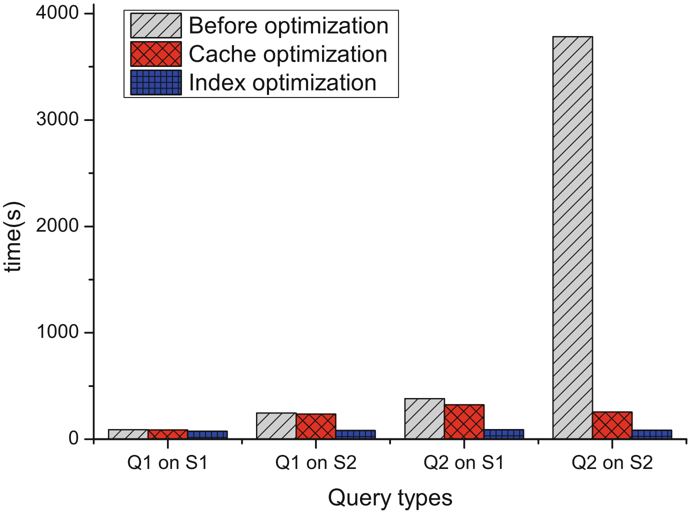


Fig. 7. Query performance comparison on optimization.

Table 5. CPU and memory usage after optimization

Num of users (10^4)	1	2	4	6	8	10
CPU(%)	39.03	45.58	50.28	45.01	50.63	48.43
Memory (a total of 8,134 MB)	4160	3974	4021	4033	4112	4225

6 Conclusions

This paper analyzed the special query phenomenon of the popular application, online ticketing, and summarized the query characteristics, namely tightly coupled access and temporary high concurrency. Considering cost effectiveness of instant performance improvement, this paper focused on database schemas to discuss the potential performance optimization for high concurrency, which mainly cared about the data granularity decided by database schemas. The number of database connections and query cache are also covered to exploit the potential ability of both existing hardware and software. Extensive experiments also proved that database schemas and these related factors can improve query performance when maintaining those current hardware, which provided some effective optimal basis for high concurrency.

Acknowledgments. This study is supported by the National Natural Science Foundation of China (No. 61462017, 61363005, U1501252), Guangxi Natural Science Foundation of China (No. 2017GXNSFAA198035), and Guangxi Cooperative Innovation Center of Cloud Computing and Big Data.

References

1. Brucato, M., Beltran, F.J., Abouzied, A., Meliou, A.: Scalable package queries in relational database systems. Proc. VLDB Endow. **9**(7), 576–587 (2016)
2. Loesing, S., Pilman, M., Etter, T., Kossmann, D.: On the design and scalability of distributed shared-data databases. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD 2015), pp. 663–676 (2015)
3. Zhu, Q., Wu, B., Shen, X.P., Shen, K., Shen, L., Wang, Z.Y.: Understanding co-run performance on CPU-GPU integrated processors: observations, insights, directions. Front. Comput. Sci. **11**(1), 130–146 (2017)
4. Yoon, Y.D., Mozafari, B., Brown, P.D.: DBSeer: pain-free database administration through workload intelligence. Proc. VLDB Endow. **8**(12), 2036–2039 (2015)
5. Mior, J.M., Salem, K., Aboulnaga, A., Liu, R.: NoSE: schema design for NoSQL applications. In: Proceeding of IEEE 32nd International Conference on Data Engineering (ICDE 2016), pp. 181–192 (2016)
6. Wu, Y.J., Chan, Y.C., Tan, K.L.: Transaction healing: scaling optimistic concurrency control on multicores. In: Proceedings of the 2016 ACM SIGMOD International Conference on Management of Data (SIGMOD 2016), pp. 1689–1704 (2016)
7. Arasu, A., Eguro, K., Kaushik, R., Kossmann, D., Meng, P.F., Pandey, V., Ramamurthy, R.: Concerto: a high concurrency key-value store with integrity. In: Proceedings of the 2017 ACM SIGMOD International Conference on Management of Data(SIGMOD 2017), pp. 251–266 (2017)
8. Mozafari, B., Curino, C., Jindal, A., Madden, S.: Performance and resource modeling in highly-concurrent OLTP workloads. In: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD 2013), pp. 301–312 (2013)
9. Dong, B., Li, X.Q., Xiao, L.M., Ruan, L.: A new file-specific stripe size selection method for highly concurrent data access. In: Proceedings of ACM/IEEE 13th International Conference on Grid Computing (GRID 2012), pp. 22–30 (2012)
10. Makreshanski, D., Giannikis, G., Alonso, G., Kossmann, D.: MQJoin: efficient shared execution of main-memory joins. Proc. VLDB Endow. **9**(6), 480–491 (2016)
11. Olma, M., Karpathiotakis, M., Alagiannis, I., Athanassoulis, M., Ailamaki, A.: Slalom: coasting through raw data via adaptive partitioning and indexing. Proc. VLDB Endow. **10**(10), 1106–1117 (2017)
12. Zheng, J.J., Lin, Q., Xu, J.T., Wei, C., Zeng, C.W., Yang, P.A., Zhang, F.: PaxosStore: high-availability storage made practical in WeChat. Proc. VLDB Endow. **10**(12), 1730–1741 (2017)

Time-Based Trajectory Data Partitioning for Efficient Range Query

Zhongwei Yue[1], Jingwei Zhang[1,2], Huibing Zhang[1], and Qing Yang[3](✉)

[1] Guangxi Key Laboratory of Trusted Software,
Guilin University of Electronic Technology, Guilin 541004, China
[2] Guangxi Cooperative Innovation Center of Cloud Computing and Big Data,
Guilin University of Electronic Technology, Guilin 541004, China
[3] Guangxi Key Laboratory of Automatic Measurement Technology and Instrument,
Guilin University of Electronic Technology, Guilin 541004, China
gtyqing@hotmail.com

Abstract. The popularity of mobile terminals has given rise to an extremely large number of trajectories of moving objects. As a result, it is critical to provide effective and efficient query operations on large-scale trajectory data for further retrieval and analysis. Considering data partition has a great influence on processing large-scale data, we present a time-based partitioning technique on trajectory data. This partitioning technique can be applied on the distributed framework to improve the performance of range queries on massive trajectory data. Furthermore, the proposed method adopts time-based hash strategy to ensure both the partition balancing and less partitioning time. Especially, existing trajectory data are not required to be repartitioned when new data arrive. Extensive experiments on three real data sets demonstrated that the performance of the proposed technique outperformed other partitioning techniques.

Keywords: Trajectory data · Partitioning effectiveness
Massive data management

1 Introduction

With the rapid development of mobile Internet and the wide applications of mobile terminals (e.g., mobile phones, sensing devices), the collected trajectory data present an explosive increasement. For instance, T-Drive [1] contains 790 million trajectories generated by 33,000 taxis in Beijing over only a three month period, the total length of all trajectories generated in DiDi platform reached around 13 billion kilometers in 2015. These data not only reflect the spatio-temporal mobility of individuals and groups, but may also contain behavior information from people, vehicles, animals and other moving objects, which are very valuable for route planning, urban planning, etc. [2]. For example, [3] proposed user similarity estimation based on human trajectory, [4] used shared

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 24–35, 2018.
https://doi.org/10.1007/978-3-319-91455-8_3

bike data to plan urban bicycle lanes, and [5] introduced the personalized route recommendation based on urban traffic data. For those above applications, trajectory query is a primary and frequent operation, how to perform queries on massive trajectory data efficiently has become a challenging problem.

Considering efficient distributed processing requirement on large-scale trajectory data, Spark [6], a distributed big data processing engine, has been the first choice for its flexible data organization and in-memory computation. Spark has witnessed great success in big data processing, which include both low query latency and high analytical throughput contributed by its distributed memory storage and computing framework, and good fault tolerance contributed by data reconstruction ability based on the dependency between RDD (Resilient Distributed Datasets). But for a distributed computing environment, data distribution is an important factor for processing performance. A good data partition will enhance the performance of Spark.

Furthermore, there are a variety of queries on large-scale trajectory data, such as range query, trajectory similarity query, SO (Single Object)-based query [7–9], KNN (K Nearest Neighbor)-based query [7,10,11], etc. When processing query requests in a distributed environment, a common optimization mechanism includes the following phases, partitioning, local and global indexing, and querying. Partitioning is a key step for the following two phases because it can improve the balancing data distribution, and what is more the partitioning result directly decides the shapes of local and global indexes that have a great influence on the performance of trajectory query. A good partitioning method can improve query performance greatly by making each node with an appropriate size of data block.

Inspired by above observations, we focused on data partition techniques for distributed in-memory environments and proposed a time-based trajectory data partitioning method, which is mainly applied to improve the efficiency of range query of large-scale trajectory data on Spark and has the following advantages,

- avoiding the repartition process of those existing trajectory data when new data arrive by introducing time-based trajectory data distribution mechanism.
- omitting data preprocessing time by adopting reasonable hash strategy to assign trajectory data directly to each node.
- designing and conducting extensive experiments to verify that this proposed partitioning technique makes the range query more efficient than those existing partitioning methods.

2 Related Work

Considering a comprehensive view on query optimization, we review the work related to query optimization in the following three aspects, including query implementation, indexes and partitioning techniques. Especially, we will focus on those related work on distributed environments, such as Spark.

Query Implementation. Multiple query operations on massive trajectory data have been implemented on Spark or integrated with the related platforms extended from Spark. LocationSpark [12] supports the range query and the KNN query for spatial data. [13] implements box range query, circle range query and KNN(only 1NN) query on SpatialSpark. GeoSpark [14] embedds the box range query, the circle range query, KNN query and distance join operation for spatial data. TrajSpark [15] implements SO-based query, STR (Spatio-Temporal Range)-based query and KNN query on large-scale trajectory data. Box range query, circle range query, KNN query, distance join and KNN join are all covered by Simba [16], a trajectory data processing platform evolved from Spark. [17] provides trajectory similarity queries on both the real-world and synthetic datasets.

Indexes. For distributed environments, local indexes, built on slave nodes, and global indexes, working on master nodes, are often constructed to improve query performance. R-tree [18], KD-tree [19] and quadtree [20] are popular index structures for trajectory data. LocationSpark [12] provides a grid and a regional quadtree as the global index, which also permits users to customize local indexes for various application scenarios, such as a local grid index, local R-tree, a variant of quadtree, or an IR-tree. GeoSpark [14] uses grid as the global index and introduces both R-tree and quadtree as the local indexes. R-tree is applied as a local index in Simba [16], and a sorted array of the range boundaries is provided as Simba's global index when indexing one-dimensional data. For multi-dimensional cases, more complex index structures, such as R-tree or KD-tree, can be used for Simba's global index. A two-level $B+$ tree is used for the global index in TrajSpark [15].

Partitioning Techniques. Data partitioning is an important measure for distributed environments to balance the node workload and to improve query performance. There are three kinds of basic partition methods, which are partition on KD-tree, partition on grid and partition on STR(Sort-Tile-Recursive) [21]. Simba applies STR to partition spatial data. [17] also adopts STR partitioning strategy for trajectory data. GeoSpark automatically partitions spatial data by creating one global grid file. In order to partition trajectory data, TrajSpark defines a new partitioner which contains a quadtree or a KD-tree index. In addition, [22] provides a detailed comparison among various partitioning techniques including grid, quadtree, STR, STR+, KD-tree, Z-curve, and Hilbert curve.

3 Problem Statement

This section presents a detailed statement for our problem, including related definitions and notations. Table 1 lists the frequently used notations.

A trajectory is a group of pairs sorted on time, and each pair is composed of three or more parts, namely a coordinate point, a time stamp, a user identifier, etc. A coordinate point is a sampling point from some user's locations at fixed intervals, here, we only consider longitude and latitude that correspond to **traj.locationin** in Table 1. The time stamp stores the sampling time information, which is represented as **traj.time** in Table 1. Obviously, the trajectory data reflect the spatio-temporal information of moving

Table 1. Notations.

Notation	Description
traj	A sampling point from a single trajectory
traj.location	The location information of *traj*
traj.time	The time information of *traj*
n	The number of all trajectory data partitions
par(i)	The i_{th} partition of trajectory datasets$(0 \leq i < n)$
R	The trajectory data set, $R = (traj1, traj2, \cdots)$
Range(Q, R)	Querying all sampling points both in R and covered by the spatial region Q

objects, a trajectory related with *user-x* can be formalized as a sequence of n-tuple, namely $< (location_1, time_1, user - x, \cdots), (location_2, time_2, user - x, \cdots), \cdots, (location_n, time_n, user - x, \cdots) >$, $time_1 < time_2 < \cdots < time_n$.

Definition 1: General range query. Given a spatial region Q and a trajectory data set $R = \{traj_1, traj_2, traj_3, \cdots\}$, a range query, denoted as range(Q,R), asks for all records existing in Q from R. Formally, $range(Q, R) = \{traj | traj.location \in R \wedge cover(traj.location, Q)\}$. $cover(traj.location, Q)$ represents $traj.location$ is an internal point of Q.

The above general range query can be evolved into the following three kinds of queries.

- spatial range query. Equivalent to the general range query and to output those records in the specified region
- temporal range query. A semantics variant of range query and to output those records in the specified time slot
- spatio-temporal range query. A combination of spatial range query and temporal range query and to output those records satisfying both the time constraints and the space constraints

Data partitioning means that a given raw data set is divided into a specified number of blocks according to the specified constraints. The common partition constraints are partition size, loading balance and data locality. Partition size is a primary factor since it is necessary for computing nodes to avoid memory overflow. Data locality and load balancing are key to speeding up query performance. For this work, our primary objective is to make the range queries more efficient by partitioning a given trajectory data set R into n partitions.

4 Partitioning Method

4.1 Common Partitioning Techniques

First, we briefly analysis three kinds of partitioning technologies, Grid-based, STR-based and KD-tree based partition. Unlike quadtree-based partition that needs a merging operation to construct the specified number of partitions, the above three kinds of partition methods can directly divide the trajectory data into a specified number of partitions. Figure 1 presents a simple illustration for these three techniques, where sampling points and partition boundaries are represented as dots and rectangles respectively. For the following discussion, we use a d-dimensional vector to express a sampling point, denoted as $P = (p_1, p_2, \cdots, p_d)$, and let $k = \sqrt[d]{n}$, n is the number of partitions (see Table 1). In addition, let r be an integer, then we have $P/r = (p_1/r, p_2/r, \cdots, p_d/r)$, $P * r = (p_1 * r, p_2 * r, \cdots, p_d * r)$, and $P \pm Q = (p_1 \pm q_1, p_2 \pm q_2, \cdots, p_d \pm q_d)$.

For grid-based partition, it divides the whole region into equal-size cells. First, it finds the minimum value of each dimension in the whole data set and constructs a new d-dimensional data, denoted as $Pmin = (pmin_1, pmin_2, \cdots, pmin_d)$. In the same way, all the maximum values of each dimension can be assembled into $Pmax = (pmax_1, pmax_2, \cdots, pmax_d)$. Then let $e_i = (Pmax_i - Pmin_i)/k, 1 \leq i \leq d$. Finally, we will get n partitions by increasing e_i in the i_{th} dimension in turn. Obviously, grid-based partition provides a good data locality but can not ensure loading balance since it only provide an uniform split on the whole region.

For STR-based partition, the first step, $j = 1$, is responsible for sorting the whole data set in ascending order by all values in the first dimension, and then the sorted data set is uniformly divided into k partitions. Then, $j+1$ is assigned to j and each new partition output in the last step is individually sorted in ascending by all values in the j_{th} dimension, and then each sorted partition is also uniformly divided into k partitions. Repeating the above processing until j is equal to d, you will get n partitions. KD-tree based partition has a similar process with STR-based partition. For j_{th} step, it also sorts each output partition in last step in ascending order by all values in the j_{th} dimension, but only divides each sorted partition uniformly into two new partitions. For each iteration, $(j+1)\%d+1$ is assigned to j. KD-tree based method repeats the above process until the number of the new partitions is n. Compared with grid-based partition, STR-based and KD-tree based partitions have better loading balance.

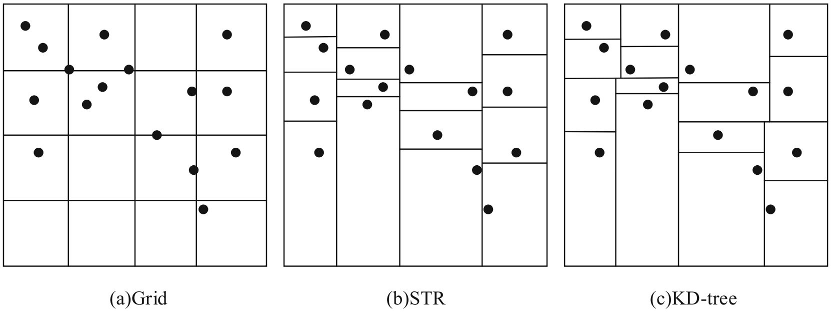

(a)Grid (b)STR (c)KD-tree

Fig. 1. Illustration of different partitioning techniques.

4.2 Time-Based Partitioning Method

The above three partitioning techniques have advantages for queries such as SO (Single Object)-based query, trajectory similarity queries, etc. But range queries often touch a limited number of partitions contributed by the above partition methods because of their data locality, which will cause a bad query loading and resource usage when dealing with large-scale data, especially for continuous range queries.

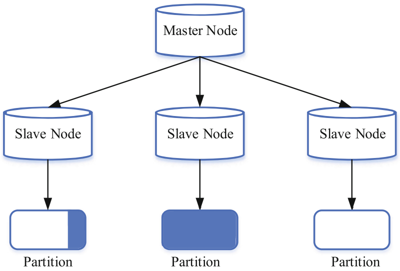


Fig. 2. An example of rang query (the shaded part represents query results).

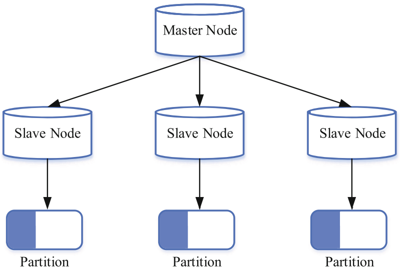


Fig. 3. Ideal partitioning model for range query.

Figure 2 presents an example that most of query results are resided in a single partition. Obviously, each node of the cluster will take a different query time when submitting a same range query. Since the query completion time is determined by the node that takes the longest time, the partition strategy should be improved. Figure 3 presents an ideal partition result, which can balance the query loading well and improve query performance.

In order to achieve the partitioning effect in Fig. 3, we proposed a time-based partitioning method. An item *traj* in the trajectory data set is assigned to the i_{th} partition, the value of i is determined by the formula, $i = traj.time\%n$, here n is the number of all partitions. In order to explain the time-based partitioning technique in detail, Table 2 illustrates an example by a trajectory data set $R = \{traj_1, traj_2, traj_3, traj_4, traj_5, traj_6\}$ and $n = 3$.

Table 2. An example of the time-based partitioning technique.

Record	traj.location (longitude, latitude)	traj.time (ms)	Partition no. (n = 3)
$traj_1$	116.000132, 32.0005	123304	1
$traj_2$	116.000133, 32.0006	123305	2
$traj_3$	116.000134, 32.0007	123307	1
$traj_4$	116.000135, 32.00011	123309	0
$traj_5$	116.000136, 32.0002	123311	2
$traj_6$	116.000137, 32.0004	123312	0

For those sampling points from one trajectory of some user, they are sequential in time and close in space. The above feature decides that our proposed partitioning method does not follow data locality but can present good loading balance when contributing an appropriate size data block for each node. Depending on the time information, the proposed method provides a reasonable distribution of trajectory data for range queries. At the same time, one single trajectory of some user can be viewed as a continuous polygonal line, which ensures that those records related with this trajectory can be distributed into each node approximately evenly and also ensures to let more nodes take the query loading and provide a more quick query response.

When partitioning the trajectory data set, the total processing time is a factor that should be considered. At present, the processing time are mainly composed of preprocessing time and partitioning time. The preprocessing time is consumed to decide the corresponding boundary of each partition, and the partitioning time corresponds to the time cost of loading data into each partition. For those existing partition techniques, the preprocessing time is more longer than the partitioning time because the preprocessing often involves a large number of sorting. One big advantage of our proposed partitioning method is to use a hash operation based on the time information to assign data into different partitions, which only needs negligible preprocessing time. In addition, the existing partitioning techniques are to establish partitions on the whole data set. When new data arrive or the whole trajectory data set is changed, a repartition process should be started to ensure loading balance. Because the proposed partitioning method is only based on time, it is only responsible for partitioning new data when they arrive, and those existing data are not necessary to be repartitioned.

5 Experiments

5.1 Experimental Setup and Datasets

All experiments were conducted on a cluster with 1 master node and 4 slave nodes. Each node has a dual-core Intel Xeon processors @2.53 GHz and 2.6 GB main memory reserved for Spark. Each node is configured with Red Hat 4.4.7-3 and Spark 1.6.0. All nodes in the cluster are connected by a switcher. R-tree is used as a local index and global indexes are not considered since they have no effect on the verification. We compared partitioning performance with those common methods, including Grid-based partition, STR-based partition, and KD-tree based partition.

In order to verify the validity of time-based partitioning on different trajectory data sets, three real data sets are applied. The first data set is Beijing taxis trajectory data set, which has 21,658,278 trajectories. The second data set is Suzhou taxis trajectory data set, which includes 11,741,688 trajectories. The third data set is provided by Microsoft [23], which has 23,667,828 trajectories. The microsoft trajectory data set was collected by Microsoft's Geolife project, and contributed by 178 users in a period of over four years (from April 2007 to October 2011), which includes not only life routines like going home and going to work but also some entertainments and sports activities, such as shopping, sightseeing, dining, hiking, and cycling.

We introduce three metrics for evaluating the partitioning performance, which are the processing time, the partitioning balance and the query time. The processing time is indicated by both the preprocessing time and the partition time. The partitioning balance is expressed by the number of the trajectory of each partition. In general, when the number of each partition is close to each other, the partition is more balanced. The query time is the completion time of the specified range query.

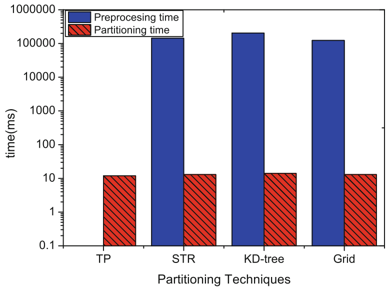


Fig. 4. The partitioning time for Beijing taxi trajectory data

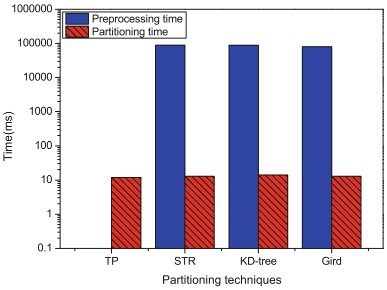


Fig. 5. The partitioning time for Suzhou taxi trajectory data

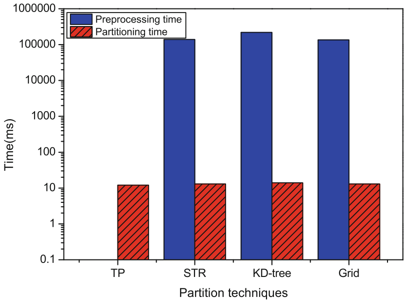


Fig. 6. The partitioning time for Microsoft trajectory data

5.2 Experimental Results and Analysis

Preprocessing and Partitioning Time. The processing time consumed by four partitioning methods on the three data sets are shown in Figs. 4, 5 and 6. Here, TP represents our proposed method, namely time-based partition, STR corresponds to STR-based partition, KD-tree stands for KD-tree based partition, and Grid is just grid-based partition. According to the experimental results, the partition time of each data set contributed by the four partition methods are close and are all less than 15ms, which is because the concrete partition process for each data set is mainly composed of its IO operations. Our proposed method show great advantage on the preprocessing time, which can be negligible. At the same time, the processing time consumed by KD-tree based and STR-based method are longer than grid-based method, which is because both KD-tree and STR have an additional cost on sorting while grid only needs to traverse data one time.

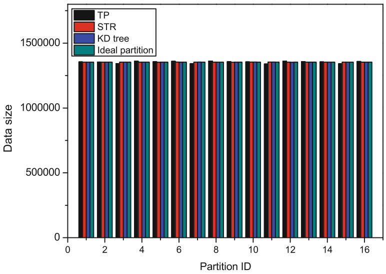


Fig. 7. The partitioning balance for Beijing taxi trajectory data

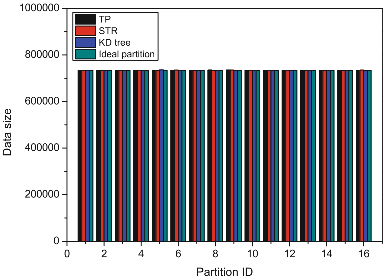


Fig. 8. The partitioning balance for Suzhou taxi trajectory data

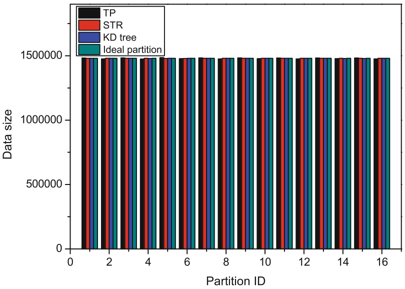


Fig. 9. The partitioning balance for Microsoft trajectory data

Partitioning Balance. This group of experiments are designed for verifying the balance of data partition, and we use an ideal partition method, namely partition by uniform distribution, as a baseline. A total of 16 partitions are designed in our experimental platform. Figures 7, 8 and 9 present the number of trajectory records in each partition for the three data sets. The experimental results show that KD-tree contributed the best partitioning balance, followed by STR. Our proposed method has similar partitioning balance with STR-based partition and KD-tree based partition, all of three methods are close to the state of the ideal partition. We use the formula,$100\% - (|x - y|)/y$, as a quantifiable indicator for partitioning balance, in which x is the size of a specified partition output by some partitioning method and y is the average partition size contributed by the ideal partition method. The minimum values for time-based partitioning on three data sets are 99%, 99.8%, 99.7% respectively. But grid-based partition is extremely unbalanced to partition the trajectory data, whose partitioning results on those three data sets are listed in Table 3 separately. Most of records are allocated on a limited number of partitions while other partitions are empty.

Table 3. The partitioning results of grid-based method on data sets

Data set	Partition size (16 partitions)
Beijing taxi data set	72068,0,0,0,0,0,0,0,0,0,0,84,0,0,0,11669536
Suzhou taxi data set	5,3,2,6,1109,0,0,0,21657150,0,0,0,2,0,0,1
Microsoft data set	271342,33852,194094,23168539,0,0,0,0,0,0,0,0,0,0,0,1

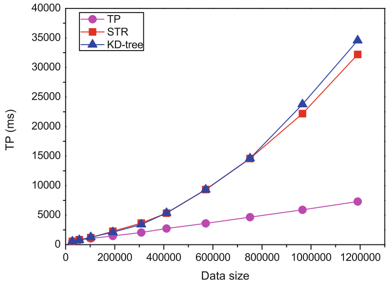


Fig. 10. The query time for Beijing taxi trajectory data

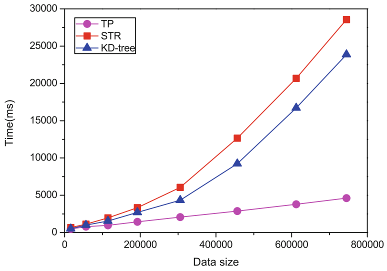


Fig. 11. The query time for Suzhou taxi trajectory data

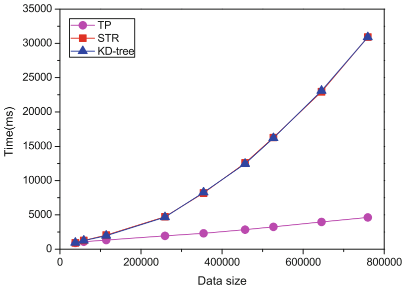


Fig. 12. The query time for Microsoft trajectory data

Query Time. In this section, we designed a group of experiments to test the query performance contributed by different partitioning methods. Considering grid-based partition is extremely uneven and is not suitable for the required queries, it is not included for comparison. We varied the number of the records for output to design different experiments. Figures 10, 11 and 12 show the query time for three partitioning methods under different number of query outputs and on those three data sets. Our proposed method outperformed STR-based and KD-tree based methods in all queries, STR-based method and KD-tree based method present close performance since they own the similar partition mechanism. The Spark cluster has a total of 8 core, time-based partition almost made range queries work on all 8 cores at the same time, while STR-based partition and KD-tree based partition sometimes made queries work on only one core. According to the experimental results, the maximum efficiency of time-based partition can reach to 6.5 times that of the other two partitioning techniques.

6 Conclusions

In this paper, we present time-based partitioning method to optimize range query on large-scale trajectory data. The proposed method breaks the data locality but provides better loading balance. Compared with those existing partitioning techniques, time-based partition needs less preprocessing time and avoids repartitioning process when new data arrive. Extensive experiments show the effectiveness of the proposed method, including providing loading balance and improving range query performance.

Acknowledgments. This study is supported by the National Natural Science Foundation of China (No. 61462017, 61363005, U1501252), Guangxi Natural Science Foundation of China (No. 2017GXNSFAA198035), and Guangxi Cooperative Innovation Center of Cloud Computing and Big Data.

References

1. Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., Huang, Y.; T-drive:driving directions based on taxi trajectories. In: Proceedings SIGSPATIAL, pp. 99–108 (2010)
2. Zheng, Y.: Trajectory data mining: an overview. ACM Trans. Intell. Syst. Technol. **6**(3), 41 (2015). Article 29
3. Xiao, X., Zheng, Y., Luo, Q., Xie, X.: Inferring social ties between users with human location history. J. Ambient Intell. Hum. Comput. **5**(1), 3–19 (2014)
4. Bao, J., He, T., Ruan, S., Li, Y., Zheng, Y.: Planning bike lanes based on sharing-bikes' trajectories. In: Proceedings SIGKDD, pp. 1377–1386 (2017)
5. Zheng, Y., Xie, X., Ma, W.Y.: GeoLife: a collaborative social networking service among user, location and trajectory. IEEE Data Eng. Bull. **33**(2), 32–39 (2010)
6. Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M.J., Shenker, S., Stoica, I.: Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. In: Proceedings NSDI, pp. 15–28 (2012)

7. Lange, R., Drr, F., Rothermel, K.; Scalable processing of trajectory-based queries in space-partitioned moving objects databases. In: Proceedings SIGSPATIAL, 10 p. (2008). Article 31
8. Ma, Q., Yang, B., Qian, W., Zhou, A.: Query processing of massive trajectory data based on mapreduce. In: Proceedings CIKM, pp. 9–16 (2009)
9. Tan, H., Luo, W., Ni, L.M.: CIoST: a Hadoop-based storage system for big spatio-temporal data analytics. In: Proceedings CIKM, pp. 2139–2143 (2012)
10. Wang, H., Zheng, K., Zhou, X., Sadiq, S.W.: SharkDB: an in-memory storage for massive trajectory data. In: Proceedings SIGMOD, pp. 1099–1104 (2015)
11. Nishimura, S., Das, S., Agrawal, D., Abbadi, A.E.: MD-hbase: design and implementation of an elastic data infrastructure for cloud-scale location services. Distrib. Parallel Databases **31**(2), 289–319 (2013)
12. Tang, M., Yu, Y., Malluhi, Q.M., Ouzzani, M., Aref, W.G.: Locationspark: a distributed in-memory data management system for big spatial data. Proc. VLDB **9**(13), 1565–1568 (2016)
13. You, S., Zhang, J., Gruenwald, L.: Large-scale spatial join query processing in cloud. In: Proceedings ICDE Workshops, pp. 34–41 (2015)
14. Yu, J., Wu, J., Sarwat, M.: Geospark: a cluster computing framework for processing large-scale spatial data. In: Proceedings SIGSPATIAL, 4 p. (2015). Articles 70
15. Zhang, Z.G., Jin, C.Q., Miao, J.L., Yang, X.L., Zhou, A.Y.: TrajSpark: a scalable and efficient in-memory management system for big trajectory data. In: Proceedings APWeb-WAIM, Part I, pp. 11–26 (2017)
16. Xie, D., Li, F.F., Yao, B., Li, G., Zhou, L., Guo, M.: Simba: efficient in-memory spatial analytics. In: Proceedings SIGMOD, pp. 1071–1085 (2016)
17. Xie, D., Li, F., Phillips, J.M.: Distributed trajectory similarity search. Proc. VLDB **10**(11), 1478–1489 (2017)
18. Guttman, A.: R-trees: a dynamic index structure for spatial searching. In: SIGMOD, pp. 47–57 (1984)
19. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Commun. ACM **18**(9), 509–517 (1975)
20. Finkel, R.A., Bentley, J.L.: Quad trees: a data structure for retrieval on composite keys. Acta Inf. **4**, 1–9 (1974)
21. Leutenegger, S.T., Lopez, M., Edgington, J. et al.: STR: a simple and efficient algorithm for r-tree packing. In: Proceedings ICDE, pp. 497–506 (1997)
22. Eldawy, A., Alarabi, L., Mokbel, M.F.: Spatial partitioning techniques in spatial Hadoop. Proc. VLDB **8**(12), 1602–1605 (2015)
23. Zheng, Y., Zhang, L.Z., Xie, X., Ma, W.Y.: Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings World Wild Web, pp. 791–800 (2009)

Evaluating Review's Quality Based on Review Content and Reviewer's Expertise

Ju Zhang[1], Yuming Lin[1](✉), Taoyi Huang[1], and You Li[2]

[1] Guangxi Key Laboratory of Trusted Software,
Guilin University of Electronic Technology, Guilin 541004, Guangxi, China
ymlin@guet.edu.cn

[2] Guangxi Key Laboratory of Automatic Detecting Technology and Instruments,
Guilin University of Electronic Technology, Guilin 541004, Guangxi, China

Abstract. User reviews, containing a wealth of user opinion information, play an important role for product's online word of mouth, which have great reference value for potential customers and service/product providers. But the problem of information overload caused by the massive reviews makes users difficult to find high-quality reviews effectively. Most current methods of evaluating review quality focus on review's content. However, the reviewer's expertise also has a positive effect on evaluation of review's quality. In this paper, we propose a new method to rank the reviews according to their quality. Firstly, reviewer's quality of special topic is measured based on his/her historical review data with a topic model. Then, the coverage of attributes described in review content are integrated to measure the review's quality based on a learning to rank model. A series of experiments are implemented on a real world dataset to verify the proposed method's effectiveness.

Keywords: Review quality · Content modeling · Reviewer modeling
Topic modeling · Ranking learning · Quality assessment

1 Introduction

With the rapid development of e-commerce, more and more users like to share their opinions and experiences on shopping websites such as Amazon[1] and Ebay[2]. Since review is rich in user's personal opinion, it is of great value for potential customer and manufacturer. The former can know about the product's performance, quality and users' experiences by reading the according reviews, they would make a reasonable purchase decision further. The latter can know about not only the deficiencies on his product, but also the costumers' requirements, which would be helpful to improve his products' quality. However, the number of

[1] www.amazon.com.

[2] www.ebay.com.

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 36–47, 2018.
https://doi.org/10.1007/978-3-319-91455-8_4

reviews in shopping sites is huge, which easily leads to information overloading of users and makes it difficult for users to quickly find the high-quality reviews. Thus, it brings urgent need for evaluating the reviews' quality automatically.

There are some previous works on this topic. Review contents were treated as independent texts to extract relevant features for predicting the quality of reviews in [1–4]. However, in addition to the textual content, there is more information available for this task. Lu et al. and Zhou et al. used user identities and social network information as features or regularization constraint to improve the prediction of review quality in [5,6] respectively. But their methods can only use user quality when the website contains user social networks. Besides, these methods have ignored the reviewer's expertise, which would make a positive effect on the review's quality. For example, a review on iphone written by a reviewer who often writes reviews on electronics could tend to be with high quality. In order to accurately evaluate the reviews' quality, we introduce the reviewer's expertise to evaluate the reviews' quality based on the content-based features. The existing methods use both user's historical reviews and user's ratings to calculate reviewer's expertise. What's more, we note that helpful votes have an impact on reviewer's expertise, where helpful votes indicate a shopping website's long term review for a given reviewer's expertise level under a specific topic. Reviewers with high expertise tend to receive high helpful votes for their reviews. These motivate us to exploit both the user ratings and the helpful votes to get reviewer's expertise.

In this paper, we propose a method to evaluating the reviews' quality by integrating review's content and reviewer's expertise. Firstly, reviews, product category information, helpful votes and rating deviation are used to measure the reviewer's expertise based on a topic model. Then, content-based features like the coverage of product attributes contained in review are extracted from the review content. Finally, a learning to rank model is trained to rank the reviews according to their quality.

This paper proceeds as follows. In Sect. 2, we briefly introduce the related work. In Sect. 3, we define the ranking problem on review quality. In Sect. 4, we propose a Topic-biased Reviewer's Expertise model to measure reviewer's expertise. Section 5 introduces content-based features and their extraction methods. Experimental results are presented in Sect. 6. Section 7 concludes this paper and discusses direction for future work.

2 Related Work

To evaluate the quality of the review, researchers systematically analyzed the factors that influence the quality of the reviews from different perspectives, and verified the validity of the results through a large number of experiments. Kim et al. [1] automatically evaluated the helpfulness of reviews by using textual features of structure, lexical, syntactic, semantic, and metadata. Experiments showed that review length, unigrams and product ratings were the key features in determining the quality of reviews. Liu et al. [2] depended on the reviewers' professional knowledge, writing style and the timeliness of the reviews to evaluate the quality of the reviews. These factors reflected that the high quality of the reviews catered to the users' preferences. Lu et al. [3] analyzed the quality of the review from user preference, they used needs fulfillment, information reliability and mainstreaming opinion to evaluate review's quality. The experimental results showed that the user-preference features had better effect on the quality of the review. Yang et al. [4] considered review quality as an intrinsic attribute of review texts. It used linguistic and psychological lexicons to represent the review content as semantic dimension features: LIWC and INQUIRER, which demonstrated that semantic features were more experimental than structure, unigrams, and emotional features, and the quality assessment model was more accurate and transferable. This type of work treated reviews as independent text, and extracted the textual features to assess the quality of the reviews, but the accuracy of such methods is greatly compromised when spammers are involved.

Based on the analysis of the text features of reviews, Tang et al. [5] and Lu et al. [6] used both user identities and social network information as a feature or regularization constraint to improve the prediction of review quality. Experiments showed that the accuracy of text-based classifiers was low when using a small amount of training data, the use of regularization in social networks greatly improved the accuracy of model prediction. However, this kind of method is not universal applicability when user has no social network. From different angle, Zhou et al. [7] proposed a reputation-based algorithm which used the overall deviation between user ratings and product ratings to calculate user reputation and used user reputation as a weight to eliminate the impact of poor ratings on the rating system. However, the above method did not consider the impact of user expertise on user quality. Li et al. [8] considered that different reviewers have different educational background, knowledge, and expertise. Experiments verified that user quality fluctuates between topics but is more stable within a topic. For this reason, they took the user expertise into consideration, proposing a topic-biased model based on user reputation model. The experimental results showed that the proposed model was more accurate than the current user reputation model. But it did not consider the impact of the number of helpful votes, where the number of helpful votes indicates the degree of acceptance of the review from other reviewers. Liu et al. [9] built Topic Expertise Model for users, which used Q & A information and tag information to obtain the user's topic distribution, and then used the helpful votes to calculate the user expertise and recommend the high-level expertise users to the questions. But this model did not consider the impact of the other reviewers' quality under the same product.

In summary, based on the features of the content, we use user ratings and helpful votes to calculate a topic-biased reviewer's expertise in order to achieve a more accurate evaluation of the review quality.

3 Problem Definition

A given set of products $P = \{p_1, p_2, ..., p_n\}$, where a review set $R_i = \{r_{i1}, ..., r_{im}\}$ about each product p_i is associated with the corresponding label set $y_i = \{y_{i1}, ..., y_{im}\}$, m is the total number of reviews of product p_i, r_{ij} is the j-th review of product p_i, $y_{ij} = 5 \cdot \lfloor \frac{helpful_+}{helpful_+ + helpful_-} \rfloor$ is a label of the quality level of r_{ij}, where $helpful_+$ is the number of helpful votes for r_{ij}, $helpful_-$ is the number of unhelpful votes for r_{ij}, y_{ij} represents the discrete quality score from 0 to 5. Each review can be expressed by the feature vector $\overrightarrow{x}$, the review set can be expressed as $x_i = \{\overrightarrow{x}_{i1}, ..., \overrightarrow{x}_{im}\}$, the definition of the ranking model is as follows:

$$h : X \to Y \tag{1}$$

where X is the feature vector space for all reviews, and Y is the space in which all permutations of the reviews may be composed. The ranking model outputs the review score $\overrightarrow{s_i}$ based on the set of given feature vectors and sort by it. Assuming that the parameter of the ranking model is $\overrightarrow{\omega}$, which can be expressed as

$$h(\overrightarrow{\omega}, x) \tag{2}$$

where x is the set of feature vector $\overrightarrow{x}$ of all reviews. The ranking model consists of scoring function $f(\overrightarrow{\omega}, \overrightarrow{x})$ and sort function *sort*. Where f grades each review, and the *sort* is used to rank the output in descending order based on the score of each review. In the training phase, the review score $\overrightarrow{s_i} = f(\overrightarrow{\omega}, \overrightarrow{x_i})$ is obtained from the scoring function f, and the optimization parameter $\overrightarrow{\omega}$ is generally learned by optimizing an objective function that measures the correctness of the output $\overrightarrow{s_i}$, which is called the loss function and can be defined as

$$\sum_{i=1}^{n} D(P(\pi|f(\overrightarrow{\omega}, \overrightarrow{x_i}))||P(\pi|\overrightarrow{y_i})) \tag{3}$$

Where D is the K-L divergence, the probability of the original entire review list can be approximated by the permutation probability of the top k terms, then the Top-K probability is as follows:

$$P(\pi|\overrightarrow{s_i}) = \Pi_{j=1}^{k} \frac{exp(s_{i\pi^{-1}}(j))}{\sum_{u=j}^{n} exp(s_{i\pi^{-1}}(u))} \tag{4}$$

where π represents one possible permutation of the review, $\pi^{-1}(j)$ represents the j-th review in π, and each $\frac{exp(s_{i\pi^{-1}}(j))}{\sum_{u=j}^{n} exp(s_{i\pi^{-1}}(u))}$ is the conditional probability. $P(\pi|\overrightarrow{y_i})$ represents the distribution function calculated from the real annotation result. Finally, the ranking model is obtained by optimizing the loss function (3) which uses a neural network-based back propagation algorithm. The focus of this article is to extract features which affect the reviews' quality, so parameter learning will not be described further.

4 Topic-Biased Reviewer's Expertise Assessment

Reviewer's expertise is measured not only by the degree to which a user ratings are close to the products' "true" scores, but also by the number of helpful votes received from other users. If a user ratings always deviate from product'

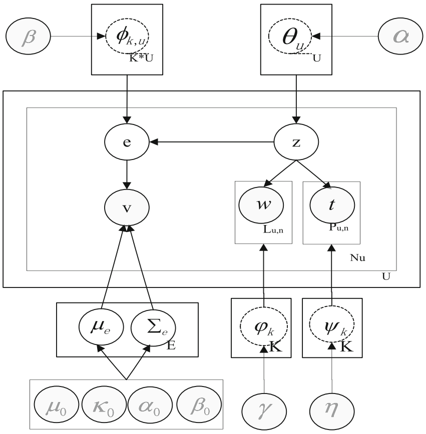


Fig. 1. Topic-biased reviewer's expertise model

"true" scores, his/her expertise is low, in contrast, his/her expertise is high; if a user's review receives more helpful votes, his/her expertise is high, in contrast, his/her expertise is low. So we can model user historical reviews and product category information to get user topic distribution, and calculate the topic-biased reviewer's expertise by making use of the review's helpful votes and user ratings deviation.

4.1 Topic-Biased Reviewer's Expertise Based on Helpful Votes

In this paper, the model of reviewer's expertise is built using reviewer historical reviews, product category information and helpful votes to calculate the reviewer's expertise under different topics. The model is shown in Fig. 1, the symbols and their descriptions are shown in Table 1. Where reviewer "topical expertise" e is the level of quality of a user u under a topic z, to model this information, we assume there exist E expertise levels, each with a Gaussian distribution on vote scores. The more helpful votes, the higher reviewer's expertise and the higher mean is in the Gaussian distribution. We use the Gibbs sampling model to get the following parameters:

$$\mu_e = \frac{\kappa_0 \mu_0 + n_e \overline{v_e}}{\kappa_0 + n_e} \tag{5}$$

$$\Sigma_e = \frac{\alpha_0 + \frac{n_e}{2}}{\beta_0 + \frac{\sum_{v:\{v_i\}_{e_i=e}} \sqrt{(v-\overline{v_e})}}{2} + \frac{\kappa_0 n_e \sqrt{(\overline{v_e}-\mu_0)}}{\kappa_0 + n_e}} \tag{6}$$

$$\theta_{u,k} = \frac{C_u^k + \alpha}{\sum_{k=1}^{K} C_u^k + K\alpha} \tag{7}$$

$$\psi_{k,t} = \frac{C_k^t + \eta}{\sum_{t=1}^{T} C_k^t + T\eta} \tag{8}$$

Table 1. Notations and descriptions

Symbol	Description
U	The total number of users
N_u	The total number of review of user
$L_{u,n}$	The total number of words in u's n-th review
$P_{u,n}$	The total number of product category info in u's n-th review
K	The total number of topics
E	The total number of reviewer's expertise
T	The total number of unique category info
V	The total number of unique words
μ	Mean of Gaussian distribution
Σ	Precision of Gaussian distribution
w, t, v, e, z	Label for word, category info, vote, quality, topic
W, T, V, E, Z	Vector for word, category info, vote, quality, topic
θ_u	User specific topic distribution
$N(\mu_e, \Sigma_e)$	Quality specific vote distribution
ψ_k	Topic specific category info distribution
φ_k	Topic specific word distribution
$\phi_{k,u}$	User topical quality distribution
$\alpha, \beta, \eta, \gamma$	Dirichlet priors
$\alpha_0, \beta_0, \mu_0, \kappa_0$	Normal-Gamma parameters
$Ng(\alpha_0, \beta_0, \mu_0, \kappa_0)$	Normal-Gamma distribution

$$\varphi_{k,w} = \frac{C_k^w + \gamma}{\sum_{w=1}^{V} C_k^w + V\gamma} \tag{9}$$

$$\phi_{k,u,e} = \frac{C_{k,u}^e + \beta}{\sum_{e=1}^{E} C_{k,u}^e + E\beta} \tag{10}$$

Where $\alpha, \beta, \eta, \gamma, \kappa_0, \mu_0$ are priori distributed parameters, usually given in advance. n_e represents the total number of votes for the reviewer's expertise is e and $\overline{v_e}$ represents the average number of votes with reviewer's expertise e. v, w, t are known variables, representing the number of helpful votes, the words in review text, and the words in product category information, respectively. C_u^k represents the number of times that topic k is assigned to user u, C_k^t represents the number of times the word t is assigned to topic k, C_k^w represents the number of times that the word w is assigned to topic k, $C_{k,u}^e$ represents the number of times the quality e is assigned to topic k of user u. The quality of user u_i under topic z is

$$CScore_{z,u_i} = \sum_{e=1}^{E} \phi_{z,u_i,e} \cdot \mu_e \tag{11}$$

and the expertise of u_i under the product p_j is defined as:

$$C_{ij}^1 = sim(r_{ij}, u_i).\sum_{z=1}^{Z} CScore_{z,u_i} = (1 - JS(\theta_{ij}, \theta_{u_i})) \cdot \sum_{z=1}^{Z} CScore_{z,u_i} \tag{12}$$

Where r_{ij} represents the review of user u_i on product p_j, $sim(r_{ij}, u_i)$ represents the similarity of the topic distribution between the review r_{ij} and user u_i, $JS(.)$ represents the JS-divergence, θ_{u_i} represents the topic distribution of user u_i, which can be obtained directly from the model; θ_{ij} represents the topic distribution of review r_{ij}, which can be obtained from its prior distribution,

$$\theta_{ij,z} = \theta_{u_i,z} \sum_{w:w_{r_{ij}}} \varphi(z,w) \sum_{t:t_{r_{ij}}} \psi(z,t) \tag{13}$$

Where w and t denote the words of review and the words of product category information, $\varphi(z,w)$ and $\psi(z,t)$ are directly obtained from the model.

4.2 Topic-Biased Reviewer's Expertise Based on Rating Deviation

Rating deviation refers to the rating deviation between reviewer ratings and "true" ratings of product. However, the "true" ratings does not exist in the rating system, so we can use a weighted mean of all reviewers' rating under the product as the "true" rating. If a reviewer ratings always deviates from the "true" rating of the product, the reviewer's expertise will be lower; on the contrary, the reviewer's expertise will be higher. The given rating set $F_{ij}(u_i \in U, p_j \in P)$, for the topic vector Z, the topic distribution of each product denotes as $B_{|P|\times|Z|}$, and b_{jk} represents the distribution of the product p_j belonging to topic z_k. The reviewer's expertise under different topics is $C_{|U|\times|Z|}$, c_{ik} is the expertise of user u_i under topic z_k, then c_{ik}^s is the expertise of user u_i under topic z_k after s iterations, and F_j^s is the "true" rating of the product p_j after s iterations, then we have,

$$F_j^{s+1} = \frac{1}{|U_j|} \sum_{u_i \in U_j} F_{ij}(\sum_{z_k \in Z} c_{ik}^s b_{jk}) \tag{14}$$

$$c_{ik}^{s+1} = 1 - \frac{\lambda \sum_{p_j \in N_i} b_{jk} |F_{ij} - F_j^{s+1}|}{\sum_{p_j \in N_i} b_{jk}} \tag{15}$$

Where $\lambda \in (0,1)$ is the damping factor, U_j is the set of users for product p_j, N_i is the set of products which are included in the reviews written by user u_i, and $b_{jk} = \frac{\sum_{u_i \in U_j} \theta_{ij,z_k}}{|U_j|}$, where θ_{ij,z_k} is calculated from the formula (12). When $F_j^{s+1} - F_j^s < \tau(\tau$ as a threshold), the iteration is over, and the reviewer's expertise based on the rating deviation can be expressed as:

$$C_{ij}^2 = \sum_{z_k \in Z} c_{ik}^s b_{jk} \tag{16}$$

5 Content-Based Review Quality Assessment

The review text is rich in user viewpoint and has a good guiding role for users and manufacturers. The analysis of review text is extracting the structural feature, unigram feature and the coverage of product attributes from the content. From [1], structural feature observes the structure and format of review text, and unigram feature is the $tf - idf$ statistic of each frequent word occurring in a review. Product attributes are specific description of the product performance, the higher coverage of the product attributes with opinion words in reviews, the higher review quality. However, different users' attention to the attributes of the products will be different because of the user's needs. In order to provide users with the key information of the product, we'll give different weights to the product attributes by using the frequency of the attributes appearing in the product reviews and the frequency appearing in the similar products. It is assumed that the review set R_i of the product $p_i \in P$ covers a set of product attributes A, $R_a \subseteq R_i$ describes that the review contains the attribute $a \in A$. $P' \subseteq P$ represents a set of similar products that the product p_i belongs to, and some reviews of the product $P'_a \subseteq P'$ refer to the attribute a. The weight $w(a)$ of the attribute a can be defined as:

$$w(a) = \frac{|R_a|}{|R_i|} \cdot \frac{|P'_a|}{|P'|} \tag{17}$$

Where $|R_i|$ is the number of reviews of the product p_i, $|R_a|$ is the number of reviews which contain product attribute a, $|P'_a|$ is the number of products with the same attribute a, and $|P'|$ is the number of similar products. Assuming that one review r contains a set of product attributes $A_r \in A$, the product attributes coverage of the review r may be defined as:

$$Cov(r) = \frac{\sum_{a \in A_r} w(a)}{\sum_{a' \in A} w(a')} \tag{18}$$

6 Experimental Results and Analysis

6.1 Dataset and Evaluation Measures

The experimental dataset is one part of Amazon's 1996–2014 reviews [10], where each review contains user ID, product ID, review text, rating, review time, helpful votes and other information. Due to a large number of repetitive reviews on e-commerce websites such as Amazon, a simple method which filters redundant reviews is required for this. It matches the Bigram repetition rate between two reviews, if the repetition rate exceeds 80%, the review is marked as a duplicate review. At the same time, there are many duplicate products in the Amazon.com, such as the same kind of product with different colors. For duplicate products, their reviews are often repeated. Therefore, if a product's reviews overlap with reviews of other products, only one product with more reviews is remained.

In addition, in order to ensure the performance of the ranking system, we delete the reviews which number of votes less than 5. After processing, the final dataset contains 83,904 reviews, 37,214 users and 29576 products. In the experiment, a 5-fold cross-validation is used to verify the experimental results, 80% of the reviews are training sets and the remaining 20% are test sets.

In the field of information retrieval, $NDCG$ is used to evaluate the ranking quality of the pseudo-correlation feedback returned by the retrieval system. We adopt $NDCG$ (Normalized Discounted Cumulative Gain) to evaluate the rank performance about the review quality. $NDCG$ at position n is as follows:

$$NDCG@n = \frac{DCG@n}{IDCG@n} = \frac{\sum_{i=1}^{n} \frac{2^{r(w_i)}-1}{\log(1+i)}}{IDCG@n} \tag{19}$$

Where $NDCG@n$ represents the ranking quality evaluation of the top n reviews in the ranking list, w_i represents the i-th review, $IDCG@n$ is the ideal ranking result of the top n reviews in the product. $r(w_i)$ is the quality level of w_i, Where w_i is the same as y_{ij} in Sect. 3. Simultaneously, we use Pearson and Spearman rank correlation coefficients to measure the correlation between the model's ordered list of reviews and the ground-truth ordered list of reviews. The larger the rank correlation coefficient is, the stronger the correlation is.

6.2 Results and Analysis

In our experiments, we totally built 4 ranking models, which are as follows:

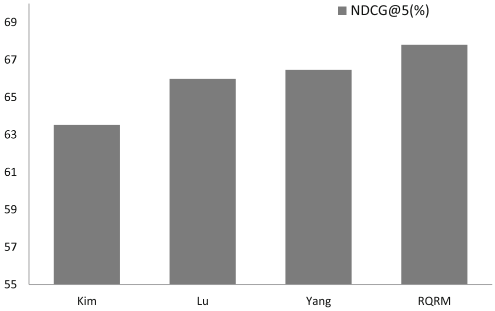


Fig. 2. The NDCG@5 of different review quality ranking models

Kim [1]: a ranking model whose features are extracted from reviews: structure, lexical, syntactic, semantic, metadata, uses a learning to rank model ListNet [11] to learn an ordered list of reviews quality.

Lu [3]: a ranking model whose features are extracted from the review: capture rate of needs fulfillment, volition and tense for reliability and the divergence from mainstreaming opinion, uses a learning to rank model ListNet to learn an ordered list of reviews quality.

Yang [4]: a ranking model whose features are extracted from the review content: structural features, unigram features and semantic, uses a learning to rank model ListNet to learn an ordered list of reviews quality.

RQRM: a ranking model uses a learning to rank model ListNet to learn an ordered list of reviews quality, which features are composed of reviewer's expertise which is extracted from Sect. 4 and content-based features which is extracted from Sect. 5.

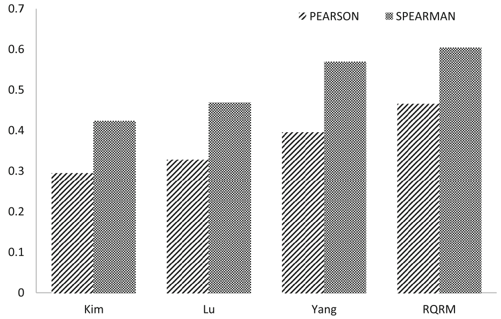


Fig. 3. The correlation coefficients of different review quality ranking models

As shown in Fig. 2, we use NDCG@5 to evaluate the performance of rank model. It can be seen from the figure that the RQRM has the best ranking performance. This model uses features which are based on review content and reviewer's expertise. Its performance is 1.34% and 4.27% higher than that of Yang and Kim, respectively. And in Fig. 3, RQRM can provide a rank list of reviews with a higher correlation coefficient than other methods in terms of relevance criteria. Its Pearson and Spearman rank correlation coefficients is 0.17 and 0.18 higher than that of Kim, respectively.

In addition, we delete the different features based on the RQRM to determine the impact of the feature on the ranking performance. For ease of description, "AC" indicates that the coverage of product attribute modified by opinion words; "STR" indicates structural features; "UNI" indicates unigram features; "UC_H" indicates reviewer's expertise based on helpful votes; "UC_R" indicates reviewer's expertise based on rating. Related experimental results are shown in Fig. 4. The "RQRM-*" represents the model which deletes the "*" feature based on the RQRM, where "*" indicates "AC", "STR", "UNI", "UC_H", and "UC_R" feature.

As can be seen from Fig. 4, the performance of RQRM is higher than that of other rank models. The result shows that the quality of review is not only related to review content but also to the users who wrote the review. In addition, the performance of "RQRM-UC_H" and "RQRM-UC_R" decrease by 3.24% and 2.13% respectively compared with that of RQRM. The result shows that helpful votes and user ratings can accurately predict the quality of users to a certain extent;

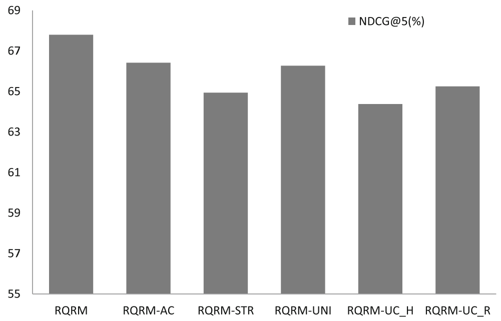


Fig. 4. The influence of different features on the quality of reviews

However, the performance of "RQRM-UC_R" is higher than that of "RQRM-UC_H", and the result shows that the helpful votes can predict the quality of users more accurately, it may be due to inconsistencies between user ratings and review content [12], and reviewer's expertise calculated by user ratings deviation cannot accurately reflect the true expertise of reviewers.

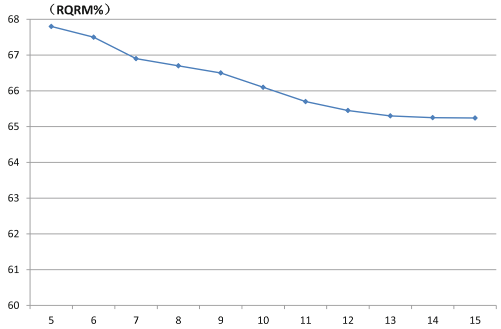


Fig. 5. Ranking accuracies in terms of NDCG@n on RQRM model

NDCG@n represents the NDCG of the top n of the ranking list, where each product contains at least n reviews, so in the test set we strictly require the product to contain at least n reviews. In Fig. 5, we use the different n to evaluate the RQRM ranking performance. From the figure, we have the best performance of the RQRM ranking model when n = 5. Therefore, we use NDCG@5 to evaluate the ranking performance.

7 Conclusion

In order to solve the problem of obtaining high-quality review quickly and accurately. We propose a method to evaluating the review's quality by integrating review's content and reviewer's expertise, where a topic-biased reviewer's expertise is calculated using reviewer's historical review data. This method can analyze reviewer's expertise when there is no social network for reviewers. Experimental results show that the proposed review quality rank model improves the performance of the Kim model by 4.27%. In order to obtain a rank model with good performance, more labeled samples are often needed, but labeling samples needs to consider the labeling cost. In the future work, reducing the cost of labeling will be the issue we need to consider.

Acknowledgements. The work is supported by National Natural Science Foundation of China (Nos. 61562014, U1501252, U1711263), the Guangxi Natural Science Foundation (No. 2015GXNSFAA139303), the project of Guangxi Key Laboratory of Trusted Software, the project of Guangxi Key Laboratory of Automatic Detecting Technology and Instruments (No. YQ17111), the general Scientific Research Project of Guangxi Provincial Department of Education (No. 2017KY0195).

References

1. Kim, S.M., Pantel, P., Chklovski, T., Pennacchiotti, M.: Automatically assessing review helpfulness, pp. 423–430 (2006)
2. Liu, Y., Huang, X., An, A., Yu, X.: Modeling and predicting the helpfulness of online reviews, pp. 443–452 (2008)
3. Hong, Y., Lu, J., Yao, J., Zhu, Q., Zhou, G.: What reviews are satisfactory: novel features for automatic helpfulness voting, pp. 495–504 (2012)
4. Yang, Y., Qiu, M., Yan, Y., Bao, F.S.: Semantic analysis and helpfulness prediction of text for online product reviews, vol. 2, pp. 38–44, January 2015
5. Lu, Y., Tsaparas, P., Ntoulas, A., Polanyi, L.: Exploiting social context for review quality prediction, pp. 691–700 (2010)
6. Tang, J., Gao, H., Hu, X., Liu, H.: Context-aware review helpfulness rating prediction, pp. 1–8 (2013)
7. Zhou, Y., Lei, T., Zhou, T.: A robust ranking algorithm to spamming. EPL **94**(4), 1034–1054 (2011)
8. Li, B., Li, R.H., King, I., Lyu, M.R., Yu, J.X.: A topic-biased user reputation model in rating systems. Knowl. Inf. Syst. **44**(3), 581–607 (2015)
9. Yang, L., Qiu, M., Gottipati, S., Zhu, F., Jiang, J.: Cqarank: jointly model topics and expertise in community question answering, pp. 99–108 (2013)
10. He, R., Mcauley, J.: Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering, pp. 507–517 (2016)
11. Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F., Li, H.: Learning to rank: from pairwise approach to listwise approach, pp. 129–136 (2007)
12. Zhang, R., Gao, M., He, X., Zhou, A.: Learning user credibility for product ranking. Knowl. Inf. Syst. **46**(3), 679–705 (2016)

Tensor Factorization Based POI Category Inference

Yunyu He[1], Hongwei Peng[1], Yuanyuan Jin[1], Jiangtao Wang[1](✉),
and Patrick C. K. Hung[2]

[1] National Trusted Embedded Software Engineering Technology
Research Center (NTESEC), East China Normal University, Shanghai, China
{yyhe,yyj}@stu.ecnu.edu.cn, penghongwei_phw@163.com,
jtwang@sei.ecnu.edu.cn

[2] Faculty of Business and Information Technology,
University of Ontario Institute of Technology (UOIT), Oshawa, Canada
patrick.hung@uoit.ca

Abstract. Trajectory data is an important kind of data with different aspects of the user information like demographics, user behavior and activities. Therefore, it is significant and essential to infer point-of-interests (POI) categories from trajectory data for user modeling and user preferences mining in many location-based services (LBS). Recent researches focus more on recommendation and prediction of next POI, which are based on the check-in data. Check-in data is only a partial aspect of the user's behavior which collected by a certain LBS, while trajectory data describes the user from all around, which can help modeling user's interest preferences in a great degree. However, due to a deviation between the GPS-coordinate and the actually visited location, it is significant to infer the ultimate POI categories people accessed from trajectory data instead of mapping location coordinates to POIs directly. In this paper, we propose a collaborative inferring framework to analyze the actually visited POI categories from users' historical trajectory data. Through modeling relationships among the user, time and POI category, the tensor decomposition method can effectively complement the missing data and provides accurate predictions when user trajectory data is absent. Extensive experiments have been conducted with various state-of-the-art baseline on real-world trajectory data, and experiment results have demonstrated the promising performance in this framework.

1 Introduction

With the increasing popularity of GPS equipped mobile devices and vehicles, geographical records have become prevalent on the web. These geographical records not only reflect the user's mobility patterns, but also provide some help for modeling the user's interest preferences through the point-of-interests (POI) user visited. Besides, it is significant to mine users' preferences [9,17] and establishing user profile from user's geo-spatial data for various location-based services (LBS),

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 48–63, 2018.
https://doi.org/10.1007/978-3-319-91455-8_5

such as personalized advertising services, urban planning and location-based recommendation services. Some recent researches focus on recommendation POI [10,18], which also consider the user's interest preferences. But these all based on the check-in data that only relates to a certain LBS, such as Facebook, Twitter, Foursquare, etc. Only when using these platforms to share their experience and check-in information associated with a POI, the location data will be collected by these platforms. They can only model the user's interest preferences and make recommendation based on the certain platform, but not suitable for modeling user profile. Compared with the check-in data, the trajectory data can track the user's behavior better and record the user's access information more completely. It could be concluded that the trajectory data has a greater value to estimate the POIs user has accessed, which is the key step of modeling user profile and can provide better help for follow-up work.

However, the number of POIs is substantially larger than the number of POI categories in the whole city map. Compared with POIs, the POI categories can perform the common characteristics of users' interest offering more help for user profile. Thus, the POI category are chosen in this work to represent user's interest preference, which can better reflect similar behavior patterns among users.

To this end, the crucial problem solved in this paper is how to infer the POI categories visited by each user from his or her trajectory data. By our best knowledge, it is a challenging research task due to following reasons:

First of all, there is usually a large deviation between the GPS-coordinate and the actual visited POI category. In our daily life, the location acquisition device is not always close to the user within a small range. For example, if a user wants to eat in a restaurant next to a supermarket parking lot he/she chooses to park the car, it is obvious that the supermarket parking lot is not really the POI category user has visited.

What's more, the POI categories user accessed may suffer from data sparsity for representing human mobility among a certain period, given that few people are willing to along with this device all the time or to use the device every day.

To cope with these challenges, some related methods that can be used to made some exploration. Recent studies have also found that human mobility follows a high degree of regularity over time [5]. Based on these observations, a collaborative method called tensor factorization is adopted, which can learn a universal model for solving this problem from users' trajectory data, instead of learning a separate model for each user isolated. Although, the tensor factorization considering the global factor collaboratively, it will ignore the specific geographical factors for each stopping point. Yi et al. [19] defined *negative-unlabeled* (NU) learning problem which can be used to take advantage of the specific geographical situations for each stopping point. However, this work conduct a matrix decomposition losing a part of latent factors of the tensor and reducing accuracy to some extent.

Considering the advantages and disadvantages of tensor factorization and negative-unlabeled learning problem, we propose a novel collaborative framework to solve the POI categories inferring problem. First, hidden relationships among users, time slots and POI categories are integrated into a three-dimensional tensor $\mathcal{X}$ and conduct a tensor factorization method with Tikhonov Regularization. This tensor factorization method helps to overcome sparsity problem of data, meanwhile it complements the missing data. So that, the framework can make accurate predictions when user trajectory data is absent. Then, negative-unlabeled constraints are adopted to make use of the specific geographical situations for each staying point by normalizing the probability of the POI categories within the candidate set. At last, an efficient alternating minimization algorithm is employed to combine these two method and solve the problem.

To summarize, the major contributions of this paper are:

1. We develop a collaborative method under negative-unlabeled constraints to model relationships among user, time and POI category, which overcomes the data sparsity problem and infers POI categories accurately.
2. The proposed collaborative inferring framework can complement the missing data and make accurate predictions when user trajectory data is absent.
3. Based on two real-word dataset collected, the extensive experiments has been conducted to validate the effectiveness of the proposed approach. The results show that our approach vitally outperforms baseline models with a significant margin in several scenarios.

The rest of this paper is structured as following: Sect. 2 introduces related work. Data preprocessing and problem definition is presented in Sect. 3. The collaborative inferring framework under negative-unlabeled constraints will be proposed in Sect. 4. Experimental results based on a large-scale dataset are presented in Sect. 5. Finally, the conclusion of the paper will be drawn in Sect. 6 with a brief discussion of limitations and directions of future research.

2 Related Work

In order to introduce the related work of this paper in a more orderly way, this section is divided into two parts: tensor factorization and user profile on trajectory data.

2.1 Tensor Factorization

Comparing with many other collaborate methods, tensor has great advantages. As a generalization of matrices, tensor can model correlations among more than two dimensions while matrix factorization methods can only apply to two-order data. In recent years, tensor factorization has been widely applied in a variety of fields. Narita et al. [14] and Ge et al. [4] focus on how to utilize auxiliary information improve the quality of tensor decomposition. Zhong et al. [22] extracted feature from heterogeneous data set based on tensor factorization to infer user profiles. Yi et al. [19] focus on the similar problem with this paper which learned mobile users' location categories from highly inaccurate mobility data and proposed NUTF. NUTF treat this problem as a NU learning problem which assigned the non-zero probabilities with any non-negative values that sum up to one, then optimized objective function by a low-rank tensor factorization. But there is a trade off between accuracy and efficiency of the algorithm. The NUTF decomposed the tensor by unfolding it into a two-dimension matrix and adopted a randomness matrix decomposition [6], which could reduce the complexity but loss a part of latent factors of the tensor as well as accuracy to some extent.

2.2 User Profile on Trajectory Data

Since former researches seldom focused the same task, we review three widely studied researches contributing to user profile on trajectory data:

User's Important Locations Detection: These researches identify important locations (e.g., home or working place) to understand users' behavior at these locations. Cao et al. [1] proposed a framework which can extract staying points from each users' GPS data and then clustering them to find significant semantic locations. An unsupervised collaborative approach is applied in Liu et al. [12] to identify home and working locations of individuals from geo-spatial trajectory data which also define the user-location signatures to describe users' behavior at the location.

Interesting Regions Discovery: Zheng et al. [21] means the culturally important places and then models multiple individuals' location histories to mine interesting locations. Van Canh et al. [16] discovered regional communities by exploiting methods based on spatial latent Dirichlet allocation (SLDA). Yuan et al. [20] discovers regions of different functions using both human mobility and POIs located in a region. These works try to learn semantics of regions so that it can do help for user preference mining, as they are meaningful for finding user communities rather than modeling individuals' profile.

POI Prediction: Feng et al. [3] predicts which POI the user will visit at next time through historical check-in records. Li et al. [10] recommends POIs in location-based social networks (LBSN) by uncovering potential check-in information primarily based on Matrix Factorization. Like most recommendation systems, they are all based on actual check-in dataset collected from a certain LBSN. From this perspective, the user preference based on above studies can only describe the interest user perform on the certain LBSN rather than a comprehensive profile of the person.

3 Preliminaries

We first introduce the negative-unlabeled constraints, and then formally define the POI category inferring problem for trajectory data.

3.1 Problem Formulation

As shown in Fig. 1, the brown circle draws the certain range out and all the POI categories in the circle (i.e. theater, hotel, restaurant and mansion) will be defined as the candidate set of the GPS coordinate. Indeed, user's true visited POI must be within a certain range of the coordinate updates and the POI categories out of the range is impossible to be visited.

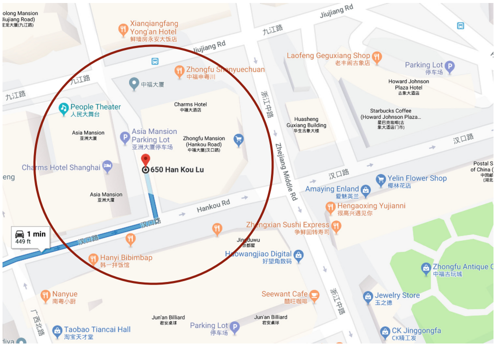


Fig. 1. The blue line illustrates the trajectory of vehicle, the red mark means where the car is parked. (Color figure online)

The collection vehicle trajectory data is in the form: <vehicle id, time stamp, location coordinates, vehicle state[1]>. Our ultimate goal is to infer the probability of POI category user visited after getting off his or her car. To achieve the goal, we first preprocess the raw data by several key steps:

1. Extract users' significant visiting points and dwelling time by filtering vehicle state.
2. Normalize time into time slots with the consideration of the proposition that human mobility would follow a high degree of regularity over time.
3. Match GPS coordinates with POI information. For the staying point in each time slot, the possible POI categories considered can be within an uncertain range of the update GPS coordinates. More details will be explained later in the experiments described in Sect. 5.

Let $I_{ut:}$ be the indicator of POI possible categories the user n visited during the time slot t. In this paper, we formally define the POI category inferring problem as follows:

Definition *(POI category inferring):* Given a targeting user n, a observation time slot t, a candidate set $I_{ut:}$ of POI category, the POI category inferring problem is to predict the value of probability $\mathcal{X}_{utc} \in [0, 1]$. Specifically, the value

[1] The state flag illustrates whether the vehicle is running or stopping.

$\mathcal{X}_{utc}$ more close to one means that the user n has greater chance to visit the category c within time slot t.

The Table 1 lists the notations and their meanings used in this paper.

3.2 Negative-Unlabeled Constraints

Based on the precondition mentioned above, possible POI categories must be within an candidate set of the staying point extracted from the vehicle trajectory data. The possible POI categories contribute a candidate set. In addition, there can be only one actual category of visited POI for a user at one time. Therefore, the probabilities of POI categories in the candidate set sum up to one.

If we treat this problem as a two-class learning problem, we can easily label the POI categories out of range as negative class, leaving the categories in the candidate set unlabeled.

Table 1. Notation and description

Notation	Description
U, T, C	Users set, time lots set, category set
u, t, c	User id, time lot id, POI category
$\mathcal{X}$, $\mathcal{Y}$	POI probability tecnsor
$\mathcal{X}_{utc}$	The probability of user u visiting POI category c at time slot t
I_{utc}	Indicates whether POI category c is in the candidate set of user u and time t
R	The number of latent factors

Under this scenario, we turn to negative-unlabeled learning, which is a counterpart concept to positive-unlabeled learning (PU), to solve this problem. PU [11] also called learning from positive and unlabeled examples, which aims to build a binary classifier to classify the test set containing positive and unlabeled examples into two classes. The research works [2,7] propose semi-supervised PU learning methods to take advantage of the unlabeled data, which contains the instances belonging to the predefined class rather than the labeled categories. On the opposite, negative-unlabeled learning problem samples all the labeled examples from the negative class while the unlabeled examples come from both negative and positive classes. Accordingly, our collaborative inferring model is developed under the negative-unlabeled constraints.

4 Collaborative Inferring Framework

4.1 Tensor Factorization Using Tikhonov Regularization

To make use of the collaborative capabilities among users, times and categories and to complete missing data at the same time, we propose to estimate the possibility of different POI categories based on tensor models.

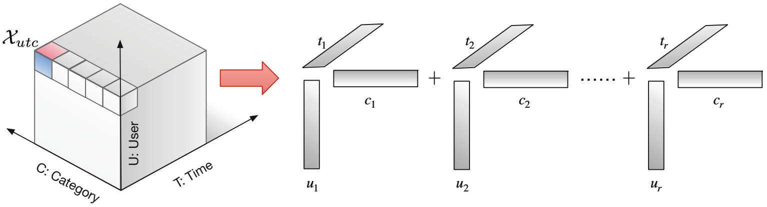


Fig. 2. Tensor decomposition model.

As shown in Fig. 2, we use the tensor $\mathcal{X}$ to represents the observed data. The three-ways of $\mathcal{X} \in \mathbb{R}^{U,T,C}$ represent users, time slot and POI category respectively, where each element $\mathcal{X}_{utc} \in [0,1]$ represent the probability of user u visiting POI category c during time slot t. U, T, C respectively denoting the number of user, time slot and POI category.

In order to capture the common behavioral characteristics of user, each day is segmented into several time bins, which results in $\mathcal{X}$ being sparse and low-rank. Similar to matrix factorization, tensor factorization can decompose a tensor into the sum of several rank-one tensors that can best approximates the given tensor. This paper will build our model on a CANDECOMP/PARAFAC (CP) decomposition model and can be represented as follow:

$$\mathcal{X} \approx \sum_{r=1}^{R} u_r \circ t_r \circ c_r \tag{1}$$

Where u_r, t_r, c_r are latent vectors of size $U \times 1$, $T \times 1$, and $C \times 1$ respectively, $R \leqslant min\,\{U, T, C\}$ is the number of latent factors as the rank of a tensor, and the symbol "$\circ$" stands for the outer product. More specifically, $\mathcal{X}$ is approximately equal to the sum of R tensors.

$$\underset{\mathcal{X},\mathcal{Y} \in \mathbb{R}^{U,T,C}}{minimize} \|\mathcal{X} - \mathcal{Y}\|_F^2 \tag{2}$$

The vectors u_r, t_r, c_r have be collected in latent factor matrices U, T, C for user, time and category, i.e. $U = [u_1, u_2 \cdots u_R]$, which are of sizes $U \times R$, $T \times R$, and $C \times R$, respectively. With these definitions, it can also be represented in matrix form:

$$\begin{aligned} \mathcal{X} &\approx [\![\text{U,T,C}]\!] = \sum_{r=1}^{R} u_r \circ t_r \circ c_r \\ \mathcal{X}_{(1)} &= U(T \odot C)^{\mathsf{T}} \\ \mathcal{X}_{(2)} &= T(C \odot U)^{\mathsf{T}} \\ \mathcal{X}_{(3)} &= C(U \odot T)^{\mathsf{T}} \end{aligned} \tag{3}$$

The symbol "$\odot$" denotes the Khatri-Rao product [2] and the $\mathcal{X}_{(i)}$ means the model - i unfolding of tensor $\mathcal{X}$.

[2] Khatri-Rao product of matrices A and B with k columns, given by $A \odot B = [a_1 \otimes b_1 \ a_2 \otimes b_2 \cdots a_k \otimes b_k]$, where $\otimes$ denotes Kronecker product.

Further more, to avoid overfitting and to provide a unique solution, Tikhonov regularization terms is added with the regularization parameter $\lambda_U, \lambda_T, \lambda_C > 0$ to the objective function. Thus, the goal of tensor decompose problem can be represented by the following optimization problem:

$$min \left\| \mathcal{X} - [\![\text{U,T,C}]\!] \right\|_F^2 + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_T}{2} \|T\|_F^2 + \frac{\lambda_C}{2} \|C\|_F^2 \tag{4}$$

Where the symbol "−" denotes the element-wise subtraction (which computes a tensor with each element equals $\mathcal{X}_{utc} - \mathcal{Y}_{utc}$) and "$\|\cdot\|_F$" indicates the Frobenius Norm of Tensor (similar to matrix) which is defined as: $\|\mathcal{X}\|_F = \sqrt{\sum_{u=1}^{U} \sum_{t=1}^{T} \sum_{c=1}^{C} \mathcal{X}_{utc}{}^2}$.

To solve the above optimization problem, we chose the alternating least square (ALS) algorithm, which is commonly used for CP decomposition. It works by iteratively optimizing one parameter while leaving the others fixed (i.e. fixes T and C to update U) on the base of Eqs. 5, 6 and 7, until meeting the convergence condition.

$$U = \mathcal{X}_{(1)}(T \odot C) \left[(T \odot C)^{\intercal}(T \odot C) + \lambda_U I_R\right]^{\dagger} \tag{5}$$

$$T = \mathcal{X}_{(2)}(U \odot C) \left[(U \odot C)^{\intercal}(U \odot C) + \lambda_T I_R\right]^{\dagger} \tag{6}$$

$$C = \mathcal{X}_{(3)}(U \odot T) \left[(U \odot T)^{\intercal}(U \odot T) + \lambda_C I_R\right]^{\dagger} \tag{7}$$

Where I_R is the unit matrix of size $R \times R$ and $[\cdot]^{\dagger}$ means generalized inverse matrix.

In this part, tensor decomposition method model the hidden relationships among users, time slots and POI categories, and generate collaborative latent factors. It helps to relieve sparsity problem of data and complement the missing data, which provides the Collaborative Inferring Framework the capability to make accurate predictions when user trajectory data is absent.

4.2 Collaborative Inferring Framework Under Negative-Unlabeled Constraints

For the POI category inferring problem, it is not sufficient to utilize only collaborative latent factors obtained from the tensor decomposition, because it only take the global factor into consider. To take advantage of the specific geographical factors for each staying point, the tensor is required to satisfy the negative-unlabeled constraints. For each user u and time slots t, there can be only one actual category of visited POI meanwhile several possible POI categories. Therefore, the possible POI categories contributes a candidate set and the probabilities of them should be sum up to one. Meanwhile, it is impossible to visit the POI categories out of candidate set of every staying point for user u and time slots t. Similar to [19], the NU constraints under our problem definition can be given as following:

$$\begin{cases} \mathcal{Y}_{utc} \geq 0, \forall u, t, c \\ \mathcal{Y}_{utc} = 0, \forall u, t, and\ c \notin I_{ut:} \\ \mathcal{Y}_{ut:}{}^{\intercal} I_{ut:} = 1, \forall u, t \end{cases} \tag{8}$$

Algorithm 1. Projection vector under NU constraints

```
Input: $\mathcal{X}$,$\mathcal{I}$
Output: $\mathcal{Y}$
for $\forall$ $u$, $t$ do
    initial $v \in \mathcal{X}_{ut:}$
    sort $v$ in the desending order
    $j = max\left\{c \in [\mathcal{I}_{ut:}] \mid v_c + \frac{1}{c}(1 - \sum_{i=1}^{c} v_i) > 0\right\}$
    $\rho = \frac{1}{j}\left(1 - \sum_{i=1}^{j} v_i\right)$
    $\mathcal{Y}_{ut:} \leftarrow s$ s.t. $s_i = max\left\{v_i + \rho, 0\right\}, i \in [\mathcal{I}_{ut:}]$
end for
```

The category latent vector is projected onto the probability simplex for each user u and time slot t to achieve the goal. Only the categories among candidate set $I_{ut:}$ can be calculated and the value of them are sum up to one, while the other categories not included in the candidate set $I_{ut:}$ are assigned to zero. As described in Algorithm 1., the project algorithm can be efficiently computed in $O(|I_{ut:}| \times \log |I_{ut:}|)$ time and perform better than other normalization method i.e. *softmax function* in the case of similar vector values. So far, we have considered the specific geographic information around each staying point, which improve the accuracy of inferring the POI categories.

Combining tensor decomposition and NU constraints, the final model in this paper can be expressed as follows:

$$\underset{\mathcal{X},\mathcal{Y} \in \mathbb{R}^{U,T,C}}{minimize} \left\|\mathcal{X} - \mathcal{Y}\right\|_F^2 \tag{9}$$

$$\begin{aligned} s.t. \quad & min \left\|\mathcal{X} - \hat{\mathcal{X}}\right\|_F^2 + \frac{\lambda_U}{2}\left\|U\right\|_F^2 + \frac{\lambda_T}{2}\left\|T\right\|_F^2 + \frac{\lambda_C}{2}\left\|C\right\|_F^2 \\ & where \quad \hat{\mathcal{X}} = [\![\mathrm{U,T,C}]\!] \\ & \begin{cases} \mathcal{Y}_{utc} \geq 0, \forall u, t, c \\ \mathcal{Y}_{utc} = 0, \forall u, t, and\ c \notin I_{ut:} \\ {\mathcal{Y}_{ut:}}^{\top} I_{ut:} = 1, \forall u, t \end{cases} \end{aligned} \tag{10}$$

In order to solve the collaborative inferring framework efficiently, we employ an alternating minimization scheme that iteratively updates one of $\mathcal{X}$ and $\mathcal{Y}$ and minimize their difference.

As shown in Algorithm 2, the learning strategy is summarized through alternating least square (ALS). First, to initialize the output tensor $\mathcal{Y}$ and the number of iterations. And then the tensor decomposition with tikhonov regularization procedure is described from lines 2 to 8. It is important to note that after updating all the three latent matrix, it is demanded to update the three unfolding of tensor $\mathcal{X}$ in each round of decomposition iteration. After the tensor decomposition procedure reaches the convergence condition, lines 9–10 expound projection procedure to meet the negative- unlabeled constraints by using $\mathcal{X}$ generated in last procedure as input.

Algorithm 2. Solving CFNU via ALS

```
Input: $\mathcal{X}, \lambda_U, \lambda_T, \lambda_C$
Output: $\mathcal{Y}$
Initialize: $\mathcal{Y} \leftarrow \mathcal{X}$, iter $= 0$
while Not Converged and $iter \leq I_{max}$ do
    repeat
        $\mathcal{X} \leftarrow \mathcal{Y}$
        Update $\tilde{U}, \tilde{T}, \tilde{C}$ according to Equation (5), (6), (7)
        Update $\mathcal{X}$ by update each unfolding with $\tilde{U}, \tilde{T}, \tilde{C}$ according to Equation (3)
    until Converged
    Compute $\mathcal{Y}$ according to Algorithm 1
end while
return $\mathcal{Y}$
```

5 Experiments

In this section, we conduct experiments to validate the effectiveness of the proposed framework for inferring the actual POI category that user visited. Concretely, the experiments aim at answering following questions:

1. How effective is the proposed method compared with alternative state-of-the-art methods on inferring POI categories from trajectory data?
2. How do the parameters contribute to the inferring accuracy? That is to say, it is needed to give special care for tuning the approach, or is there a wider choice of parameters leading to high robustness?
3. How is the data complementing ability? Can the framework effectively predict POI categories when user GPS-coordinate is absent?

5.1 Datasert

Two real-world datasets are evaluated: electric vehicle trajectory data[3] and DianPing check-in data. All the methods are run on the same machine with an Intel Core 2.90 GHz CPU and 16 GB RAM of a single-thread for fair comparison.

The trajectory data set is sampled from fifty electric vehicles in Shanghai between 1 June 2015 and 31 December 2015. When the vehicle is used, various types of information such as local time, GPS coordinates, vehicle states, travelled distance, running speed, etc. are uploaded every 30 to 50 s. In this study, we conduct three pretreatment steps on the raw data in order to apply the framework: (1) We first extract meaningful staying points from the raw trajectory data by estimating the dwell time. The dwelling time users spend on the staying points can be calculated easily based on local time, GPS coordinates and vehicle states. In this work, the threshold of dwelling time is half an hour. (2) Then, each day is split into 7 time bins as following: 0am–7am,

[3] provided by Shanghai EV data platform: http://www.shevdc.org.

7am–10am, 10am–13pm, 13pm–16pm, 16pm–19pm, 19pm–22pm, 22pm–24pm. The non-uniform scheme is chosen since there is little activity during the early morning. (3) In the end, we convert the GPS coordinates associated with POI categories and build the candidate set via Baidu map API[4]. Specifically, 28 POI categories of Baidu map hierarchy are chosen, including parking lot, hospital, school, governmental agency, tourist attraction, etc. Finally, each staying point can be represented by a tuple <user id, time bins id, category candidate set>.

DianPing data set, which contains over three hundred thousand check-in records from 2756 users and 22212 POIs corresponding to 66 categories over the whole 2014, is also evaluated. Each check-in includes a user ID, a time stamp, a POI ID, and the category of the POI.

5.2 Experimental Setup and Metrics

To investigate the quality of the proposed framework, we adopt the Accuracy@k as evaluation metric. Since there is only one true category for every user during a time slot, precision and recall are essentially equal in this situation.

Accuracy@k represents the percentage of correct category emerging in the top-k predictions and is calculated as:

$$Accuracy@k = \frac{\sum_{u=1}^{|U|} \sum_{t=1}^{|T|} \left| S_{u,t} \bigcap \tilde{S}_{u,t} \right|}{\#staying\ points} \tag{11}$$

Where $S_{u,t}$ is the visited categories set observed by the user u at time slot t and $\tilde{S}_{u,t}$ is the predicted value set about user u and time slot t. Besides, $\#staying\ points$ is the total number of staying points and k means the number of predicted values. To achieve the best performance, different k values are picked due to different feature of the two data sets for the parameters. Since experimental results are insensitive to regularization parameters in Tikhonov regularization (Eq. 4), we set $\lambda_U, \lambda_T, \lambda_C = 0.01$.

5.3 Baseline

To the best of our knowledge, there is seldom model directly predicting POI category from human trajectory data. Our collaborative method is then compared with the following baselines.

- *Negative-Unlabeled Tensor Factorization (NUTF)*: This baseline computes user's location categories by unfolding the inferring tensor to a matrix and adopting random SVD algorithm.
- *Non-negative matrix factorization (NMF)*: This baseline [8] is a matrix decomposition method under the condition that all the elements in the matrix are non-negative constraints. It is used to solve Collaborative filtering problems in recommend system [13].

[4] http://lbsyun.baidu.com/.

– *Singular Value Decomposition (SVD)*: We adopt the regularized SVD [15], which is a collaborative filtering algorithm predicting users' preferences for items, to obtain a prediction for POI category.

5.4 Evaluating over Electric Vehicle Trajectory Data

Since the ground truth (the actual POI category visited by electric vehicle user) cannot be obtained from the trajectory data, we make a rule to pick ground truth as the verification set which can evaluate prediction performance. If the categories in candidate set of a staying point are all the same, this category is considered as the true single user visited, which is also called ground truth. For both the verification set and the remaining data set, the noisy data is added by generate categories randomly according to the same user id and time bins id. The ratio of created noise categories to total categories is represented by p. Our experiment also evaluate the influence of p. And then all the three data mentioned above are integrated as the training set to our framework.

Table 2. The performance on trajectory data for accuracy

Method	p = 40%			p = 60%			p = 80%			Avg improvement
	k = 1	k = 3	k = 5	k = 1	k = 3	k = 5	k = 1	k = 3	k = 5	
CFNU	0.3134	0.5993	0.7875	0.3133	0.6242	0.7875	0.3133	0.6242	0.7874	N/A
NUTF	0.2395	0.4954	0.5754	0.2762	0.5432	0.7190	0.2967	0.5817	0.6190	18.51%
NMF	0.2057	0.5309	0.6519	0.2385	0.4697	0.6270	0.1581	0.4640	0.5635	38.71%
SVD	0.2312	0.4713	0.5305	0.2779	0.5478	0.7247	0.2968	0.5787	0.7231	18.76%

Table 3. The performance on Check-in data for accuracy

Method	p = 40%			p = 60%			p = 80%			Avg improvement
	k = 1	k = 5	k = 10	k = 1	k = 5	k = 10	k = 1	k = 5	k = 10	
CFNU	0.1342	0.5241	0.8202	0.1319	0.5226	0.8182	0.1330	0.5249	0.8124	N/A
NUTF	0.1244	0.4269	0.6135	0.1124	0.3315	0.5230	0.1068	0.3691	0.5298	35.11%
NMF	0.1135	0.3439	0.5266	0.0793	0.2642	0.4003	0.0520	0.2076	0.3474	93.06%
SVD	0.1634	0.4979	0.7004	0.0741	0.2714	0.4752	0.0522	0.2602	0.4534	64.78%

The model performance in terms of accuracy and improvement are shown in Table 2. For this dataset a fraction of noise data is selected randomly. Then the possibility of each POI category is estimated. The results are reported at the fraction of 40%, 60% and 80%. Based on the results, we can observe that NMF perform worse than other methods in general. It is interesting that SVD and NUTF have similar performance on this dataset. The random svd method is also used in the NUTF, and NUTF can not effectively improve the accuracy with a small number of users, they have almost the same accuracy on this datasets.

Our approach CFNU (short for Collaborative Inferring Framework under Negative-Unlabeled Constraints) outperforms baseline methods and achieves an average improvement of 18.51% relative to NUTF when k equals to 1, 3 and 5. These results likely due to the lack of exploitation of the potential relevance between time, user and category.

On the other hand, with the increase of fraction p, NMF shows a significant decrease while our method still maintains the high accuracy. A possible reason is that not only the user's preference but also the time latent factors have been learned in the training, so that the possibility on the target POI category can be accurately predicted. This consequence also answers the question 2 raised at beginning of this section, and the parameter p has little influence on the accuracy of the proposed model.

5.5 Evaluating over Check-In Data

To further illustrate the effectiveness of our approach, DianPing check-in data set is evaluated. DianPing check-in data is different from general trajectory data, it records the users visited POIs through the web service, that can directly obtain corresponding POI categories. Hence, to simulate the real situation defined to solve in the Sect. 3, we randomly sample 10% of all the check-in records as the validation set and add the noise data in the same way as trajectory data. In addition, the 80:20 among validation set split is chosen to divide training set and testing set for testing complement and prediction capability. More specifically, to evaluate the accuracy of inferring result, only 80% of the validation set is selected, the remaining data (without ground truth) and the created noise data are used as the input of the framework. For evaluating complement and prediction capability, the other 20% of the validation set is divided as the test data without putting into the framework. The result verifies that proposed framework can predict POI category accurately even in the deficiency of check in data.

Firstly, the fraction of noise data is set as $p = 40\%$, 60% and 80% and Table 3 presents the evaluation results for inferring POI categories on check-in data.

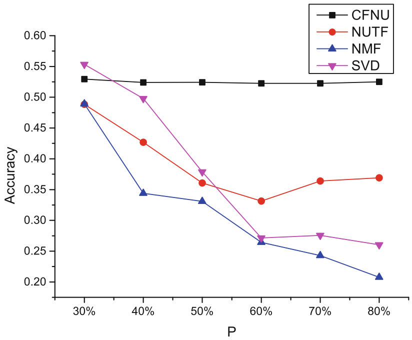
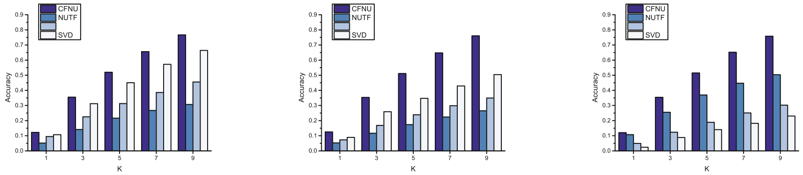


Fig. 3. The performance comparison of basic methods at different p.

It is shown in the table that the proposed model consistently outperforms NUTF, NMF and SVD across k and gives an average improvement of over 30% in terms of accuracy over other alternative methods, demonstrating the effectiveness of our method.

Effects of Noise Proportion: To simulate the real situations, the noise percentage p is set from 30% to 80% with the step of 10%. The accuracy of these models under this scenario are shown in Fig. 3. When the p is at a low percentage, the function of this area is relatively simple, which gives the NUTF, NMF and SVD methods an opportunity to outperform the proposed model. But with the increase of p, the POIs in the region will be more diversified and these methods show a huge downward trend, while our model consistently performs the high accuracy. This result is encouraging since it demonstrates that the proposed framework for POI categories inferring can achieve high robustness in a realistic scenario where varies categories are rounding the staying points. It also proves that the parameter p has little influence on the accuracy, and gives evidence of the high robustness since there is a wider choice of parameter p.



(a) accuracy@p=40% (b) accuracy@p=60% (c) accuracy@p=80%

Fig. 4. The performance of accuracy at p = 40%, 60%, 80%.

Performance of Complement and Prediction: To test the complementing and predictive capability of the model without check-in data, the predictions on the test data (20% of the validation set without putting into the framework) are evaluated. Figure 4 shows the predicted result with varying number of k. SVD achieves a relatively high accuracy when the noise proportion is low, but drops rapidly as the noise ratio increases. However, the proposed method has better performance than others in this scenario regardless of the p, which indicates proposed method can predict POI categories people visiting in the future accurately. This result provides the answer to question 3, and the proposed collaborative inferring method can complement the missing data meanwhile make effectively predictions when user check-in data is absent.

From these results, under various scenarios, our proposed collaborative inferring method consistently performs best among all and outperforms state-of-the-art methods with a significant improvement. The effectiveness, robustness and superiority of the proposed model for the POI category inferring problem is empirically confirmed.

6 Conclusion

Inferring the user's visited POI category is indispensable for modeling the user's interest preferences and establishing user profile. It plays an important role in the location-based service because of the comprehensive, accurate, detailed user portraits for every individual, and can help the system to provide better personalized service. The problem differs with many current POI researches with several new characteristics, which requires the deployment of new models. Our proposed collaborative inferring method exploits the collaborative capabilities among users, time slots and categories. Meanwhile, it effectively alleviates the problem of sparsity and improve the inferring accuracy. Through complementing the missing data using tensor decomposition with Tikhonov regularization, this framework can provide accurate predictions when user trajectory data is absent. Extensive experiments with two real-world data sets have validated the effectiveness of our collaborative inferring model. In our future work, contextual information can be explored to improve the prediction accuracy of current framework further.

Acknowledgments. This work is supported by NSFC grants (No. 61532021), Shanghai Knowledge Service Platform Project (No. ZF1213), SHEITC and Shanghai Agriculture Applied Technology Development Program (No. G20160201). Jiangtao Wang is the corresponding author.

References

1. Cao, X., Cong, G., Jensen, C.S.: Mining significant semantic locations from GPS data. Proc. VLDB Endowment **3**(1–2), 1009–1020 (2010)
2. Fei, G., Liu, B.: Social media text classification under negative covariate shift. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2347–2356 (2015)
3. Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y.M., Yuan, Q.: Personalized ranking metric embedding for next new poi recommendation. In: IJCAI, pp. 2069–2075 (2015)
4. Ge, H., Caverlee, J., Zhang, N., Squicciarini, A.: Uncovering the spatio-temporal dynamics of memes in the presence of incomplete information. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 1493–1502. ACM (2016)
5. Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.L.: Understanding individual human mobility patterns. Nature **453**(7196), 779 (2008)
6. Halko, N., Martinsson, P.G., Tropp, J.A.: Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev. **53**(2), 217–288 (2011)
7. Hu, H., Sha, C., Wang, X., Zhou, A.: A unified framework for semi-supervised PU learning. World Wide Web **17**(4), 493–510 (2014)
8. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems, pp. 556–562 (2001)

9. Lee, R.K.W., Hoang, T.A., Lim, E.P.: On analyzing user topic-specific platform preferences across multiple social media sites. In: Proceedings of the 26th International Conference on World Wide Web, pp. 1351–1359. International World Wide Web Conferences Steering Committee (2017)
10. Li, H., Ge, Y., Hong, R., Zhu, H.: Point-of-interest recommendations: learning potential check-ins from friends. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 975–984. ACM (2016)
11. Li, X.-L., Liu, B.: Learning from positive and unlabeled examples with different data distributions. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 218–229. Springer, Heidelberg (2005). https://doi.org/10.1007/11564096_24
12. Liu, R., Buccapatnam, S., Gifford, W.M., Sheopuri, A.: An unsupervised collaborative approach to identifying home and work locations. In: 2016 17th IEEE International Conference on Mobile Data Management (MDM), vol. 1, pp. 310–317. IEEE (2016)
13. Luo, X., Zhou, M., Xia, Y., Zhu, Q.: An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. IEEE Trans. Industr. Inf. **10**(2), 1273–1284 (2014)
14. Narita, A., Hayashi, K., Tomioka, R., Kashima, H.: Tensor factorization using auxiliary information. Data Min. Knowl. Disc. **25**(2), 298–324 (2012)
15. Paterek, A.: Improving regularized singular value decomposition for collaborative filtering. In: Proceedings of KDD Cup and Workshop, vol. 2007, pp. 5–8 (2007)
16. Van Canh, T., Gertz, M.: A spatial LDA model for discovering regional communities. In: 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 162–168. IEEE (2013)
17. Wan, M., Wang, D., Goldman, M., Taddy, M., Rao, J., Liu, J., Lymberopoulos, D., McAuley, J.: Modeling consumer preferences and price sensitivities from large-scale grocery shopping transaction logs. In: Proceedings of the 26th International Conference on World Wide Web, pp. 1103–1112. International World Wide Web Conferences Steering Committee (2017)
18. Ye, M., Yin, P., Lee, W.C., Lee, D.L.: Exploiting geographical influence for collaborative point-of-interest recommendation. In: Proceedings of the 34th international ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 325–334. ACM (2011)
19. Yi, J., Lei, Q., Gifford, W., Liu, J.: Negative-unlabeled tensor factorization for location category inference from inaccurate mobility data. arXiv preprint arXiv:1702.06362 (2017)
20. Yuan, J., Zheng, Y., Xie, X.: Discovering regions of different functions in a city using human mobility and POIs. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 186–194. ACM (2012)
21. Zheng, Y., Zhang, L., Xie, X., Ma, W.Y.: Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of the 18th International Conference on World Wide Web, pp. 791–800. ACM (2009)
22. Zhong, Y., Yuan, N.J., Zhong, W., Zhang, F., Xie, X.: You are where you go: inferring demographic attributes from location check-ins. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pp. 295–304. ACM (2015)

ALO-DM: A Smart Approach Based on Ant Lion Optimizer with Differential Mutation Operator in Big Data Analytics

Peng Hu[1], Yongli Wang[1](✉), Hening Wang[2], Ruxin Zhao[1],
Chi Yuan[1], Yi Zheng[3], Qianchun Lu[3], Yanchao Li[1],
and Isma Masood[1]

[1] Department of Computer Science and Engineering,
Nanjing University of Science and Technology, Nanjing 210094, China
Yongliwang@njust.edu.cn
[2] School of Foreign Studies, Nanjing University, Nanjing 210023, China
[3] Department of Wireline Product Operation, ZTE Corporation,
Nanjing 210012, China

Abstract. The ant lion optimizer (ALO) is a novel swarm intelligence optimization algorithm, but its population diversity and convergence precision can be limited in some applications. In this paper, we proposed an approach based on ALO and differential mutation operator that called ALO-DM. In this method, differential mutation operator and greedy strategy enhance the diversity of the population. In addition, combining it with data mining algorithms can be useful and practical in big data analytics problems. The simulation results not only show that the ALO-DM is able to obtain accurate solution, but also demonstrate that it is feasible and effective.

Keywords: Big data · Ant lion optimizer · Differential mutation operator
Swarm intelligence

1 Introduction

The big data has increasingly attracted attentions at present. Most of the big data researches not only concern over the huge amount of data, but also focus on handling the high dimensional data and the multiple objectives in solving big data problems. Swarm intelligence optimization algorithm is a collection of nature-inspired searching techniques [1]. It can be used to solve large amount of data, high dimensional data, dynamical data, and multi-objective optimization in big data analytics [2], for example, there are many objectives which need to be satisfied simultaneously in intelligent transport systems, such as environmental pollution, transportation scheduling, and rapid transportation.

In recent years, many swarm intelligence optimization algorithms are proposed, such as Genetic Algorithm (GA) [3], Differential Evolution (DE) [4], Particle Swarm Optimization (PSO) [5], Ant Colony Optimization (ACO) [6], Cuckoo Search (CS) [7] etc. The No Free Lunch Theorem [8] states that all algorithms perform equal when

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 64–73, 2018.
https://doi.org/10.1007/978-3-319-91455-8_6

solving all optimization problems. It means that one algorithm may be very effective to solve certain problems but ineffective in solving other problems. Due to this foundation, many new nature inspired optimization techniques are proposed in literature.

The Ant Lion Optimizer (ALO) is proposed by Mirjalili in [9], it is a novel swarm intelligence optimization algorithm by simulating hunting behavior of ant lions. And it is also concerned by many researchers in different areas of optimization. Yao et al. proposed a dynamic adaptive ant lion optimizer for route planning of unnamed aerial vehicles in [10]. Zawbaa et al. [11] used another variant of ALO named chaotic antlion optimization for feature selection. Emary et al. [12] also applied Binary antlion approaches for feature selection successfully. Rajan et al. [13] developed a weighted elitism based Ant Lion Optimizer to solve optimum VAr planning problem. In this paper, a novel optimization algorithm ALO-DM is proposed. In this method, differential mutation operator and greedy strategy enhance the diversity of the population. The ALO-DM algorithm is validated by 10 benchmark functions. The simulation results demonstrate that this proposed algorithm is feasible and effective.

The remainder of the paper is organized as follows: Sect. 2 describes optimization algorithm ALO-DM. Section 3 introduces the details of the experiments and validation. Finally, conclusion can be found in Sect. 4.

2 The Proposed ALO-DM Algorithm

In this section, the basic concepts of ALO and ALO-DM algorithm are introduced. Though the performance of original ALO algorithm is satisfactory but its population diversity and convergence precision can be limited in some applications. In ALO, as rand walk of ant, some antlions cannot be updated by ant. That means this phenomenon may cause food shortage for the antlions. And Scharf has stated that the antlions relocate in response to food shortage [14]. So this behavior can help antlion get their prey, this is the motivation behind this work.

2.1 The Basic Concepts of ALO

The ALO implemented the idea of random walk of ants around antlions, building traps, entrapment of ants in traps, catching ants and rebuilding traps. In order to model these interactions, we have to convert the aforementioned interactions into equations. Therefore, the random walk of ants' movement can be defined as follows:

$$X(t) = [cum(2r(t_1) - 1), cum(2r(t_2) - 1), \cdots, cum(2r(t_n) - 1)] \tag{1}$$

where *cum* calculates the cumulative sum, *n* is the maximum number of iteration, *t* shows the step of random walk, and *r*(*t*) is a stochastic function defined as follows:

$$r(t) = \begin{cases} 1, & rand > 0.5 \\ 0, & rand \leq 0.5 \end{cases} \tag{2}$$

where *rand* is a random number created with uniform distribution in the interval of [0, 1]. To keep the random walks inside the search space, random walk is normalized using the following equation:

$$X_i^t = \frac{X_i^t - a_i}{b_i - a_i} \times \left(d_i^t - c_i^t\right) + c_i^t \tag{3}$$

where a_{i} is the minimum of random walk of *i-th* variable, b_i is the maximum of random walk in *i-th* variable, c_i^t is the minimum of *i-th* variable at *t-th* iteration, and d_i^t indicates the maximum of *i-th* variable at *t-th* iteration.

Then to model building traps, it is assumed that every ant randomly walks around a selected antlion by the roulette wheel and the elite simultaneously as follows:

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \tag{4}$$

where R_A^t is the random walk around the antlion selected by the roulette wheel at *t-th* iteration, R_E^t is the random walk around the elite at *t-th* iteration, and Ant_i^t indicates the position of *i-th* ant at *t-th* iteration.

In order to imitate the entrapment of ants in traps, the following equations are presented in this regard:

$$c^t = \frac{c^t}{I} \tag{5}$$

$$d^t = \frac{d^t}{I} \tag{6}$$

where I is a ratio, c^t is the minimum of all variables at *t-th* iteration, and d^t indicates the vector including the maximum of all variables at *t-th* iteration. And $I = 10^w t/T$ where t is current iteration, T is the maximum number of iterations, w is a constant based on the value on current iteration ($w = 2$ when $t > 0.1T$, $w = 3$ when $t > 0.5T$, $w = 4$ when $t > 0.75T$, $w = 5$ when $t > 0.9T$, $w = 6$ when $t > 0.95T$).

The final stage of hunt is catching ants and rebuilding traps. The following equation is proposed in this regard:

$$Antlion_j^t = Ant_i^t \; if \, f\left(Ant_i^t\right) > f\left(Antlion_i^t\right) \tag{7}$$

where t shows the current iteration, $Antlion_j^t$ shows the position of selected *j-th* antlion at *t-th* iteration, and Ant_i^t indicates the position of *i-th* ant at *t-th* iteration.

2.2 Antlion's Relocation

In order to model the antlion's relocation capability, the differential mutation operator and greedy strategy is employed. Differential evolution algorithm [4], which is proposed by scholars Storn and Price, it is a random optimization algorithm and it has been successfully applied to many optimization problems. Differential mutation operator is the main core operator of differential evolution algorithm, and its mathematical formula is defined as follows:

$$\begin{aligned} Antlion_i^{t+1} &= Antlion_i^t + (1 - Y) \times \left(Antlion_l^t - Antlion_m^t\right) \\ &+ Y \times \left(Antlion_g^t - Antlion_j^t\right) \end{aligned} \tag{8}$$

$$Y = \frac{t}{T} \tag{9}$$

where t is current iteration, T is the maximum number of iterations, Y is the mutation factor, $Antlion_j^t$ shows the position of selected *j-th* antlion at *t-th* iteration, and *l, m, g, j* are different random number that is not the same as i.

For this process, it is assumed that antlion's relocation occur when antlion's position generated by Eq. (8) becomes fitter than its current position. Here this triggering condition is called greedy strategy. So the antlion's relocation in ALO-DM will help the antlion deal with food shortage, and enhance its chance of catching new prey.

2.3 ALO-DM Algorithm

Specific implementation steps of the ALO-DM algorithm can be summarized in the pseudo code shown as follows:

The ALO-DM algorithm

```
Initialize a population of ants and antlions with random solutions;
Calculate the fitness of all ants and antlions;
Determine the best antlion as the elite;
while the end criterion is not satisfied
    for every ant
        Select an antlion using Roulette wheel;
        Update $c$ and $d$ using Eqs. (5) and (6);
        Create a random walk and normalize it using Eqs. (1) and (3);
        Update the position of ant using Eq. (4);
    end for
    Calculate the fitness of all ants;
    Replace an antlion with its corresponding ant, if it becomes fitter (Eq. (7));
    for every antlion
        Update the position of antlion using Eq. (8) according to the antlion's relocation phase.
    end for
    Update elite if the current best antlion becomes fitter than the elite;
    end while
Return elite
```

3 The Experiments and Validation

In this section, two types of test functions are employed with different characteristics to validate the performance of the ALO-DM algorithm from different perspectives.

3.1 Experimental Setup

All of the algorithms are performed in MATLAB R2012a on Intel(R) Core(TM) i3-4130 Processor (3 M Cache, 3.40 GHz) with 4 GB RAM.

3.2 Benchmark Functions

The test functions (Table 1) are divided into two groups: f_1–f_7 are unimodal functions, f_8–f_{10} are multi-modal functions [15–18]. As their names imply, unimodal test functions have one optimum hence these are highly significant to benchmark exploitation and convergence of the algorithm. Multimodal functions have many optima, and there is only one global optima and many local optima. Mostly it happens that the searching stagnates into the local optima and restricts to reach the global optima. Hence the algorithm should be capable of avoiding local optima. Therefore, exploration and local optima avoidance of algorithms can be tested by the multi-modal test functions. Note that the minima of all the unimodal and multi-modal test functions are equal to 0.

Table 1. Ten benchmark functions.

Function	*Dim*	Range	f_{min}
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	[−100,100]	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[−10,10]	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	30	[−100,100]	0
$f_4(x) = \max_i\{\lvert x_i \rvert, 1 \le i \le n\}$	30	[−100,100]	0
$f_5(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	30	[−30,30]	0
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[−100,100]	0
$f_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	30	[−1.28,1.28]	0
$f_8(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	30	[−5.12,5.12]	0
$f_9(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$	30	[−32,32]	0
$f_{10}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[−600,600]	0

3.3 Results and Discussion

The proposed ALO-DM algorithm compared with optimization algorithms PSO [5], and ALO [9] respectively. In this paper, the results are obtained in 30 independent trials run on the test functions. The Best, Mean, Worst and Std. represent the optimal fitness value, mean fitness value, worst fitness value and standard deviation, respectively. Same as that in literature [9], each of the test functions is solved using 30 candidate solutions (antlions) over 1000 iterations and the results in Table 2. For some functions of f_1–f_{10}, Figs. 1, 2, 3 and 4 are the fitness convergence curves.

Table 2. Simulation results of benchmark functions

Functions	Algorithm	Best	Worst	Mean	Std.
f_1	PSO	2.88E − 17	1.00E + 04	6.67E + 02	2.54E + 03
	ALO	1.48E − 06	2.64E − 05	1.01E − 05	6.35E − 06
	ALO-DM	4.65E − 15	2.40E − 11	4.68E − 12	5.82E − 12
f_2	PSO	1.32E − 07	2.00E + 01	4.67E + 00	6.29E + 00
	ALO	5.18E + 00	1.09E + 02	3.82E + 01	3.33E + 01
	ALO-DM	2.85E − 08	1.24E − 06	5.51E − 07	3.22E − 07
f_3	PSO	1.18E + 01	2.49E + 04	9.25E + 03	5.96E + 03
	ALO	4.53E + 02	2.74E + 03	1.32E + 03	6.41E + 02
	ALO-DM	4.65E − 14	2.42E − 10	6.36E − 11	6.54E − 11
f_4	PSO	0.49E + 00	5.37E + 00	2.11E + 00	1.28E + 00
	ALO	7.41E + 00	2.66E + 01	1.65E + 01	4.71E + 00
	ALO-DM	9.91E − 09	7.76E − 07	2.48E − 07	2.05E − 07
f_5	PSO	0.65E + 00	9.00E + 04	3.15E + 03	1.64E + 04
	ALO	7.48E + 00	1.86E + 03	2.05E + 02	3.97E + 02
	ALO-DM	0.03E + 00	2.82E + 01	5.64E + 00	1.13E + 01
f_6	PSO	2.79E − 17	6.17E − 13	5.78E − 14	1.32E − 13
	ALO	5.51E − 07	3.40E − 05	9.63E − 06	7.96E − 06
	ALO-DM	6.85E − 05	2.78E − 04	1.57E − 04	5.60E − 05
f_7	PSO	1.60E − 01	5.55E + 00	9.16E − 01	1.06E + 00
	ALO	6.11E − 02	3.41E − 01	1.61E − 01	6.35E − 02
	ALO-DM	5.55E − 06	2.61E − 04	1.03E − 04	7.85E − 05
f_8	PSO	4.78E + 01	1.76E + 02	9.99E + 01	3.42E + 01
	ALO	3.48E + 01	1.52E + 02	6.77E + 01	2.75E + 01
	ALO-DM	0.00E + 00	8.19E − 12	9.95E − 13	1.67E − 12
f_9	PSO	6.58E − 09	1.44E + 01	2.32E + 00	2.51E + 00
	ALO	3.52E − 04	1.33E + 01	2.15E + 00	2.47E + 00
	ALO-DM	1.22E − 08	8.12E − 07	2.20E − 07	2.01E − 07
f_{10}	PSO	0.00E + 00	9.05E + 01	6.04E + 00	2.30E + 01
	ALO	6.68E − 04	4.31E − 02	1.26E − 02	1.05E − 02
	ALO-DM	2.61E − 13	1.72E − 11	3.44E − 12	3.81E − 12

Results on Unimodal Test Functions. The results of ALO-DM on unimodal test functions f_1–f_7 are shown in Table 2 for 30 dimensions. The average values in Table 2 depict strong improvements for proposed algorithm except function f_6. Standard deviations depict that the superiority of the proposed algorithm is stable. For functions f_1, f_5, Figs. 1 and 2 are the fitness convergence curves. From these Figs, we can easily

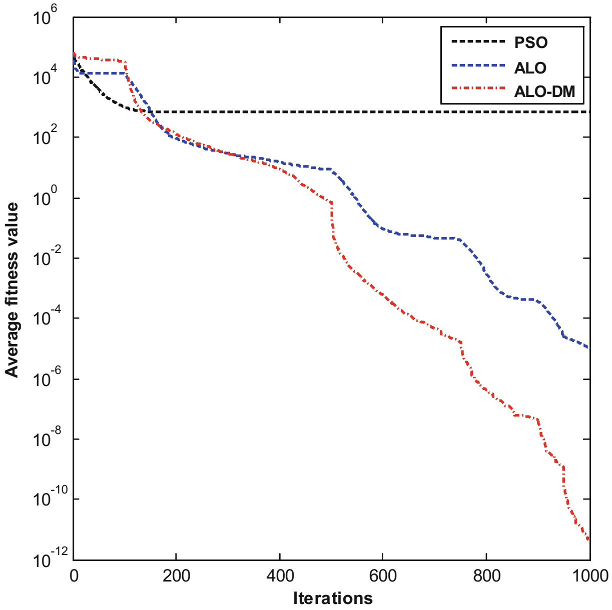


Fig. 1. Convergence curves of fitness value for f_1.

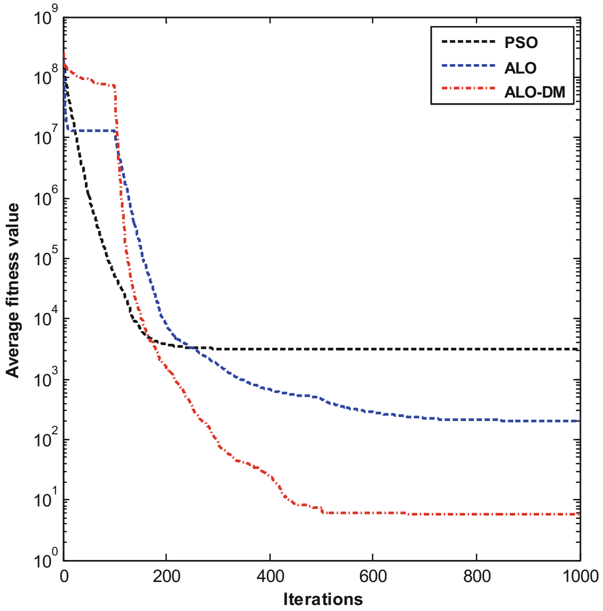


Fig. 2. Convergence curves of fitness value for f_5.

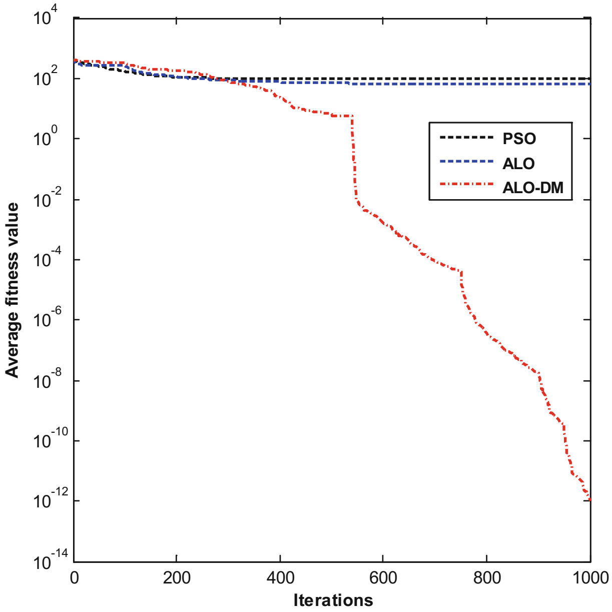


Fig. 3. Convergence curves of fitness value for f_8.

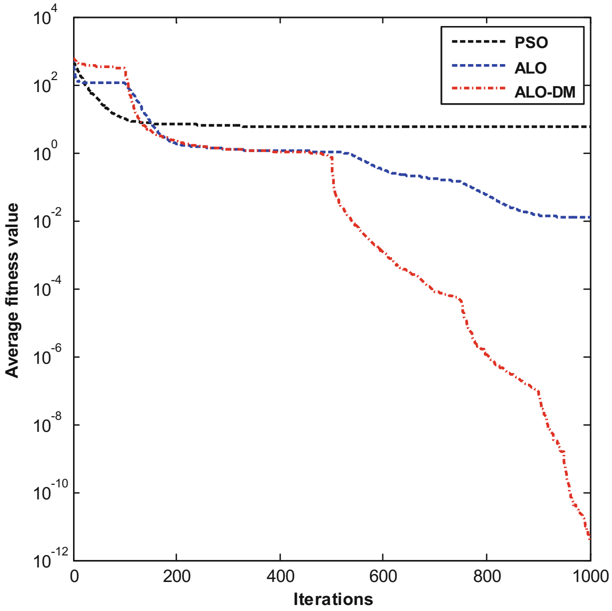


Fig. 4. Convergence curves of fitness value for f_{10}.

find that the values obtained by ALO-DM are closer to the optimal value of benchmark functions. These show that ALO-DM has a better precision than ALO and PSO. It justifies that the proposed algorithm has better performance of exploitation than original ALO.

Results on Multi-modal Test Functions. The results of ALO-DM on multi-modal test functions f_8–f_{10} are presented in Table 2 for 30 dimensions. The average values in Table 2 again depict strong improvements for ALO-DM. Standard deviations show that the superiority of the proposed algorithm is stable. For functions f_8, f_{10}, Figs. 3 and 4 are the fitness convergence curves. From these Figs, we can easily find that ALO-DM converges faster than other algorithms, and the values obtained by ALO-DM are closer to the optimal value of benchmark functions. These show that ALO-DM has a faster convergence speed and a better precision than ALO and PSO. According to the characteristics of multi-modal test functions, it may be show that ALO-DM algorithm has a high level of exploration and satisfaction of local optimal avoidance.

4 Conclusion

Swarm intelligence optimization algorithms have shown significant achievement on big data analytics problems, especially in data mining to solve single objective and multi-objective problems. So it is meaningful to improve the performance of the Swarm intelligence optimization algorithms. In this paper, according to the behavior of antlion's relocation in response to food shortage, differential mutation operator has been incorporated into the ALO to generate the ALO-DM optimization algorithm. In this algorithm, differential mutation operator and greedy strategy enhance the diversity of the population. And the ALO-DM algorithm is validated by 10 benchmark functions. The simulation results demonstrate the performance of ALO-DM is better than, or at least comparable with other algorithms mentioned in this paper.

Acknowledgments. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work is supported in part by the National Natural Science Foundation of China under Grant 61170035, 61272420 and 81674099, Six talent peaks project in Jiangsu Province (Grant No. 2014 WLW-004), the Fundamental Research Funds for the Central Universities (Grant No. 30916011328).

References

1. Kennedy, J., Eberhart, R.C., Yuhui, S., Shi, Y.: Swarm Intelligence. Morgan Kaufmann, San Francisco (2001)
2. Cheng, S., Shi, Y., Qin, Q., Bai, R.: Swarm intelligence in big data analytics. In: Yin, H., Tang, K., Gao, Y., Klawonn, F., Lee, M., Weise, T., Li, B., Yao, X. (eds.) IDEAL 2013. LNCS, vol. 8206, pp. 417–426. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41278-3_51
3. Holland, J.H.: Adaptation in Natural and Artificial System: An Introduction with Application to Biology, Control and Artificial Intelligence. University of Michigan Press, Ann Arbor (1975)

4. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**, 341–359 (1997)
5. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceeding of the IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1948 (1995)
6. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed optimization by ant colonies. In: Proceedings of the First European Conference on Artificial Life, pp. 134–142. Elsevier Publishing, Paris (1991)
7. Yang, X.S., Deb, S.: Cuckoo search via Levy flights. In: Proceedings of the World Congress on Nature and Biologically Inspired Computing, NaBIC 2009, pp. 210–214. IEEE Publication, USA (2009)
8. Wolpert, D.H., Macready, W.G.: No free lunch theorems for search. Technical report SFI-TR-95-02-010. Santa Fe Institute (1995)
9. Mirjalili, S.: The ant lion optimizer. Adv. Eng. Softw. **83**, 80–98 (2015)
10. Yao, P., Wang, H.: Dynamic Adaptive Ant Lion Optimizer applied to route planning for unmanned aerial vehicle. Soft. Comput. **21**(18), 5475–5488 (2016)
11. Zawbaa, H.M., Emary, E., Grosan, C.: Feature selection via chaotic antlion optimization. PLoS ONE **11**(3), e0150652 (2016)
12. Emary, E., Zawbaa, H.M., Hassanien, A.E.: Binary ant lion approaches for feature selection. Neurocomputing **213**, 54–65 (2016)
13. Rajan, A., Jeevan, K., Malakar, T.: Weighted elitism based Ant Lion Optimizer to solve optimum VAr planning problem. Appl. Soft Comput. **55**, 352–370 (2017)
14. Scharf, I., Ovadia, O.: Factors influencing site abandonment and site selection in a sit-and-wait predator: a review of pit-building antlion larvae. J. Insect Behav. **19**, 197–218 (2006)
15. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Trans. Evol. Comput. **3**, 82–102 (1999)
16. Digalakis, J., Margaritis, K.: On benchmarking functions for genetic algorithms. Int. J. Comput. Math. **77**, 481–506 (2001)
17. Molga, M., Smutnicki, C.: Test functions for optimization needs. Test functions for optimization needs (2005)
18. Yang, X.S.: Test problems in optimization. arXiv preprint arXiv:1008.0549 (2010)

A High Precision Recommendation Algorithm Based on Combination Features

Xinhui Hu[1,2], Qizhi Liu[1](✉), Lun Li[1], and Peizhang Liu[2]

[1] State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing, China
lqz@nju.edu.cn
[2] ZTE Corporation, Shenzhen/Nanjing, China

Abstract. Conventional recommendation systems often use binary relations matrices, which could not represent raw data sets efficiently and uniformly. The graph model can represent multiple relationships and form a unified standard of the feature space to recommend the candidate items. Existing graph-based work is generally based on the path to establish the feature space of the data, only concerned about the impact of an item on the description of the user. Utilizing combination features to construct user profiles, this paper concentrates on the contribution maked by combination of items and designs a user-based collaborative filtering algorithm (CFC), and validates the validity of the algorithm in the prototype of the proposed system. The experimental results show that the recommendation algorithms can significantly improve accuracy of the recommendation.

Keywords: Recommendation system · Graph model · Combination features
Collaborative filtering

1 Introduction

The recommendation system can build user profiles based on user behavior habits, such as past click, audition, read, collect, and purchase, as well as infer the user's possible future behavior to help users filter a large amount of irrelevant or low-relevant information. The development of e-commerce, social networking, Internet advertising, search engines, and other areas [19] has promoted the application of recommendation systems and has exposed the problems of traditional recommendation technologies.

With the further popularization of the Internet, users create a large number data while benefiting from information. The growth of data volume is very rapid and highly inline. It is a big challenge to how to make use of numerous highly inline data to perform complex queries. The dataset associated with the recommender system is usually composed of triples like (users, purchase, items) or (users, have, attributes) or (items, have, attributes), which are similar to the representation of RDF, so it is appropriate to describe these semi-structured data in a graph model. The graph model is not forced to rely on the intermediate index structure, has a more flexible data model, can easily improve the recommended system engine when data types and data sources change, and is easier to retrieve association data, suitable for a large number of highly inline data

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 74–86, 2018.
https://doi.org/10.1007/978-3-319-91455-8_7

management. According to existing cases, the graph database is 1000 times more efficient in some applications than a relational database with the same information representation capability, and the code amount is reduced by 10–100 times [38].

The existing recommendation method often uses a binary relationship matrix to represent the data model. However, the binary relationship matrix does not effectively and uniformly express the raw data set because elements in the natural world have multiple relationships. Furthermore, in the graph model, the attributes of node elements can also be integrated into the graph model as nodes, so the graph model greatly enriches the expressiveness of the data set model. For example, users and items can constitute a bipartite model, and users, items, and tags can constitute a three-part graph model. By building a heterogeneous graph model, some different types of nodes and edges are linked to the graph model, that can express multiple relationships.

In fact, the attributes of node elements can also be integrated into the graph model as nodes. The rich expressive ability of the graph model can unite multiple recommendation methods (such as collaborative filtering, content-based filtering) into a complete model, and form a feature space with uniform standards, from which to obtain the required multiple elements for recommendation. For example, there are 3 users {*Alice, John, Tom*} and 3 movies {*AngelsAndDaemons*, *TheDaVinciCode*, *TheShining*}, 7 edge types {*literaryGenre, subject, author, noblework, influencedBy, subsequentWork, broader*}. From the example, semantic path features can be extracted for each user for the recommendation algorithm. In addition, the graph model itself can also provide potential features to enrich the node's feature properties. Therefore, the ability to express the graph model should be fully demonstrated.

Related graph-based research usually uses two types of collections (users, items) or three types of collections (users, items, tags) as models, and the elements of a single collection are independent of each other. This model does not fit well with the complex models of the real world for ignoring many valuable information. Using graph data models can create very flexible data patterns. In the recommendation method based on the graph [12, 17], user profiles are usually constructed only by a single item, which may generate noise and drop the recommendation accuracy. The contributions of this paper:

- We propose combination features based on graph data model.
- We build heterogeneous graphs and feature space using combination features to compute user profiles, and make full use of the advantages of graph data model in terms of strong expressiveness.
- We design the related recommendation algorithm based on multiple relation matrix and the combined features.
- We use HetRec MovieLens dataset in the experiments to evaluate the performance of our method. The results show superior performance over other graph-based recommendation algorithms and benchmark algorithms in terms of precision and recall.

In the rest of this paper, we introduce the research background, main contents and objectives, as well as give the definition and research basis of related problems of the recommendation system in Sect. 2. In Sect. 3, we elaborate the combination features and their formal representations, and point out the problems that may arise from extracting features from a single item. We detail the feature model of the user-item pair based on the combination feature and the feature index structure which can reduce the duplicate calculation. In Sect. 4, we describe the user-based collaborative filtering recommendation algorithm (CFC) which is based on the combined features. We present the result of our experiment and the process of our algorithm implementation in Sect. 5. Finally, we discuss our conclusions.

2 Research Foundation and Problem Definition

For a given user set U and item set I, $|U| = m$, $|I| = n$, forming a user-item pair of $|U \times I| = m * n$, assuming that p ($p < m* n$) user-item pairs have already been scored, then the recommendation system is a system that can predict the score of the remaining $m* n - p$ unknown-rated user-item pairs. The ultimate goal is to give each user a list in which all the items are sorted by the score and to minimize the system losses.

Definition 1. Top-k recommendation: For a given user u, the top-k recommendation refers to selecting the top k items by the system as the recommendation list which have not been scored by u.

If $k = 1$, i.e. top-1 recommendation, the hit rate will be very low. Therefore, $k = 5$, $k = 10$ or $k = 15$ is usually used to evaluate the algorithm. If the test set's sample appears in the top-k recommended list, it is counted as a hit.

In the applications of television programs, movies, music, news, books, labels, etc., there are two main types of recommendation modes, collaborative filtering (CF) and content-based filtering (CBF). CF uses a user's past behavior (collection or evaluation) to build a model to determine which users have similar behaviors or interests with the user, so as to predict the user's possible interest points. CBF uses a series of similar discretized features of an item to recommend other items. These two methods are often mixed, that is, mixed recommendation. However, these two modes are very vague, leaving a lot of space for different algorithms to use [12]. In addition, there are many other recommended methods, such as that based on inherent rules, based on mutual relations, based on graphs, based on global relevance etc. The graph -based recommendation method is a recent research hotspot.

Definition 2. Graph data model: The graph data model of the recommendation system refers to a heterogeneous directed graph $G = (V, E, TV, TE, \varphi V, \varphi E, L, \varphi EW)$, where V is a set of finite nodes, $E \in V \times V$ is a set of finite edges, TV is a set of finite node types, TE is a set of finite edge types, φV: $V \rightarrow TV$ is a mapping function from node to node type, φE: $E \rightarrow TE$ is a mapping function of edge-to-edge type. Each node vi has a corresponding attribute table $li \in L$, li refers to a set of pairs (attribute, value). For each attribute, a li contains one corresponding attribute value pair at most. φE: $E \rightarrow w$ means that each edge can correspond to a weight value w, such as φEW $(ei) = wei$.

The recommendation system can represent various types of data sets (e.g., relation, XML, RDF triples) through heterogeneous graph-based models. A graph data model is built from a data set and integrates multiple different data sets to form a larger, richer data set by linking the same entities [12]. Typical approaches include random walks [15], matrix decomposition [16], and sort-based learning [17].

3 Combination Features

3.1 Feature Space Construction

The graph-based recommendation method is often focused on the path-based graph element. That is, starting from a single item of a user to extract features for the user, so that only the impact of a single item constructs the profile, which can easily cause noise problems in collaborative filtering. For example, in Fig. 1, we predict the score of the dashed line based on the score of the solid line, or recommend *A* for *J*. In that case, only the score data of *B* is helpful for the recommendation prediction (without content recommendation, ignoring the connection between other items and *J*) because only *B* also has a score of *J*. If *A*–*K* and *A*–*L* are used as the basis for the recommendation, since *C* and *D* have score records for *K* and *L*, respectively, *C*–*K* and *D*–*L* are also calculated in this single item model. However, for the desired results (*A*–*J*), it is actually better that these two records are not considered in the calculation. In the worst case, because the score of *C* and *D* are taken into consideration, *M* is recommended to *A*.

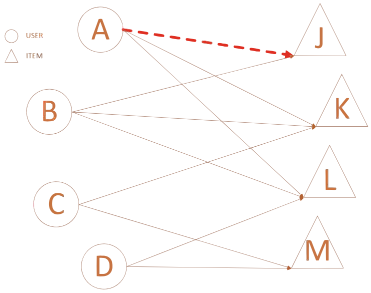


Fig. 1. Noise problems with single item features

This paper proposes a solution to the combination of items and the concept of combined features for the recommendation system. As shown in Fig. 2, *K* and *L* are combined together (coarse line in the figure). That is, only when the user has *K* and *L* at the same time, the two can work together to portray users, thus strengthening the similarity between *A* and *B*, to increase the likelihood of accurately recommending *J* to *A*.

Definition 3. Combination features: Given the user-item pair set *S* of the recommendation system, suppose that the number of related items from the *S* for the user *u* is *n*, and we take the *m* items ($m < n$) as the basis combination features of *u*, namely the combination features: $CombFeature(u) = Combination(f, u, m, n) = \{f(\{item1, item2, \ldots, itemm\}), f(\{item1, item2, \ldots, itemm-1, itemm+1\}), \ldots, f(\{itemn-m+1, itemn-m+2, \ldots, itemn\})\}$, where *f* refers to a map of the *m* items to the final feature.

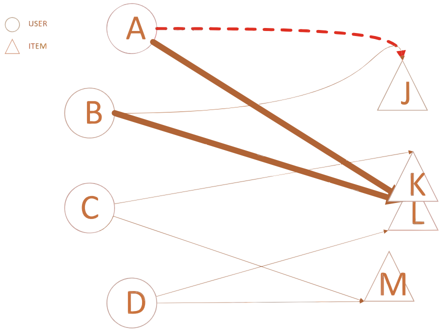


Fig. 2. Solution for combination of items

Different from focusing on the contribution of a single item, the combination features use multiple items to build a profile of the user and build the user's feature space.

Definition 4. Feature space: Given the set S of user-item pairs in a recommendation system, all elements (users or items) of S or features that can be extracted constitutes a feature space. Eg., the recommendation system S contains $user1$-$item1 \Rightarrow \{feature1, feature2\}$, $user2$-$item2 \Rightarrow \{feature2, feature3\}$ and $user3$-$item1 \Rightarrow \{feature3, feature4\}$, the feature space of S is $featureSpace = \{feature1, feature2, feature3, feature4\}$.

The original number of combination feature is the combination number. For ranking learning algorithms, effectively controlling the size of feature space is very important. So we give a mapping function to reduce the size of the feature space.

3.2 User-Item Feature Construction

Combination features generally represents the user's taste and preferences. However, the intermediate process of the algorithm based on the ranking learning is to predict the value of the user-item, so we need to rebuild the feature space for the user-item.

In order to predict the value of the user-item, each user-item pair needs to be given a feature space about the item. If the user's characteristics are (likes romantic comedy, likes American movies) and the characteristics of the items are (romantic comedy, American movies), then the characteristics of such a pair of user-items are (like romantic comedy, like American movies, romantic comedies, American movies), then such user-items get relatively high scores. On the contrary, if the characteristics of the item are (war, Chinese film) and the resulting user-item characteristics are (like to waste a comedy, like American movies, war, Chinese movies), the predicted score is relatively low.

One of the basic assumptions of this paper is that a user has a certain preference for similar types of movies with the movie which is rated by the user. Of course, similar types of movies may resulted in scoring errors for the quality of the movie is not the same. Therefore, we introduce popularity, which refers to how much an item is rated by a user. That is, if an item is connected with more users, the item with the higher degree of welcome. Here we give the definition of the popularity of the item collection.

Definition 5. Popularity of the collection of items: Given collection of items set_{items}, the popularity of set_{items} is the centrality of set_{items}, represent the number of users who are also associated with set_{items}, referred to as $Count(set_{items})$.

Definition 6. User-item combination feature: Given the user-item pair set S of the recommendation system, suppose there are p users and n items for the user u. For the item i, we first take the feature vector a of u. then compute the feature vector b of i. Assume that the user's feature space is $\{feature1, feature2, \ldots, featurep\}$, and the feature space of the item is $\{featurep+1, featurep+2, \ldots, featuren\}$. The user-item feature space is a combination of feature space $\{feature1, feature2, \ldots,$ feature$n\}$. And the feature vector of u-i is $[a,]$. which is a concatenation of a and b. In order to take into account the popularity, the value of each dimension feature is changed according to the corresponding popularity.

Suppose there is a user and that the item sets have three feature modes: $P1$, $P2$, $P3$ after they have been mapped, which constitute the user's feature space ($P1$, $P2$, $P3$). The feature space of the items is (*USA*, *Comedy*, *Donald Petrie*). Then we get the feature table as shown in Table 1. Since the users are the same, the features of the four user-items are all three-dimension (1, 1, 1), and the items features are different.

We also constructed a feature index structure for the dataset to organize the feature space of all users and user-items, speeding up the predictor and avoiding redundancy calculation.

Table 1. Example of feature representation

Users - Items	$P1$	$P2$	$P3$	USA	Comedy	Donald Petrie
User-A	1	1	1	1	1	1
User-B	1	1	1	0	0	0
User-C	1	1	1	1	0	0
User-D	1	1	1	0	1	0

4 CFC Algorithm

In the collaborative filtering algorithm based on the combination feature (CFC), to predict user u's score on item i, supposing u's feature set is $Sf = \{F1, F2, F3, \ldots, Fp\}$, we first traverse the feature set in the training data, and find all users with each feature, and determine for each user with the same feature whether or not i has a score. In this way, while looking for similar users, determine the contribution of each feature to i and finally get a prediction score for u on i. This method does not directly calculate the user's similarity, but indirectly looks for similar users through the features of each dimension, and adds up the predicted values of user-items pairs. Due to a coarse-grained feature model, the features of each dimension have been able to express enough similarities, the approach is feasible.

The input of CFC is a combination feature index T, the user u of the recommendation service, the parameter k of the top-k, the combination feature parameter (n, m). The output is the recommended list of u. Algorithm pseudo code:

Algorithm 1：CFC

```
CFC(Feature Table $T$ , User $u$ , $k$ , $n$ ,$m$ )
Input: user set $U$ , item set $I$ , Users $u$ need to provide recommendation service , combination feature table $T$ , parameter $k$ of top-k, combination feature parameter $n$ , $m$
Output: top-k recommendation list $l$ of ;
1: Initialization: $X$ , $X$ refers to $u$ 's scoring array;
2: Find the corresponding feature of $u$ in $T$ to get the feature array $F$ ;
3: for Feature : $F$ do
4:     for User : $U$ do
5:         if $isFeature(u , f )$ then
6:             for Item : $I$ do
7:                 if $isConnecte(u , i )$ then
8:                     $updat(X )$;
9:                 end if
10:            end for
11:        end if
12:    end for
13: end for
14: Sort $X$ in descending order;
15: $l \leftarrow$ top-k elements of $X$ ;
16: return $l$
```

In Algorithm 1, the way of updating (line number 8) is free to design according to statistical results. This paper uses formula 1 to update the scoring array:

$$X(i) = Xu(i) + countfui/countfu \tag{1}$$

Where *countfui* refers to the number of users associated with an item in a user with feature *f*, *countfu* refers to the number of users.

The effect of parameters (*n*, *m*) in the algorithm on dataset MovieLens is shown in Figs. 3 and 4.

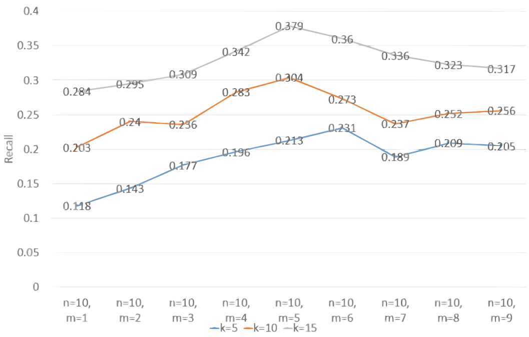


Fig. 3. Recall rate affected by m value

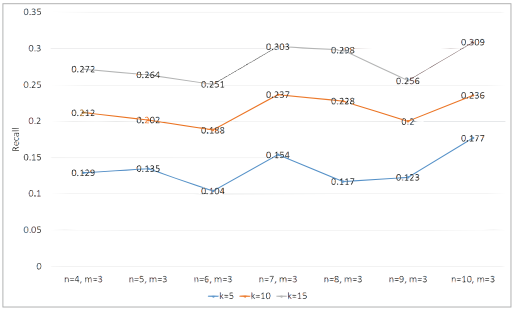


Fig. 4. Recall rate affected by n value

5 Experimental Results Analysis

The recommendation system can be roughly divided into three modules: user profiles modeling, item profiles modeling and recommendation algorithm implementation. This paper mainly focuses on the user profiles modeling and the recommendation implementation modules. In the item modeling, the basic item features are extracted according to their attributes, which is used as an auxiliary process for user-item feature construction.

5.1 Construction of Recommended Systems

In a recommendation system, at first, there should have a user list, an item list and a binary relationship between them, i.e. the original bipartite model, which is the data core of the system. Usually, a corresponding user profiles library and an item profiles library are also needed. After that, a recommendation algorithm framework is built on top of the two profiles libraries. Then the user can get the recommendation service. The user can score the recommended items and the system can feedback and update the data core to re-adjust the user and item library. Figure 5 shows the operational flow of the recommendation system.

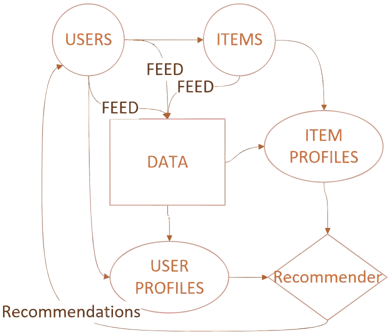


Fig. 5. Recommended system model

5.2 Baseline

- Popularity-based recommendations

Popularity-based recommendation (POP) refers to sorting the list based on the score of the item. It is not a personalized recommendation method. This paper uses a more reasonable Popularity-based recommendation method (POPN) as a baseline. POPN depends on the number of high-scoring items to determine its popularity. Its effect is much better than that of POP.

$$\text{POP}: \ Scoreitem = Coun\{user|user.itemList.contains(item)\} \tag{2}$$

$$\begin{aligned}\text{POPN}: \ & Scoreitem = Coun\{user|user.itemList.contains(item)\\ & \&\& \ (user, item) > ratingt\}\end{aligned} \tag{3}$$

Where *ratingt* is the threshold.

- Non- normalized Neighborhood collaborative filtering

Non-normalized Neighborhood CF (NNCF) is an improvement over traditional k-nearest neighbor collaborative filtering (KNNCF), which usually better than KNNCF in top-k recommendation. The top-k recommendation system aims to sort the users according to their attractiveness, so it is not necessary to normalize the score [8].

- PureSVD

PureSVD (PSVD) is a matrix decomposition method based on collaborative filtering. It reduces the rank of the scoring matrix by single value decomposition [8].

- Personalized PageRank

Personalized PageRank (PRANK) is a random walk method with reset.

- Path Guide

Path Guide (PG) [9] refers to the method of PRANK with semantic path guidance. It is a typical graph-based recommendation method. Through the process of random walk and iterate, PG can get a stable probability value.

5.3 Experiment Settings and Result Analysis

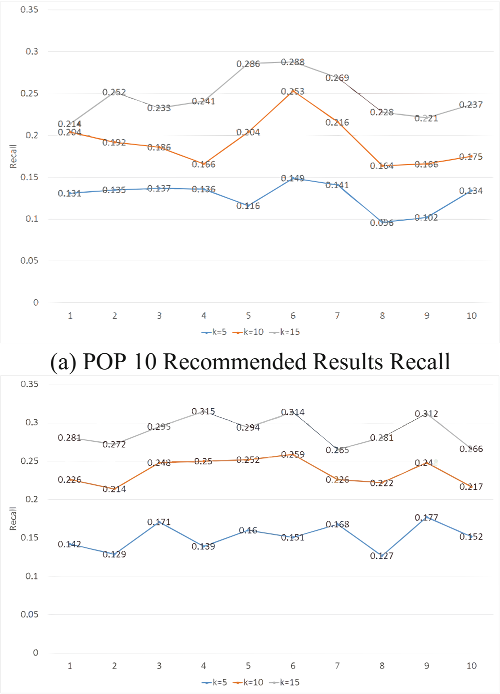
The test methods used in this paper are similar to those in [6, 15]. First, the user-item scoring list is sorted according to the time stamps, and divided into two subsets according to the scoring time. The early 90% is used as the training set. *T* high score records in the remaining 10% used as a test set, the items contained in which are the most relevant items to the user. Then, we randomly select R items that have not been scored for each user and rank them. Finally, evaluate the prediction according to *i*'s position *p* in the *Rank*. If $p <= k$, the prediction is considered as a hit, otherwise, it is considered as a fail. The overall recall rate of the recommended system is calculated according to the following formula:

$$recal(k) = \#hits/T \tag{5.3}$$

Where #*hits* refers to the number of hits. The overall recall rate is the proportion of hits in the *T*.

This paper uses the movie data set HetRec MovieLens as experimental data set. The entire dataset contains a total of 2217 users and 10197 items (movies). Each item has 5 attributes: director, category, actor, country, and label. The total data set contains 855,598 scoring records, of which the first 770031 records, sorted by time stamp, are used as training data, and 85567 records is taken out as a test case. The records are aggregated according to the user in training, i.e., one user corresponds to one query. During the test, we randomly select one record and 1000 items that were not scored by the user in the record to form a test query and count the hits.

This paper constructe the user's preferences from favorite items, not consider the items that the user does not like. For each user, a sorting list based on popularity recommended. The experimental results show that the POPN is better than the POP. Compared with POP, the average recall rate in POPN increased by 0.025(k = 5), 0.043(k = 10), 0.043(k = 15), respectively (Fig. 6).



(a) POP 10 Recommended Results Recall

(b) POPN 10 Recommended Results Recall

Fig. 6. Experimental results based on popularity

Table 2. Recommended precision

Recall(k)	CFC	POP	POPN	NNCF	PSVD	PRANK	PG
K = 5	0.213	0.127	0.152	0.181	0.182	0.152	0.213
K = 10	0.304	0.193	0.236	0.262	0.270	0.238	0.305
K = 15	0.379	0.247	0.290	0.322	0.339	0.300	0.371
Precision(k)	CFC	POP	POPN	NNCF	PSVD	PRANK	PG
K = 5	0.282	0.119	0.151	0.179	0.180	0.150	0.281
K = 10	0.335	0.185	0.224	0.254	0.262	0.237	0.335
K = 15	0.397	0.239	0.288	0.320	0.331	0.308	0.392

In order to compare the proposed method with those baseline methods in Sect. 5.2 intuitively, Table 2 shows the best results of those methods in the optimal parameters.

From Table 2, we can see that the precision of CFC (n = 10, m = 5) algorithm is far superior to all benchmark methods except PG and close to PG.

In addition, CFC does not use any other information on users or items except ID. That shows CF method based on the combination features is effective.

The running time of the CFC recommendation algorithm increases with the number of combination features and the number of users corresponding to each feature. When taking 9 combinations of 10 items, the average number of users corresponding to the average item combination is the smallest, which is also consistent with statistics law. While taking 1 combination of 10 items, the number of users corresponding to the average item combination reaches 214.2, so the running time is increasing.

In the case of $k = 5$, we can see that the CFC ($n = 10$, $m = 9$) algorithm is much better than all other baseline methods. The performance is also better than other baseline methods in the case of $k = 10$ and $k = 15$. The CFC algorithm of this paper calculates the similarity between users based on the combination of features, and predicts the user's relevance to the items, which is used to rank and produce recommendation results. Due to the relatively large feature space, the time consumed by the sorting process is proportional to the number of user's features and the average number of users with the same features. Fortunately, when $n = 10$ and $m = 9$, the result is the best (Fig. 7).

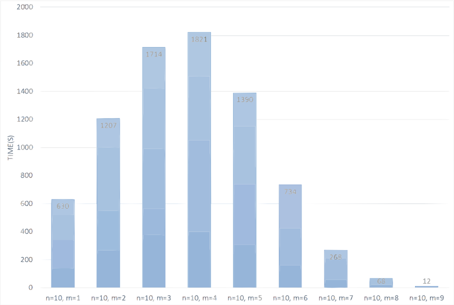


Fig. 7. CFC runtime

6 Discussion and Conclusion

Based on the graph model, this paper proposes the combination features to construct user profiles. Based on the user's features, the user-based Collaborative Filtering Recommendation Algorithm (CFC) is designed. In general, collaborative filtering based on items requires the condition that one item corresponds to multiple users, while the combination features model just requires multiple items correspond to one user, regardless of the individual item of the user. The CFC first looks for similar users based on the coarse-grained combination of features and scores the relevance of the items based on a similar user list to form a recommendation list. Finally, this paper uses the off-line evaluation method and conducts experiments in the recommendation system. The experimental results show that the effectiveness of the recommendation based on the combination of features. In the MovieLens data set, the best case is better than all comparison methods.

In this paper, the commonly used top-k recommended performance evaluation is used to determine the accuracy. Further work will be based on more evaluation indicators to conduct experiments. The experiments will be conducted on more types of data sets to study sorting based learning. Due to the reasons of data sets, the degree of heterogeneity of the graph data model in this paper is limited, and it is necessary to further study the recommendation method based on the high degree of heterogeneity graph model.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (Grant No. 61572247).

References

1. Nguyen, H., Dinh, T.: A modified regularized non-negative matrix factorization for movielens. In: 2012 IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), pp. 1–5. IEEE (2012)
2. Li, Q., Kim, B.: Constructing user profiles for collaborative recommender system. In: Advanced Web Technologies and Applications, pp. 100–110 (2004)
3. Huang, Z., Chung, W., Ong, T.H., et al.: A graph-based recommender system for digital library. In: Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 65–73. ACM (2002)
4. Lao, N., Cohen, W.W.: Relational retrieval using a combination of path-constrained random walks. Mach. Learn. **81**(1), 53–67 (2010)
5. Takács, G., Pilászy, I., Németh, B., et al.: Scalable collaborative filtering approaches for large recommender systems. J. Mach. Learn. Res. **10**, 623–656 (2009)
6. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the Fourth ACM Conference on Recommender Systems, pp. 39–46. ACM (2010)
7. Neo Technology. Powering Recommendations with a Graph Database. https://neo4j.com/

8. Vicknair, C., Macias, M., Zhao, Z., et al.: A comparison of a graph database and a relational database: a data provenance perspective. In: Proceedings of the 48th Annual Southeast Regional Conference, p. 42. ACM (2010)
9. Linden, G.D., Jacobi, J.A., Benson, E.A.: Collaborative recommendations using item-to-item similarity mappings: U.S. Patent 6,266,649[P], 24 July 2001
10. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
11. Wikipedia. https://en.wikipedia.org/wiki/Recommender_system
12. Beel, J., Gipp, B., Langer, S., et al.: Research-paper recommender systems: a literature survey. Int. J. Digit. Libr. **17**(4), 305–338 (2016)
13. Schafer, J., Frankowski, D., Herlocker, J., et al.: Collaborative Filtering Recommender Systems. In: The Adaptive Web, pp. 291–324 (2007)
14. Wang, J., De Vries, A.P., Reinders, M.J.T.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 501–508. ACM (2006)
15. Lee, S., Park, S., Kahng, M., et al.: Pathrank: a novel node ranking measure on a heterogeneous graph for recommender systems. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 1637–1641. ACM (2012)
16. Yu, X., Ren, X., Sun, Y., et al.: Personalized entity recommendation: a heterogeneous information network approach. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, pp. 283–292. ACM (2014)
17. Islam, M.S., Liu, C., Li, J.: Efficient answering of why-not questions in similar graph matching. IEEE Trans. Knowl. Data Eng. **27**(10), 2672–2686 (2015)
18. Ma, S., Li, J., Hu, C., et al.: Big graph search: challenges and techniques. Front. Comput. Sci. **10**(3), 387–398 (2016)
19. Noia, T.D., Ostuni, V.C., Tomeo, P., et al.: Sprank: semantic path-based ranking for top-n recommendations using linked open data. ACM Trans. Intell. Syst. Technol. (TIST) **8**(1), 9 (2016)

The 3rd Workshop on Big Data Quality Management (BDQM 2018)

Secure Computation of Pearson Correlation Coefficients for High-Quality Data Analytics

Sun-Kyong Hong, Myeong-Seon Gil, and Yang-Sae Moon(✉)

Department of Computer Science, Kangwon National University, Chuncheon, Korea
{hongssam,gils,ysmoon}@kangwon.ac.kr

Abstract. In this paper, we present a secure method of computing Pearson correction coefficients while preserving data privacy as well as data quality in the distributed computing environment. In general data analytical/mining processes, individual data owners need to provide their original data to the third parties. In many cases, however, the original data contain sensitive information, and the data owners do not want to disclose their data in the original form for the purpose of privacy preservation. In this paper, we address a problem of secure multiparty computation of Pearson correlation coefficients. For the secure Pearson correlation computation, we first propose an advanced solution by exploiting the secure scalar product. We then present an approximate solution by adopting the lower-dimensional transformation. We finally empirically show that the proposed solutions are practical methods in terms of execution time and data quality.

Keywords: Secure multiparty computation
Privacy preserving data mining · Pearson correlation coefficients
High-quality data analytics

1 Introduction

In recent era of cloud computing and social networks, a huge amount of data are stored and maintained in the distributed computing environment, and thus, the risk of information disclosure has also increased. Accordingly, the importance of data privacy is increasing day by day. In this paper, we focus on preserving correlation of data as well as their privacy in the distributed computing environment. Correlation is an important measure of representing the degree of how two data are related, and it is widely used in regression analysis or predictive data mining. The following examples show how the correlation can be used in real applications such as finance, insurance, and medical services and why the data privacy is important in such applications [4,11].

- A research institute wants to know which correlation is in between customers' age and their financial trade, but it cannot disclose customers' private information and their financial transactions due to privacy concerns.

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 89–98, 2018.
https://doi.org/10.1007/978-3-319-91455-8_8

- An educational institute wants to know which correlation is in between IQs and salaries, but companies cannot disclose their employees' personal information due to privacy concerns.

In this paper, we discuss the problem of *secure correlation computation*, which calculates correlation securely, that is, calculates high-quality correlation coefficients while preserving privacy of data provided by different data owners in the distributed computing environment. Figure 1 shows the conceptual procedure of secure correlation computation. As shown in the figure, for the secure correlation computation, we need to calculate the correlation between two data while both data owners do not disclose their original sensitive data, i.e. while they preserve privacy of their own data. In particular, we here focus on a secure solution of computing Pearson correlation coefficients [3,13]. The Pearson correlation coefficient is usually used for interval scale or ratio scale data such as standard scores and time-series data. In this paper, we call the secure computation of Pearson correlation coefficient *SPCC*(*secure Pearson correlation computation*) and present novel solutions of SPCC by exploiting the secure scalar product [5,8] and the lower-dimensional transformation [9,10].

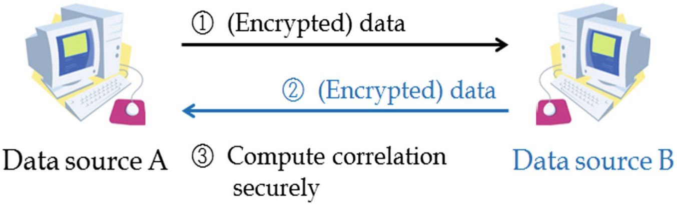


Fig. 1. Conceptual procedure of secure correlation computation.

More specifically, in this paper we present two solutions for SPCC. We first present a naive solution which applies the secure scalar product directly to SPCC. The naive solution provides high-quality (i.e. correct) Pearson correlation coefficients, but it incurs a privacy disclosure problem by exposing sensitive standard deviations of original data. To solve this problem, we present an advanced solution which uses *normalized* data instead of original data. We call this advanced solution $SPCC_{Adv}$. Through the normalization process, SPCC_{Adv} resolves the problem of disclosing raw standard deviations and at the same time obtains the same Pearson correlation coefficient, which will provide high-quality data analytical/mining results. However, SPCC_{Adv} requires that all or even more of the original data be transferred between the two data providers for secure computation, which incurs a severe computation and communication overhead. To alleviate this overhead, we next present an approximate solution that uses lower-dimensional transformed data. As the lower-dimensional transformation, we use PAA (piecewise aggregate approximation) and call this approximate solution $SPCC_{PAA}$, which applies the secure scalar product to PAA-transformed low-dimensional data rather than high-dimensional original data. Using low-dimensional data SPCC$_{PAA}$ achieves significant reduction of computation and communication overhead with only a little quality (i.e., accuracy) degradation of Pearson correlation coefficients.

The rest of this paper is organized as follows. Section 2 explains related work and defines the problem of secure correlation computation. Section 3 presents two novel solutions of SPCC and discusses their correctness and secureness. Section 4 shows the experimental results. Finally, Sect. 5 concludes the paper.

2 Related Work

Secure multi-party computation (SMC) [4,6,14] has been frequently used in privacy preserving data mining [1,2,11]. After Yao [15] has first proposed an SMC method for the comparison operation, many SMC solutions for summations, scalar products, and distance computations have been proposed [4–6]. In general, SMC has the advantage of obtaining accurate, high-quality data analytical/mining results while preserving data privacy, but has the disadvantage of incurring additional computation and communication overhead due to complex encryption and intermediate computation.

In this paper, we use the secure scalar product to calculate the correlation between data while preserving data privacy. A secure scalar product means to accurately calculate the scalar product $X \cdot Y$ without revealing the original values of both data X and Y. Representative methods are random matrix and homomorphic encryption methods [6,7,16]. In this paper, we adopt the simple random matrix method. This method is a matrix multiplication-based technique in which data is encrypted by random matrix and random values shared by both data providers. Precisely speaking, instead of sending the original data X or Y of length k, data providers transmit the encrypted data X' and Y', where $x'_i = x_i + a_{i,1} \cdot R_1 + a_{i,2} \cdot R_2 + \cdots + a_{i,k/2} \cdot R_{k/2}$ and $y'_i = a_{1,i} \cdot y_1 + a_{2,i} \cdot y_2 + \cdots + a_{k,i} \cdot y_k$ encrypted by a random matrix $\mathbf{A} = [a_{i,j}]$ and random values $R_1, \ldots, R_{k/2}$, to preserve the privacy of their original data. In particular, this random matrix-based secure scalar product has an advantage of no necessary of complex decoding operations. For a detailed description and accuracy of the random matrix method, refer to [6,8].

3 Problem Definition

We now define the problem of secure correlation computation as follows.

Problem Definition: For the data X and Y from data providers Alice and Bob, the *secure correlation computation* is the problem of computing the correlation coefficient of X and Y in the 'secure (or privacy preserved) state'. Here the 'secure state' means that Alice and Bob do not disclose their original data to each other. □

Using secure correlation computation we can preserve the privacy of original sensitive data, and at the same time we can get correct correlation coefficients for high-quality data analytical/mining services. In addition, given two data $X\,(=\{x_1,\ldots,x_k\})$ and $Y\,(=\{y_1,\ldots,y_k\})$, the Pearson correlation coefficient $Pearson(X,Y)$ is computed as the following Eq. (1).

$$Pearson(X,Y)=\frac{Cov(X,Y)}{\sigma(X)\sigma(Y)}. \tag{1}$$

In Eq. (1), $Cov(X,Y)$ is the covariance of X and Y, i.e. $Cov(X,Y)$ is computed as $\frac{\sum_{i=1}^{k}(x_i-\mu(X))(y_i-\mu(Y))}{k}$; $\sigma(X)$ and $\mu(X)$ are the standard deviation and mean of X, respectively. Here, $-1\le Pearson(X,Y)\le 1$ holds, and we can know the direction and degree of correlation from the sign and absolute value of the coefficient.

4 Secure Multiparty Computation of Pearson Correlation Coefficients

4.1 Advanced Solution Using Normalized Data

In this section, we propose SPCC$_{Adv}$, which exploits normalized data for secure computing of Pearson correlation coefficients. The reason for using the normalized data is that the *basic* solution, which simply applies the secure scalar product to SPCC, has a problem of disclosing raw standard deviation, which is sensitive information. The basic solution uses $Pearson(X,Y)$ of Eq. (1) as it is, and the covariance $Cov(X,Y)$ is calculated from $\sum_{i=1}^{k}(x_i-\mu(X))(y_i-\mu(Y))$. Note that here we can get $\sum_{i=1}^{k}(x_i-\mu(X))(y_i-\mu(Y))$ securely using the secure scalar product, and thus, the computation of $Cov(X,Y)$ is secure. However, for the calculation of $Pearson(X,Y)$, the standard deviation $\sigma(X)$ and $\sigma(Y)$ must be disclosed to the other party. The standard deviation represents the degree to which data are scattered, and it is generally very important information that reflects the characteristics of actual data. In particular, if it is published with the average, the original data can be inferred. Thus, we conclude that the basic solution, which can get the Pearson correlation coefficient correctly, does not preserve privacy even if it exploits the secure scalar product.

To solve the problem of disclosing the standard deviation, SPCC$_{Adv}$ uses the normalized data instead of the original raw data. We explain how SPCC$_{Adv}$ works in detail. First, for data X, we can get its normalized data $\bar{X}$, where each $\bar{x}_i=\frac{x_i-\mu(X)}{\sigma(X)}$. Then, the mean and standard deviation of $\bar{X}$ become 0 and 1, respectively. Thus, if using the normalized data $\bar{X}$ and $\bar{Y}$ in Eq. (1), we get $Pearson(\bar{X},\bar{Y})$ as the following Eq. (2).

$$Pearson(\bar{X},\bar{Y})=\frac{Cov(\bar{X},\bar{Y})}{\sigma(\bar{X})\sigma(\bar{Y})}=\frac{\sum_{i=1}^{k}\bar{x}_i\cdot\bar{y}_i}{k}. \tag{2}$$

We here note that the distribution of normalized data is the same as that of the original data, and thus, the correlation does not change even if the data are normalized. That is, $Pearson(X, Y) = Pearson(\bar{X}, \bar{Y})$ holds. Therefore, SPCC$_{Adv}$ can obtain $Pearson(X, Y)$ securely by calculating $\sum_{i=1}^{k} \bar{x}_i \cdot \bar{y}_i \,(= \bar{X} \cdot \bar{Y})$ by the secure scalar product, and accordingly, it can provide high-quality correlation coefficients to data analytical/mining services.

Algorithm 1 shows SPCC$_{Adv}$, which applies the secure scalar product to the normalized data. It consists of three large steps: (1) data normalization, (2) data encryption, and (3) secure computation. To use the random matrix-based secure scalar product, we first assume that data providers, Alice and Bob, share a $k \times \frac{k}{2}$ random matrix $\mathbf{A}$. In the data normalization step, Alice and Bob normalize their own data using the mean and standard deviation of the original, respectively. After then, they use the normalized data instead of the original data in computing the Pearson correlation coefficient. In the data encryption step, Alice and Bob encrypt the normalized data using the random matrix-based method and send the encrypted data to each other. That is, Alice encrypts her normalized data $\bar{X}$ into Z and sends the encrypted Z to Bob (Lines 1 to 3); Bob encrypts his normalized data $\bar{Y}$ into V and sends the encrypted V with a scalar value s to Alice (Lines 4 to 6). Finally, in the secure computation step, Alice first calculates the scalar product and then obtains the Pearson correlation coefficient (Lines 7 to 9). For the detailed procedure of the random matrix-based secure scalar product, readers are referred to [6,8].

```
**Algorithm 1 .** SPCC$_{Adv}$    // SPCC using normalized data
(1) $X$ and $Y$ are $k$-length data owned by Alice and Bob, respectively.
(2) Assume that Alice and Bob maintain the same $k \times \frac{k}{2}$ matrix $\mathbf{A} = [a_{i,j}]$.
**Alice & Bob:** Normalize $X$ to $\bar{X}$ using $\mu(X)$ and $\sigma(X)$   (or $Y$ to $\bar{Y}$ using $\mu(Y)$ and $\sigma(Y)$).
**Alice:** Encrypt $\bar{X}$.
    1. Make a $\frac{k}{2}$-length data $R$ with random numbers $r_1, r_2, \ldots, r_{\frac{k}{2}}$;
    2. Compute a $k$-length data $Z$, where $z_i = \bar{x}_i + a_{i,1} \cdot r_1 + \cdots + a_{i,\frac{k}{2}} \cdot r_{\frac{k}{2}}$;
    3. Send $Z$ to Bob;
**Bob:** Encrypt $\bar{Y}$.
    4. Compute a scalar value $s = Z \cdot \bar{Y}$;
    5. Compute a $\frac{k}{2}$-length data $V$, where $v_i = a_{1,i} \cdot \bar{y}_1 + \cdots + a_{k,i} \cdot \bar{y}_k$;
    6. Send $V$ with $s$ to Alice;
**Alice:** Compute $Pearson(\bar{X}, \bar{Y})$ in a secure way.
    7. Compute a scalar value $s' = V \cdot R$;
    8. Obtain $\bar{X} \cdot \bar{Y} = s - s'$;
      // Note that $s - s'$ is the scalar product of $\bar{X}$ and $\bar{Y}$.
    9. Compute $Pearson(\bar{X}, \bar{Y}) = \frac{X \cdot Y}{k}$;
      // Note that $Pearson(X, Y) = Pearson(\bar{X}, \bar{Y})$.
```

We now formally prove correctness and secureness of SPCC$_{Adv}$.

Theorem 1. *SPCC$_{Adv}$ performs the secure correlation computation correctly.*

Proof: For this proof, we need to show both correctness and secureness: the former means that the computed Pearson correlation coefficient is correct, and the latter means that the privacy of both original data X and Y is preserved. First, SPCC$_{Adv}$ calculates $Pearson(\bar{X}, \bar{Y})$ from the normalized data $\bar{X}$ and $\bar{Y}$. As we explained earlier, $Pearson(X, Y)$ is identical to $Pearson(\bar{X}, \bar{Y})$, and thus, SPCC$_{Adv}$ computes the Pearson correlation coefficient correctly. This correctness means that the quality of Pearson correlation coefficients is guaranteed. Second, SPCC$_{Adv}$ exploits the secure scalar product, and thus, its secureness is guaranteed by the secure scalar product [6]. Overall, SPCC$_{Adv}$ computes $Pearson(X, Y)$ correctly and securely. □

4.2 Approximate Solution Using Low-Dimensional Data

SPCC$_{Adv}$ proposed in Sect. 4.1 compute the coefficient securely and correctly, but it instead incurs a severe computation and communication overhead. That is, SPCC$_{Adv}$ encrypts long high-dimensional data $\bar{X}$ and $\bar{Y}$ and transmits the long encrypted data Z and V to each other. These encryption and transmission processes incur the severe computation and communication overhead. To alleviate this overhead, in this section we adopt a lower-dimensional transformation [11], which converts high-dimensional data into low-dimensional data with preserving the characteristics of original data as much as possible. In particular, we use PAA since it is simple and efficient and present SPCC$_{PAA}$ as an approximate solution of SPCC.

Algorithm 2 shows SPCC$_{PAA}$, the PAA-based approximate solution of SPCC. Like SPCC$_{Adv}$, it consists of three large steps: (1) data normalization & lower-dimensional transformation, (2) data encryption, and (3) secure computation. As shown in the algorithm, SPCC$_{PAA}$ obtains low-dimensional data $\bar{X}_l$ and $\bar{Y}_l$ first and then applies those low-dimensional data to SPCC$_{Adv}$. Using low-dimensional data $\bar{X}_l$ and $\bar{Y}_l$ rather than high-dimensional data $\bar{X}$ and $\bar{Y}$, SPCC$_{PAA}$ significantly reduces the computation and communication overhead. However, it may cause low accuracy of the Pearson correlation coefficient, and this low accuracy may degrade the quality of analytical/mining results.

Figure 2 summarizes the advantages and disadvantages of SPCC$_{Adv}$ and SPCC$_{PAA}$ proposed in this paper. As shown in the figure, SPCC$_{Adv}$ focuses on accuracy with a little sacrifice in efficiency while SPCC$_{PAA}$ focuses on efficiency with a little sacrifice in accuracy. Figure 2 shows this trade-off relationship between SPCC$_{Adv}$ and SPCC$_{PAA}$. We empirically show the accuracy and efficiency of SPCC$_{Adv}$ and SPCC$_{PAA}$ in Sect. 4.

5 Experimental Evaluation

In this section, we show the experimental result of SPCC. For this, we empirically compare SPCC$_{Adv}$ and SPCC$_{PAA}$ with the *insecure* Pearson correlation

Algorithm 2 . SPCC$_{PAA}$ // approximate SPCC using low-dimensional data

```
Alice & Bob: Extract low-dimensional data $\bar{X}_l$ and $\bar{Y}_l$ from the normalized data $\bar{X}$ and $\bar{Y}$ by PAA.
Alice: Encrypt her low-dimensional data $\bar{X}_l$.
    1. Execute Lines 1 to 3 of SPCC$_{Adv}$ using $\bar{X}_l$ instead of $\bar{X}$;
Bob: Encrypt his low-dimensional data $\bar{Y}_l$.
    2. Execute Lines 4 to 6 of SPCC$_{Adv}$ using $\bar{Y}_l$ instead of $\bar{Y}$;
Alice: Compute an approximate Pearson correlation coefficient in a secure way.
    3. Execute Lines 7 to 9 of SPCC$_{Adv}$;
```

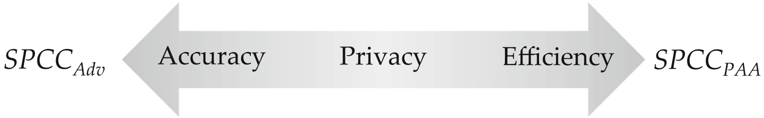


Fig. 2. Trade-off relationship between SPCC$_{Adv}$ and SPCC$_{PAA}$.

computation, *IPCC* in short, which does not use the secure computation. In the experiment, we use temperature time-series data [12], which consist of 900,000 entries that represent the daily mean temperatures for many parts of the world from 1900 to 2012. From these time-series data, we extract 256-length time-series and use those data in the experiment. We repeat each experiment 50 times and use the average as a result. The hardware platform is a PC equipped with Intel Xeon Quad Core 3.1 GHz CPU and 4 GB RAM, and its software platform is Cent OS 5.9 operating system.

Figure 3 compares execution times of three methods by varying the number of data. For Alice's each data, we measure the execution time by changing Bob's number of data from 10 to 10000 by 10 times. We then express the results as relative values for IPCC, that is, $\frac{SPCC_{Adv}}{IPCC}$ and $\frac{SPCC_{PAA}}{IPCC}$. As shown in the figure, SPCC$_{Adv}$ has a relatively large execution time compared to IPCC. We can interpret this degradation as the inevitable overhead that comes from securely calculating the correlation coefficient. For SPCC$_{PAA}$, we convert 256-length data to 128- and 64-length data and use those low-dimensional data in the experiment. We call these two cases SPCC$_{PAA}$(128) and SPCC$_{PAA}$(64), respectively. As shown in the figure, the execution time of SPCC$_{PAA}$ is similar to that of IPCC (for example, SPCC$_{PAA}$(128)) or even shorter than that of IPCC (for example SPCC$_{PAA}$(64)). This is because we can reduce the overall computation and communication overhead by using low-dimensional data. This result means that we can achieve high performance using the lower-dimensional transformation with a slight loss of accuracy.

SPCC$_{PAA}$ uses low-dimensional transformed data to reduce the overhead, but it may cause error in correlation coefficient. Table 1 shows the difference of the correlation coefficient with the original while varying the degree of low dimensional transformation for 512- and 256-length data. As shown in the table, the lower-dimensional transformation causes a small error of 0.27% to 1.77%.

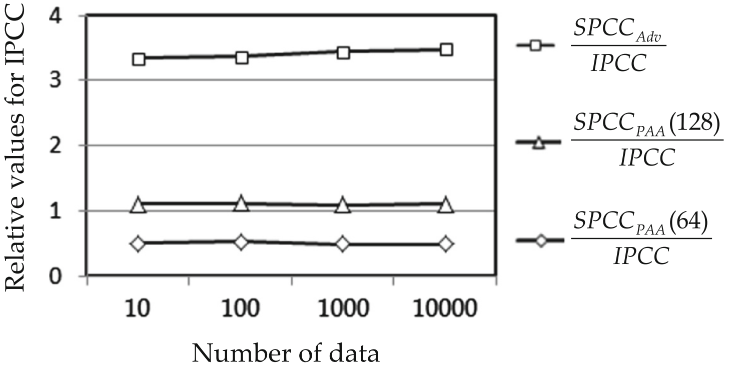


Fig. 3. Comparison of execution times by varying the number of data.

As a matter of course, the larger the low-dimensional conversion ratio, the greater the error. However, all of these errors are relatively small values of less than 2%. This result means that the proposed SPCC$_{PAA}$ can be practically useful in a real environment when we consider the trade-off between large performance improvement and small accuracy degradation.

Table 1. Correlation difference with the original for using different low dimensions.

Original (high) dimensions	Low dimensions of SPCC$_{PAA}$				
	256	128	64	32	16
512	0.27%	0.60%	0.90%	1.20%	–
256	–	0.39%	0.88%	1.27%	1.77%

6 Conclusions

In this paper, we proposed a novel solution of computing the Pearson correlation coefficient securely in the distributed computing environment. For this, we first presented an advanced solution, called SPCC$_{Adv}$, which exploited the random matrix-based secure scalar product and proved its correctness and secureness as a theorem. We next proposed an approximate solution, called SPCC$_{PAA}$, which exploited the lower-dimensional transformation to reduce the computation and communication overhead of SPCC$_{Adv}$. We finally empirically showed the efficiency and accuracy of the proposed methods and concluded that the methods were practically useful in real world distributed applications. Recently, privacy protection of individuals and groups has become increasingly important due to leakage of personal information. Therefore, we believe that this study will be a very useful research that can be applied to various fields of privacy-preserving or secure applications.

Acknowledgment. This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2016-0-00179, Development of an Intelligent Sampling and Filtering Techniques for Purifying Data Streams).

References

1. Aggarwal, C.C., Yu, P.S.: Privacy-preserving data mining: a survey. In: Gertz, M., Jajodia, S. (eds.) Handbook of Database Security, pp. 431–460. Springer, Boston (2008). https://doi.org/10.1007/978-0-387-48533-1_18
2. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: Proceedings of International Conference on Management of Data, ACM SIGMOD, Dallas, TX, pp. 439–450, June 2000
3. Blaikie, N.: Analyzing Quantitative Data. Sage Publications, London (2003)
4. Du, W., Atallah, M.J.: Secure multi-party computation problems and their applications - a review and open problems. In: Proceedings of the 2001 Workshop on New Security Paradigms, New York, NY, pp. 13–22, September 2001
5. Goethals, B., Laur, S., Lipmaa, H., Mielikäinen, T.: On private scalar product computation for privacy-preserving data mining. In: Proceedings of the 7th International Conference on Information Security and Cryptology, Seoul, Korea, pp. 104–120, December 2004
6. Jiang, W., Murugesan, M., Clifton, C., Si, L.: Similar document detection with limited information disclosure. In: Proceedings of the 24th International Conference on Data Engineering, Cancun, pp. 735–743, April 2008
7. Kaosar, M.G., Paulet, R., Yi, X.: Fully homomorphic encryption based two-party association rule mining. Data Knowl. Eng. **76–78**, 1–15 (2012)
8. Kim, S.-P., Gil, M.-S., Kim, H., Choi, M.-J., Moon, Y.-S., Won, H.-S.: Efficient two-step protocol and its discriminative feature selections in secure similar document detection. Secur. Commun. Netw. **2017**, Article ID 6841216, 1–12 (2017)
9. Lee, M., Lee, S., Choi, M.-J., Moon, Y.-S., Lim, H.-S.: HybridFTW: hybrid computation of dynamic time warping distances. IEEE Access **6**, 2085–2096 (2018)
10. Lee, S., Kim, B.-S., Choi, M.-J., Moon, Y.-S.: Coefficient control multi-step k-NN search in time-series databases. Int. J. Innov. Comput. Inf. Control **12**(2), 419–431 (2016)
11. Moon, Y.-S., Kim, H.-S., Kim, S.-P., Bertino, E.: Publishing time-series data under preservation of privacy and distance orders. In: Proceedings of the 21st International Conference on Database and Expert Systems Application, Bilbao, Spain, pp. 17–31, August 2010
12. National Climate Data Center. http://www.ncdc.noaa.gov
13. Sayal, M., Singh, L.: Privately detecting pairwise correlations in distributed time series. In: Proceedings of IEEE International Conference on Privacy, Security, Risk, and Trust and IEEE International Conference on Social Computing, Boston, MA, pp. 981–987, October 2011
14. Won, H.-S., Kim, S.-P., Lee, S., Choi, M.-J., Moon, Y.-S.: Secure principal component analysis in multiple distributed nodes. Secur. Commun. Netw. **9**(14), 2348–2358 (2016)

15. Yao, A.C.: Protocols for secure computations. In: Proceedings of the 23th IEEE Symposium on Foundations of Computer Science, Chicago, IL, pp. 160–164, November 1982
16. Yi, X., Kaosar, M.G., Paulet, R., Bertino, E.: Single-database private information retrieval from fully homomorphic encryption. IEEE Trans. Knowl. Data Eng. **25**(5), 1125–1134 (2013)

Enabling Temporal Reasoning for Fact Statements: A Web-Based Approach

Boyi Hou[1,2](✉) and Youcef Nafa[1,2]

[1] School of Computer Science, Northwestern Polytechnical University,
127 West Youyi Road, Xi'an 710072, Shaanxi, People's Republic of China
{ntoskrnl,youcef.nafa}@mail.nwpu.edu.cn

[2] Key Laboratory of Big Data Storage and Management,
Northwestern Polytechnical University,
Ministry of Industry and Information Technology,
127 West Youyi Road, Xi'an 710072, Shaanxi, People's Republic of China

Abstract. There exists a precise time period during which a given fact such as an event or a status is valid. In this paper, we propose a new approach to determine the validity time of a fact statement by leveraging unstructured and noisy data from the Web, while overcoming the limitations of existing natural language processing technologies designed for the same task. Given a fact and its temporal relevance text, the proposed solution first constructs a Semantic Bayesian Network, then estimates the validity probabilities of time points using the constructed network. In the interest of dealing with the semantic complexity of keywords, we also present a technique based on relative standard deviation to estimate distortion risks of keywords and incorporate their risk estimation into the process of probability computation. Our experiments on real data shows that the proposed approach can achieve considerable improvements in performance over 2 state-of-the-art alternatives, and the proposed risk reduction technique can effectively improve validity time reasoning's precision.

Keywords: Temporal logic · Semantic clique · Semantic distortion risk

1 Introduction

A fact such as an event or a status, usually expressed by a statement, always corresponds to a well defined time point or time interval when it is considered valid. This temporal attribute of facts plays an important role in the Q&A and knowledge discovery systems, as treated as preliminary knowledge. For instance, the fact statement "Hillary Clinton worked as U.S. Secretary of State" was only valid between the years 2008 and 2013. Without considering the fact's temporal information, a Q&A system may respond to the question of "who is the U.S. Secretary of State" with the wrong answer of "Hillary Clinton".

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 99–107, 2018.
https://doi.org/10.1007/978-3-319-91455-8_9

Table 1. A running example

Problem	Get the valid times corresponding to a statement (e.g. Hillary Clinton worked as U.S. Secretary of State)
Query keywords	The keywords of the statement (e.g. {Hillary Clinton, U.S. Secretary})
Snippets	Relevant information on Web data (e.g. s_1: Clinton meets Saudi king amid Syria, Iran tensions — Reuters Saudi King Abdullah (R) meets with Hillary (C) in Riyadh March 30, 2012. s_2: Hillary campaign staffs up in Arizona - azcentral.com Former Secretary Hillary, the 2016 Democratic front-runner.)
Candidates	Extracted times in snippets (e.g. Years 1992, 2000, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2016)
Output	Valid times corresponding to the statement (e.g. Years 2008, 2009, 2010, 2011, 2012, 2013)

Existing studies on this temporal attribute extraction are mainly based on natural language technologies, and two main tools really stand out, namely, TimeML and DAML-Time [2,8]. These tools automatically extract and produce annotations on time, events and their correlations for temporal text information. In spite of that, as the world evolves and all kinds of events are generated every day, there are still some limitations in existing natural language processing technologies such as the ones cited above, for instance: 1. The correspondence between facts and their validity times must be directly stated in the text in the form of a lexical dependency. This constraint makes it hard to relate the fact to its corresponding validity time, especially for a complicated fact that is usually represented by many relevant events. 2. The free text to be processed may be noisy and misleading, thus the extracted validity times may be incorrect. Therefore, reasoning about the validity time of a specific fact statement becomes of a great importance.

In this paper, we study how to reason about the validity time point/interval of a fact statement based on the rich but noisy Web data. A running example is shown in Table 1. The major challenge of Web-based validity time reasoning is the semantic complexity existent among keywords due to the unstructured and noisy nature of Web data. In the snippet s_1 shown in Table 1, the time "2012" is valid even though it contains only one matching query keyword, "Hillary Clinton". Semantically, s_1 describes the visit of Hillary Clinton to Saudi Arabia, an action that matches the authority entitled to a Secretary of State, even the "Secretary of State" doesn't appear in the snippet. As a result, the time present in this snippet should be valid. On the other hand, it can be observed in the snippet s_2 that the time "2016" appears along with all the query keywords. It is however *not* a correct value due to the presence of the keyword "former".

To address this challenge, we propose an approach based on Semantic Bayesian Networks, which consists of query keywords, semantic cliques and time points. A semantic clique consists of a corpus of keywords and can be considered to correspond to an event. It enables us to reason about validity times of facts based on events instead of single keywords. To better handle noisy data, we also propose to evaluate keyword risk in validity time reasoning and incorporate it into the reasoning process. Our major contributions can be summarized as follows:

1. Our work studies how to reason about the validity times of a specified fact statement by harnessing the rich but noisy Web resources, while overcoming the limitations of other NLP based tools;
2. We propose an approach based on Semantic Bayesian Networks that reasons upon events and can handle the inherent risk of Web keywords;
3. We validate our approach and keyword risk reduction technique's efficiency by carrying experiments on real test cases.

The remainder of this paper is organized as follows: Sect. 2 presents the framework; Sect. 3 describes the Semantic Bayesian Network and its construction process; Sect. 4 describes validity time reasoning, including the way to compute the validity probability followed by valid time points/intervals selection procedure; Sect. 5 presents our experimental evaluation results.

2 Framework

The framework, as shown in Fig. 1, consists of the following components: Query Keyword Extraction (QKE), Relevant Snippet Retrieval (RSR), Semantic Bayesian Network Construction (SBNC) and Validity Time Reasoning (VTR). The function of each component is described below:

1. **[Query Keyword Extraction].** Existing mainstream search engines are keyword-based, the QKE component extracts query keywords from a target

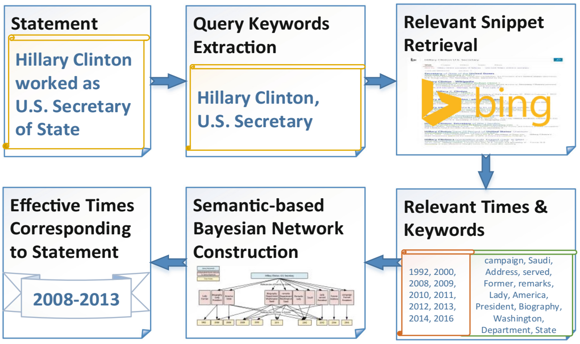

Statement
Hillary Clinton worked as U.S. Secretary of State
Query Keywords Extraction
Hillary Clinton, U.S. Secretary
Relevant Snippet Retrieval
bing
Relevant Times & Keywords
1992, 2000, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2016
campaign, Saudi, Address, served, Former, remarks, Lady, America, President, Biography, Washington, Department, State
Semantic-based Bayesian Network Construction
Effective Times Corresponding to Statement
2008-2013

Fig. 1. Framework

statement such that they can be used by the search engine to retrieve information relevant to the statement. In the running example shown in Table 1, the extracted keywords include “Hillary Clinton”, “U.S.A” and “Secretary of State”.

2. [**Relevant Snippet Retrieval**]**.** The RSR component uses a Web search engine to retrieve the information relevant to a statement based on the keywords extracted by the QKE component. We collect the text snippets returned by the search engine for further analysis.
3. [**Semantic Bayesian Network Construction**]**.** The SBNC component constructs a Semantic Bayesian Network for validity time reasoning. The network consists of query keywords, semantic cliques and time points. Spanning over a group of closely related keywords, a semantic clique corresponds to an event relevant to the target statement (refer to Sect. 3 for more details on the reason behind the use of a clique structure).
4. [**Validity Time Reasoning**]**.** The VTR component estimates the validity probabilities of different times based on the constructed Bayesian network, then selects the most suitable timepoint/time interval via a specific reasoning process.

3 Semantic Bayesian Network Construction

The structure of the Semantic Bayesian Network is illustrated by the scenario shown in Fig. 2. The first level of the network is a single context node containing the query keywords. The second level is formed by semantic cliques where each one consists of a group of keywords extracted from web snippets. Semantically, a clique can be considered to correspond to a relevant event in the context of the target statement. Finally, the bottom level consists of the different timepoints extracted from the snippets. Moreover, each edge has a weight indicating the relevance between two nodes. In the rest of this section, we first describe the semantic cliques’ construction methodology. After that, we present the process of building the edges between the nodes and setting their weights in the network.

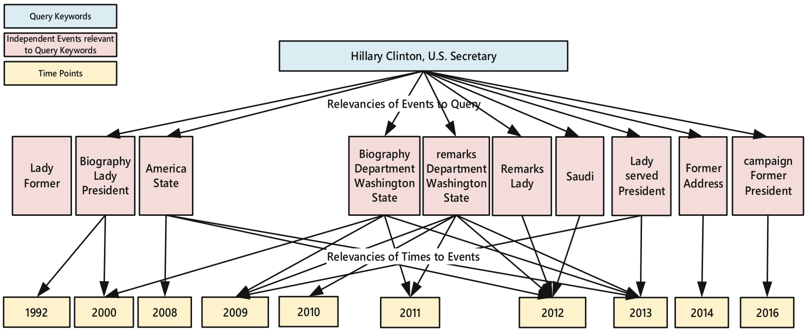

Query Keywords
Independent Events relevant to Query Keywords
Time Points
Hillary Clinton, U.S. Secretary
Relevancies of Events to Query
Lady Former
Biography Lady President
America State
Biography Department Washington State
remarks Department Washington State
Remarks Lady
Saudi
Lady served President
Former Address
campaign Former President
Relevancies of Times to Events
1992
2000
2008
2009
2010
2011
2012
2013
2014
2016

Fig. 2. Semantic Bayesian Network of {Hillary Clinton, U.S. Secretary}

3.1 Semantic Cliques Construction

Consisting of a set of closely correlated keywords, a semantic clique corresponds to an event correlated with the query context. For the construction of semantic cliques, we first depict a undirected probabilistic graphical model called perfect map [1,6] of the keywords, in order to capture their semantic relationships. In the perfect map, an edge between two keywords indicates that they are dependent on each other; that is, they are correlated with the same semantic or event. If two keywords are instead not directly connected by any edge, they are conditionally independent with regard to their common neighbors; hence, they are supposed to describe different semantics or events. In this fashion, we can detect all the semantically independent events in a perfect map by enumerating its maximal cliques, which can capture the most complete semantics of individual events. Existing techniques for maximal clique enumeration [4,10] can be employed to fulfill this task.

We construct the perfect map of keywords by taking advantage of the technique proposed in [1], which is based on Conditional Independence (CI) test. In our scenario, we perform the CI test based on the conditional point-wise mutual information between two keywords, which represent their conditional dependence. The conditional PMI between two keywords v_i and v_j with regard to a set of keywords C is computed by:

$$I(v_i, v_j|C) = \log \frac{P_o(v_i, v_j|C)}{P_o(v_i|C)P_o(v_j|C)}. \tag{1}$$

in which $P_o(\cdot)$ denotes the occurrence probability, which corresponds to the percentage of snippets containing the target keyword or keyword set in the whole set of analyzed snippets. The detailed semantic cliques construction algorithm can be found in our technical report [3].

3.2 Edge Construction

Since every semantic clique is relevant to Q, there is an edge between the query node Q and each semantic clique in the Bayesian network. However, there is no edge between any pair of semantic cliques since they are supposed to be conditionally independent. The edges between semantic cliques and timepoints is constructed as follows: The valid semantic of a timepoint t can be supposed to be represented by the keyword set of the snippets containing t. For each snippet s_i containing t, we denote the keywords contained in s_i by $K(s_i)$. Regarding the relationship between s_i and t, we consider the following two cases:

1. If there exists any semantic clique, C_i, whose keyword set is completely covered by $K(s_i)$, then we add an edge between t and C_i. In this case, the semantic context of s_i and t is supposed to match one and only semantic context, that is C_i's, and none of the other cliques.
2. Otherwise, the semantic context of s_i and t can partially match any semantic clique whose keyword set has a certain overlap with $K(s_i)$. Therefore, we create an edge between t and any semantic clique that satisfies said constraint.

4 Validity Time Reasoning

4.1 Validity Probability Estimation

Given a semantic Bayesian network, the validity probability of a time t with regard to the query Q, denoted as $P(t|Q)$, can be estimated by the law of total probability as follows

$$P(t|Q) = \sum_{\mathbf{C_t}} P(t|\mathbf{C_t}) \cdot P(\mathbf{C_t}|Q), \tag{2}$$

in which $\mathbf{C_t}$ denotes the conjunction of the semantic cliques connected to t. In $\mathbf{C_t}$, the events corresponding to a semantic clique c_i can be specified to *occur* (denoted by C_i) or *not occur* (denoted by $!C_i$). Suppose that $\mathbf{C_t} = \{C_l, C_{l+1}, \ldots, C_m, !C_{m+1}, \ldots, !C_n\}$. Due to the conditional independence of semantic cliques, we have

$$P(\mathbf{C_t}|Q) = \prod_{l \leq i \leq m} P_o(C_i|Q) \cdot \prod_{m+1 \leq i \leq n} P_o(!C_i|Q). \tag{3}$$

Since the semantic contexts of t are described in the snippets containing them, we estimate $P(t|C_t)$ based on the snippets by

$$P(t|\mathbf{C_t}) = \sum_{s_k} P(t|s_k) \cdot P(s_k|\mathbf{C_t}). \tag{4}$$

In Eq. 4, $P(t|s_k) = 0$ if s_k does not contain t; otherwise, $P(t|s_k) = \frac{1}{|T_k|}$, where $|T_k|$ denotes the total number of distinct times contained in s_k. The conditional probability of $P(s_k|\mathbf{C_t})$ is computed as the percentage of the keywords of C_t contained by s_k as

$$P(s_k|\mathbf{C_t}) = \frac{|K(s_k) \cap (\bigcup_{C_i \in C_t^+} C_i)|}{|K(s_k)|}, \tag{5}$$

where C_t^+ denotes the set of semantic cliques in $\mathbf{C_t}$ whose corresponding event is specified to occur.

4.2 Risk-Aware Probability Estimation

Due to the semantic complexity of noisy keywords, the semantic meaning of a clique expressed by keywords may distort the semantic meaning of the original query even though these keywords co-occur in the retrieved snippets with high frequency. Similar to risk analysis in investment theory [9], we regard every keyword as an investment and measure its return rates by its impact on the validity probability evaluation of different time points. Accordingly, the risk of a keyword can be measured by the relative standard deviation (a standardized measurement of dispersion of a probability or frequency distribution) of its impact rates on all the candidate timepoints as

$$R(\mathrm{w}) = \frac{\mathrm{D}(\mathrm{Imp}(\mathrm{w}, \mathrm{t_i}))}{\mathrm{E}(\mathrm{Imp}(\mathrm{w}, \mathrm{t_i}))}, \tag{6}$$

where $E(Imp(\mathrm{w}, \mathrm{t_i}))$ and $D(Imp(\mathrm{w}, \mathrm{t_i}))$ represent the expectation and the standard deviation of the impact rate of a word w, $Imp(\mathrm{w}, \mathrm{t})$, respectively. We quantify $Imp(\mathrm{w}, \mathrm{t})$ by:

$$Imp(\mathrm{w}, \mathrm{t}) = \Delta\mathrm{P}(\mathrm{w}, \mathrm{t}) = \frac{\mathrm{P}(\mathrm{t}|\mathrm{Q}) - \mathrm{P}_{\backslash\mathrm{w}}(\mathrm{t}|\mathrm{Q})}{\mathrm{P}(\mathrm{t}|\mathrm{Q})}, \tag{7}$$

where

$$P_{\backslash\mathrm{w}}(t|Q) = \sum_{\mathbf{C_t}(\backslash\mathrm{w})} P(t|\mathbf{C_t}(\backslash\mathrm{w})) \cdot P(\mathbf{C_t}(\backslash\mathrm{w})|Q), \tag{8}$$

and $\mathbf{C_t}(\backslash\mathrm{w})$ denotes a conjunction of the semantic cliques connected to t but not containing w.

Based on keyword risk estimation, we measure the risk of a semantic clique C_i by the average risk of its keywords, and incorporate the risk of semantic cliques into the probability estimation of $P(\mathbf{C_t}|Q)$, denoting as $P_R(\mathbf{C_t}|Q)$:

$$P_R(\mathbf{C_t}|Q) = \prod_{C_i \in \mathbf{C_t}} \frac{1}{e^{\frac{1}{|C_i|} \sum\limits_{\mathrm{w_j} \in C_i} R(\mathrm{w_j})}} \cdot P_o(C_i|Q) \cdot \prod_{!C_j \in \mathbf{C_t}} P_o(!C_j|Q). \tag{9}$$

Finally, the risk-aware validity probability of time t, $P_R(t|Q)$, is estimated by

$$P_R(t|Q) = \sum_{\mathbf{C_t}} P(t|\mathbf{C_t}) \cdot P_R(\mathbf{C_t}|Q). \tag{10}$$

4.3 Validity Times Selection

There are three categories of temporal attribute values in practice: one single timepoint, multiple timepoints and time intervals. In case there is only a single timepoint to look for, we always select the one with the highest validity probability according to the principle of maximum likelihood. In the case of multiple timepoints or time intervals, we will classify the candidate times into two categories by 2-means clustering technique, and select the timepoints or consecutive time interval in valid category of higher probabilities. The details of validity times selection can be found in our technical report [3].

5 Experiments

For each type of distribution on valid times, we collect the hot query objects of different topics submitted in portal sites in recent years. For each query object, we use the Microsoft Bing search engine to retrieve relevant snippets, then use the

Stanford CoreNLP [7] POS-Tagger to extract candidate time points from those snippets. We limit candidate times units to years in all our experiments, since a year is the mostly mentioned time granularity. The keywords are extracted by the online app IBM Bluemix [5], and are further supplemented with high-frequency nouns and verbs. Finally, we extract 372 candidate times of 20 topics in our experimental evaluation.[1]

Table 2. Results of single validity timepoint reasoning experiments

Topic	Freq	TimeML	SBNR−RD	SBNR+RD
	PR	PR	PR	PR
AMD graphics introduce	0.60	**0.80**	**0.80**	**0.80**
Falcom RPG release	0.25	**0.75**	0.50	**0.75**
Harry Potter movie	**1.00**	**1.00**	0.80	0.80
HUAWEI launch	**0.75**	**0.75**	**0.75**	**0.75**
Intel core release	**0.40**	**0.40**	**0.40**	**0.40**
Explorer release	0.20	**1.00**	**1.00**	**1.00**
iOS release	**1.00**	**1.00**	**1.00**	**1.00**
iPad launch	**0.83**	**0.83**	**0.83**	**0.83**
iPhone discontinued	0.60	0.60	**0.80**	**0.80**
Olympic games hold	**1.00**	**1.00**	**1.00**	**1.00**
Robert Downey movie	0.43	**0.86**	**0.86**	**0.86**
Samsung unpack	**1.00**	0.80	**1.00**	**1.00**
Blizzard release	**0.83**	**0.83**	**0.83**	**0.83**

Table 3. Results of multiple validity timepoints reasoning experiments

Topic	Freq			TimeML			SBNR−RD			SBNR+RD		
	RC	PR	F1	RC	PR	F1	RC	PR	F1	RC	PR	F1
MacOS versions	0.63	0.83	0.71	**1.00**	0.73	0.84	0.88	0.78	0.82	**1.00**	**0.89**	**0.94**
Sony release	**0.75**	0.67	**0.71**	0.56	**0.90**	0.69	**0.75**	0.67	**0.71**	**0.75**	0.67	**0.71**
Visual Art's TV anime	0.80	0.73	0.76	0.50	**0.83**	0.63	0.90	0.64	0.75	**1.00**	0.67	**0.80**

We compare the performance of proposed Semantic Bayesian Network with Risk Reduction (denoted by SBNR+RD) with 2 alternatives, Occurrence Frequency (denoted by Freq) and TimeML, on the metrics of recall (abbr. RC), precision (abbr. PR) and F1-Score (abbr. F1). Moreover, in order to estimate

[1] All the data files of our retrieved relevant snippets, extracted keywords and candidate years of the query objects of different topics are available at http://www.wowbigdata.com.cn/ValidTimeReasoning.zip.

Table 4. Results of validity time interval reasoning experiments

Topic	Freq			TimeML			SBNR−RD			SBNR+RD		
	RC	PR	F1	RC	PR	F1	RC	PR	F1	RC	PR	F1
Apple CEO	0.86	0.86	0.86	0.36	**1.00**	0.53	0.83	0.93	0.90	**0.93**	**1.00**	**0.96**
Japan Prime Minister	**0.94**	0.71	0.81	0.31	**1.00**	0.48	**0.94**	0.94	**0.94**	**0.94**	0.94	**0.94**
Microsoft Chairman	**0.73**	**0.42**	**0.53**	0.00	0.00	0.00	**0.73**	**0.42**	**0.53**	**0.73**	**0.42**	**0.53**
U.S. Secretary	0.80	0.73	0.76	0.50	**1.00**	0.67	**1.00**	0.77	0.87	**1.00**	**1.00**	**1.00**

the added value of the risk reduction component, we perform an ablation experiment by estimating our proposed approach where we remove the RD component (abbr. SBNR−RD). The comparative results for the topics with single validity timepoint, multiple validity timepoints and validity time interval are presented in Tables 2, 3 and 4 respectively. From the results, it can be observed that our proposed Semantic Bayesian Network with Risk Reduction approach performs better than all the other alternatives.

References

1. Cheng, J., Bell, D.A., Liu, W.: Learning belief networks from data: an information theory based approach. In: Proceedings of the Sixth International Conference on Information and Knowledge Management, pp. 325–331. ACM (1997)
2. Hobbs, J., Pustejovsky, J.: Annotating and Reasoning About Time and Events (2003)
3. Hou, B., Nafa, Y.: Enabling temporal reasoning for fact statements: a web-based approad (Technical report). Technical report (2018). http://www.wowbigdata.com.cn/Temporal-reasoning-technical-report.pdf
4. Hou, B., Wang, Z., Chen, Q., Suo, B., Fang, C., Li, Z., Ives, Z.G.: Efficient maximal clique enumeration over graph data. Data Sci. Eng. **1**(4), 219–230 (2016)
5. IBM: Alchemyapi. http://www.alchemyapi.com/api/keyword/textc.html
6. Koller, D., Friedman, N.: Probabilistic Graphical Models - Principles and Techniques. The MIT Press, Cambridge (2009)
7. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The stanford CoreNLP natural language processing toolkit. In: ACL (System Demonstrations), pp. 55–60 (2014)
8. Schilder, F., Katz, G., Pustejovsky, J.: Annotating, extracting and reasoning about time and events. In: Schilder, F., Katz, G., Pustejovsky, J. (eds.) Annotating, Extracting and Reasoning about Time and Events. LNCS (LNAI), vol. 4795, pp. 1–6. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75989-8_1
9. Wachowicz, J.M., Shrieves, R.E.: An argument for "generalized" mean-coefficient of variation analysis. Financ. Manage. **9**(4), 51–58 (1980)
10. Wang, Z., Chen, Q., Hou, B., Suo, B., Li, Z., Pan, W., Ives, Z.G.: Parallelizing maximal clique and k-plex enumeration over graph data. J. Parallel Distrib. Comput. **106**, 79–91 (2017)

Time Series Cleaning Under Variance Constraints

Wei Yin[1], Tianbai Yue[2], Hongzhi Wang[1](✉),
Yanhao Huang[3], and Yaping Li[1]

[1] Department of Computer Science and Technology,
Harbin Institute of Technology, Harbin, China
764406111@qq.com, {wangzh, lyp}@hit.edu.cn
[2] Harbin Institute of Petroleum, Harbin, China
letianbai1005@qq.com
[3] Electric Power Research Institute, Beijing, China
hyhao@epri.sgcc.com.cn

Abstract. A time series is a series of data points indexed (or listed or graphed) in time order. Errors are prevalent in time series, such as GPS trajectories or sensor readings. The quality of the time series greatly influences the authenticity and confidence of other operations. Existing methods on cleaning sequential data employ a constraint on value changing speeds and perform constraint-based or statistics-based repairing. However, such speed-based methods are difficult to identify and repair outliers, which does not significantly deviate from the true value and does satisfy the speed constraint. And such a statistics-based method is not perfect in terms of efficiency and accuracy. To handle such problem of time series cleaning, in this paper, we propose a first variance-based approach for cleaning time series. In order to support the stream computation, we consider the data in a window as a whole and adopt the idea of sliding window to solve the problem in stream computation.

Keywords: Time series · Data clean · Variance · Constraints

1 Introduction

A time series is a set of statistics, usually collected at regular intervals. Time series data occur naturally in many application areas. Applications include word spotting, object recognition and image retrieval systems [8], sensor pattern matching, DNA sequence analysis [6], and monitoring of bio-medical signals (e.g., EKG, ECG), and monitoring of environmental (seismic and volcanic) signals [3]. In the medical field, doctors observe the sequence of the patient's electrocardiogram (ECG) and electroencephalogram (EEG), and diagnose whether the patient is healthy by analyzing whether there is inconsistency [2]. But time series often has many anomaly values that can affect the analysis results of the application. Time series cleaning deals with detecting and removing errors and inconsistencies from data in order to improve the quality of data.

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 108–113, 2018.
https://doi.org/10.1007/978-3-319-91455-8_10

Keogh [11] proposed a sliding window smoothing technology based on time series segmentation. Papotti [12] takes a greedy strategy, it iteratively modifies the abnormal data detected in each round, which may introduce new violations to other data points, causing another round of repair. Song [10] employs a constraint on value changing speeds and performs constraint-based repairing. While such speed constraints are effective in identifying large spike errors, the small errors that do not significantly deviate from the truth and indeed satisfy the speed constraints can hardly be identified and repaired.

In this paper, we study the problem of data cleaning over time series that require summarization: our job is to find out some kind of points, which make the variance of the data in the window of size w greater than a certain threshold v. Consider the anomaly data point we found and fix it so that the window satisfies the variance constraint. Select a solution closer to the original value as a candidate. Calculate the optimal solution through the candidate set using the weighted average method.

2 Algorithm

2.1 Preliminaries

Consider a time series $x = x_1, x_2 \ldots\ldots,$ where each x_i has a timestamp t_i. And x_i represents the $i-th$ data. Our job is to find out some kind of points, which make the variance of the data in the window of size w greater than a certain threshold v, we call it an outlier.

$$D(x_i) = \frac{\sum_{i=1}^{w} (x_i - \mu)^2}{w} \leq v \tag{1}$$

where μ is the mean of x_i in a window.

w represents a period of time. Variance measures how far a set of (random) numbers are spread out from their average value. It has a central role in real setting. For example, the temperature should not be much difference within a few days.

Example 1: Consider a time series x = {3, 4, 5, 4, 2, 5} of six data points, with timestamps t = {1, 2, 3, 4, 5, 8}. Figure 1 illustrares the data points. Suppose that the variance threshold is $v = \frac{1}{2}$.

For a window size w = 2 in the variance constraint, data points x_1 and x_2, with timestamps distance 1 < 2, scilicet in a window, satisfy the variance constraint, since the variance is $\frac{(3-3.5)^2+(4-3.5)^2}{2} = \frac{1}{4} < \frac{1}{2}$. But the data points x_4 and x_5, with timestamps distance 1 < 2, are identified as violation since the variance is $\frac{(4-3)^2+(2-3)^2}{2} = 1 > \frac{1}{2}$. And although x_5 and x_6 differ greatly, their timestamp distance is 3 and they are not in a window, they have no violations.

2.2 Optimal Solution

Consider the anomaly data point we found and fix it so that the window satisfies the variance constraint. As we all know, quadratic equations generally have two solutions. Time series cleaning should not have two repairs. We can use the following method to choose the optimal solution in the two solutions. In order to ensure that the minimum change from the original data in data cleaning. The repair changes is evaluated by the difference between the original x and the repaired x':

$$\Delta(x, x') = \sum_{x_i \in x} |x_i - x_i'| \tag{2}$$

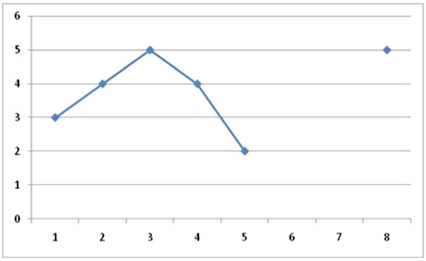


Fig. 1. Example 1

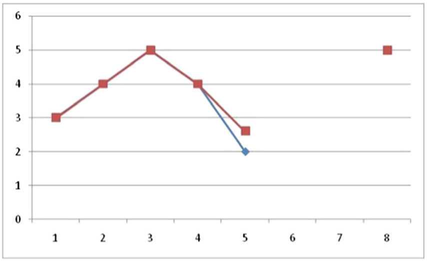


Fig. 2. Example 2

Example 2: Consider again the time series x = {3, 4, 5, 4, 2, 5}. Let x_5 be y, let's find the solution of the equation $\frac{(4-\frac{4+y}{2})^2+(x-\frac{4+y}{2})^2}{2} = \frac{1}{2}$. We can get two solutions $y_1 = 2.6$, $y_2 = 5.4$. Referring to the minimum change principle, we choose y_1 as the optimal solution, in Fig. 2.

Since the repair of the time series is repaired in order, there are

$$D(x_i) = \frac{\sum\limits_{i=k-w}^{k} (x_i' - \mu)^2}{w} \leq v, \text{where } \mu = \frac{\sum\limits_{i=k-w}^{k} x_i'}{w}.$$

x_k' is the repair value of the data point that is currently processed.

Let x_k^* be the processed data set in time series. x_{k+1} is the next data point to processed. If x_{k+1} satisfies the variance constraint, $x_{k+1}' = x_{k+1}$. Otherwise, we should solve the equation

$$D(x_i) = \frac{\sum_{i=k+1-w}^{k+1} (x_i' - \mu)^2}{w} = v \tag{3}$$

And we will get two solutions y_1 and y_2. Define a convention here is that y_1 is the smaller of the two solutions and y_2 is the bigger one. Referring to the minimum change principle, if $x_{k+1} \leq y_1$, we select y_1 for repair value of x_{k+1}, else if $x_{k+1} \geq y_2$, we select y_2.

As following:

$$x'_{k+1} = \begin{cases} y_1 & x_{k+1} \leq y_1 \\ y_2 & x_{k+1} \geq y_2 \\ x_{k+1} \quad x_{k+1}\ satisfies & constra\text{int}s \end{cases} \tag{4}$$

2.3 Candidates

consider a time series $x = \ldots\ldots x_1, x_2 \ldots\ldots x_\mathrm{k} \ldots\ldots,$ and sub-series $x' = x_1', x_2' \ldots\ldots x_{\mathrm{k}-1}'$ have been repaired in the previous steps. Each window containing x_k, $[x_i' \ldots x_k \ldots x_{i+w}]$, $\forall k \leq i < k+w$, specifies a candidate for x_k. We denote the candidate set with X_k and each candidate with $x_k^{(i)}$. There should be w values in X_k with window size w. Then, how do we find the optimal solution of x_k' through the candidate set? We use the weighted average method to calculate the repair value of x_k here. The data points before x_k have been repaired a.k.a. they are clean. Therefore, the weights are given based on the following principles. For each window that contains x_k, where the more data points before x_k, the greater the weighting of the candidate that result from it.

$$x_k' = \frac{f_1 x_k^{(1)} + f_2 x_k^{(2)} + \ldots\ldots + f_w x_k^{(w)}}{w}, f_1 < f_2 < \ldots\ldots < f_w \tag{5}$$

Where $x_k^{(1)}$ is the candidate for window $x_k, x_{k+1}, \ldots\ldots, x_{k+w-1}$, and $x_k^{(i)}$ is the candidate for window who has i data points before or equal to x_k, i.e. $x_{k-i+1}, \ldots\ldots x_k, \ldots\ldots, x_{k-i+w}$.

Algorithm 1 presents the repair algorithm of a time series under the variance

```
 Algorithm 1: Repair($x$, $v$)
 Data: a time series $x$ and variance constraint $v$
 Reault: a repair $x^{'}$ of $x$
 for $k$ ← 1 to $n$ do
 $X_k$ ← $\phi$
 for $i$ ← $k-w+1$ to $k$ do
   if $x_i ...... x_{i+w-1}$ satisfies variance constraint $v$
     $X_k$ ← $X_k \cup x_k$ ($x_k^{(i+w-k)} = x_k$)
   else complete $y_1$, $y_2$ refer to variance constraint $v$ ($y_1 < y_2$)
     if $x_k \le y_1$
       $X_k$ ← $X_k \cup y_1$ ($x_k^{(i+w-k)} = y_1$)
     else if $x_k \ge y_2$
       $X_k$ ← $X_k \cup y_2$ ($x_k^{(i+w-k)} = y_2$)
 $x_k^{'} = \frac{f_1 x_k^{(1)} + f_2 x_k^{(2)} + ...... + f_w x_k^{(w)}}{w}, f_1 < f_2 < ...... < f_w$
return $x^{'}$
```

constraint. For each point k in the series, Line 3 to 10 computes the candidate for each window containing x_k. By considering all the candidates in X_k, Line 11 computes the x_k' through the weighted average method.

It is easy to see that the number of data points in a window is at most w. The $x_k^{(i)}$ in the window can be found in $O(w)$. Considering all the n data points in the sequence, Algorithm 1 runs in $O(nw)$ time.

3 Conclusions

In this paper, we study the problem of data cleaning over time series that require summarization: our job is to find out some kind of points, which make the variance of the data in the window of size w greater than a certain threshold v. Consider the anomaly data point we found and fix it so that the window satisfies the variance constraint. Select a solution closer to the original value as a candidate. Calculate the optimal solution through the candidate set using the weighted average method.

Acknowledgement. This paper was supported by the National Key Research and Development Program of China 2016YFB1000703.

References

1. Wu, H., Salzberg, B., Zhang, D.: Online event-driven subsequence matching over financial data streams. In: ACM SIGMOD International Conference on Management of Data, Paris, France, pp. 23–34. DBLP, June 2004
2. Vullings, H.J.L.M., Verhaegen, M.H.G., Verbruggen, H.B.: ECG segmentation using time-warping. In: Liu, X., Cohen, P., Berthold, M. (eds.) IDA 1997. LNCS, vol. 1280, pp. 275–285. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0052847
3. Sakurai, Y., Faloutsos, C., Yamamuro, M.: Stream monitoring under the time warping distance. In: IEEE International Conference on Data Engineering, pp. 1046–1055. IEEE (2007)
4. Lane, T., Brodley, C.E.: Temporal sequence learning and data reduction for anomaly detection. ACM Trans. Inf. Syst. Secur. **2**(3), 295–331 (1900)
5. Bu, Y., Chen, L., Fu, W.C., et al.: Efficient anomaly monitoring over moving object trajectory streams. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 159–168. ACM (2009)
6. Zhang, M., Kao, B., Cheung, D.W., et al.: Mining periodic patterns with gap requirement from sequences. ACM Trans. Knowl. Discov. Data **1**(2), 7 (2007)
7. Keogh, E., Lin, J., Fu, A.: HOT SAX: efficiently finding the most unusual time series subsequence. In: IEEE International Conference on Data Mining. IEEE, pp. 226–233 (2006)
8. Yankov, D., Keogh, E., Wei, L., et al.: Fast best-match shape searching in rotation-invariant metric spaces. IEEE Trans. Multimedia **10**(2), 230–239 (2008)
9. Yankov, D., Keogh, E., Medina, J., et al.: Detecting time series motifs under uniform scaling. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, pp. 844–853. DBLP, August 2007
10. Song, S., Zhang, A., Wang, J., et al.: SCREEN: stream data cleaning under speed constraints, pp. 827–841 (2015)
11. Keogh, E.J., Chu, S., Hart, D., et al.: An online algorithm for segmenting time series. In: IEEE International Conference on Data Mining, pp. 289–296. IEEE (2001)
12. Papotti, P., Chu, X., Ilyas, I.F.: Holistic data cleaning: putting violations into context. In: IEEE International Conference on Data Engineering, pp. 458–469. IEEE Computer Society (2013)

Entity Resolution in Big Data Era: Challenges and Applications

Lingli Li(✉)

Heilongjiang University, Harbin, China
lilingli@hlju.edu.cn

Abstract. Entity resolution plays an important role in many fields. Due to its importance, it has been widely studied. However, in big data era, entity resolution brings new challenges including high scalability, coexistence of tautonymy and synonym, complex similarity metrics as well as the requirement of data quality evaluation based on entity resolution. Facing these challenges, we introduce our solutions briefly and discuss the possible future work for entity resolution in big data era.

Keywords: Entity resolution · Big data · Algorithm

1 Introduction

Entity resolution plays an important role in data quality management (DQM) [1], data management [2] and information retrieval [3,4]. It is also an important research area in DQM. A real-world entity may appear in one or multiple databases which may have quite different descriptions. The goal of entity resolution (ER) is to identify the records referring to the same real-world entity from multiple data sources. The result of entity resolution is widely used in other steps of data quality management, such as data cleaning and data quality evaluation. The problem that a real-world entity have quite different descriptions is a common problem that appears in many kinds of application areas. Because of its importance, entity resolution has attracted much attention in the literature [5–7]. Even though existing methods can perform ER effectively in many cases, for big data era, these ER approaches have following limitations.

- There are two problems in entity resolution, called "tautonymy" and "synonym". Tautonymy is different entities may share the identical name and synonym is different names may correspond to the identical entity. However, current research focuses on only one of the problems, without considering the general cases where both of the problems might exist.
- Traditional ER approaches obtain a result based on similarity comparison among records. They assume that records referring to the same entity are more similar to each other, called "compact set property". However, such property may not hold, so traditional ER approaches cannot identify records correctly in some cases.

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 114–117, 2018.
https://doi.org/10.1007/978-3-319-91455-8_11

- The similarity metrics used by current ER approaches do not consider the correlation between words in records and the major contribution of some specific words which describe the important features of real-world entities in entity identification. As a result, the entity resolution approaches based on current metrics sometimes cannot achieve a high performance.
- Currently, the study of data quality evaluation only includes consistency, currency, completeness and accuracy. However, a new kind of data quality problem can be evaluated according to the result of entity resolution, that is duplicated data have conflicting values in the same attributes. We call this problem as "the entity description conflict". As far as we know, the evaluation approach of entity description conflict in duplicated data has not been studied.

On the basis of the above analysis, in the background of big data era, focusing on the objectives of minimizing time complexities and maximizing the accuracy of ER result, this paper gives the solutions of entity resolution on big data. Specially, we discusses the graph-based entity resolution algorithm, the rule-based entity resolution algorithm, the entity resolution algorithm based on distance metric and the data quality evaluation algorithm based on entity resolution result for entity resolution on big data.

2 Solutions

2.1 Rule-Based Entity Resolution

The syntax and semantics of the rules for ER are designed, and the independence, consistency, completeness and validity of the rules are defined and analyzed. An efficient rule discovery algorithm and an efficient rule-based algorithm for solving entity resolution problem are proposed and analyzed in [8]. A rule maintaining method is proposed when entity information is changed. Experiments are performed on real data to verify the effectiveness.

2.2 Evaluating Entity-Description Conflict on Duplicated Data

The mathematical model of the entity-description conflict is defined based on the conflicts between attribute-values in a cluster. The problem of computing the range of entity-description conflict is proposed when the accuracy of ER-result is not 100%. To solve the problem, four primary operators are identified in [9], and it is proved that the problem of computing the range of the entity-description conflict is NP-hard. Four approximation algorithms for the four primary operators with ratio bound assurance are provided. A framework based on the four primary operators is proposed for computing the range of the entity-description conflict. Using real-life data and synthetic data, the effectiveness and efficiency of the proposed algorithm are experimentally verified.

2.3 Graph-Based Entity Resolution

The problems of "tautonymy" and "synonym" are introduced. As far as we know, [10] is the first study to address these problems. A general entity identification framework, EIF, is presented. In this framework, the similarity relationships between records have been modeled as a graph, entities are identified by exploiting the graph clustering algorithms. As an application of EIF, an author identification algorithm is proposed by using the information of author names and co-authors to solve author identification problem. The effectiveness of this framework is verified by extensive experiments. The experimental results show that the author identification algorithm based on EIF outperforms the existing author identification approaches both in precision and recall.

2.4 Entity Resolution Based on Distance Metric

A key component for ER is to choose a proper distance (similarity) function for each database field to quantify the similarity of records. Most existing ER approaches focus on how to define a proper matching rule based on generic or hand-crafted distance metrics. Two learnable string distance metrics for two kinds of ER problems are explored by employing the Principle Component Analysis (PCA) and the Largest Margin Nearest Neighbor Algorithm (LMNN) for training. Experimental results on real datasets show that our approaches can improve entity resolution accuracy over traditional techniques.

3 Conclusions and Discussions

Even though entity resolution has been widely studied, for the challenges brought by big data era, our research is just a start. Many issues are remained to be studied. We list several research problems for future work.

- For scalability issues, parallel entity resolution techniques are in demand. Some approaches have been proposed such as [11–13]. However, the efficiency could be increased especially for some scenarios with real-time requirements [3]. Thus, the scalable and efficient entity resolution algorithms for big data are to be studied.
- In practise, records may come from heterogenous data sources with various schema even data model. This makes traditional distance-based approaches invalid due to the heterogeneity in schema. It is still a problem to study how to perform efficient entity resolution over heterogenous data.
- The variety of big data brings new chances for entity resolution. For example, the entity resolution for products could be conducted on not only textual description but also the pictures of products. However, it is not trivial to conduct entity resolution on multi-modal data as the combination of record matching, anaphora resolution and pattern recognition. Many problems are remained to be studied.

– Other data quality issues such as incompleteness and inconsistency brings difficulty for entity resolution. Entity resolution and truth discovery could improve the data quality and help the cleaning of other data quality problem. Thus, involving entity resolution into the whole data cleaning process raises new research issues including the opportunity of entity resolution in the data cleaning processing and the entity resolution algorithms on dirty data.

Acknowledgements. This work was supported by NSFC61602159, 61370222.

References

1. Fan, W., Geerts, F., Wijsen, J.: Determining the currency of data. ACM Trans. Database Syst. **37**(4), 25 (2012)
2. Wang, H., Li, J., Gao, H.: Data model for dirty databases. J. Softw. **23**(3), 539–549 (2012)
3. Wang, H., Zhang, X., Li, J., Gao, H.: ProductSeeker: entity-based product retrieval for e-commerce. In: The 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1085–1086. ACM (2013)
4. Wang, L., Zhang, R., Sha, C., Wang, X., Zhou, A.: A product normalization method for e-commerce. Chin. J. Comput. **34**(2), 312–325 (2014)
5. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate record detection: a survey. IEEE Trans. Knowl. Data Eng. **19**(1), 1–16 (2007)
6. Koudas, N., Sarawagi, S., Srivastava, D.: Record linkage: similarity measures and algorithms. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, pp. 802–803. ACM (2006)
7. Wang, H., Fan, W.: Object identification on complex data: a survey. Chin. J. Comput. **34**(10), 1843–1852 (2011)
8. Li, L., Li, J., Gao, H.: Rule-based method for entity resolution. IEEE Trans. Knowl. Data Eng. **27**(1), 250–263 (2015)
9. Li, L., Li, J., Gao, H.: Evaluating entity-description conflict on duplicated data. J. Comb. Optim. **31**(2), 918–941 (2016)
10. Li, L., Wang, H., Gao, H., Li, J.: EIF: a framework of effective entity identification. In: Chen, L., Tang, C., Yang, J., Gao, Y. (eds.) WAIM 2010. LNCS, vol. 6184, pp. 717–728. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14246-8_68
11. Altowim, Y., Mehrotra, S.: Parallel progressive approach to entity resolution using MapReduce. In: 33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, 19–22 April 2017, pp. 909–920 (2017)
12. Ma, K., Yang, B.: Parallel NoSQL entity resolution approach with MapReduce. In: 2015 International Conference on Intelligent Networking and Collaborative Systems, INCoS 2015, Taipei, Taiwan, 2–4 September 2015, pp. 384–389 (2015)
13. Huo, R., Wang, H., Zhu, R., Li, J., Gao, H.: Map-reduce based entity identification in big data. J. Comput. Res. Dev. **50**(2), 170–179 (2013)

Filtering Techniques for Regular Expression Matching in Strings

Tao Qiu$^{(\boxtimes)}$, Xiaochun Yang, and Bin Wang

School of Computer Science and Engineering,
Northeastern University, Liaoning 110819, China
qiutao@stumail.neu.edu.cn, {yangxc,binwang}@mail.neu.edu.cn

Abstract. Matching a regular expression (regex) on a text is widely used in many applications, such as text editing, information extraction and instruction detection (IDS). Traditional algorithms generally compile an equivalent automaton from the regex query, then run it on the text to find all matching results. However, they have to scale linearly with the size of the text. Recent algorithms utilize various filtering techniques to quickly jump to candidate positions in a text where a matching result may appear, then only these candidate positions are verified by the automaton. In this paper, we give a full specification on filtering techniques for the regex matching problem, in which filters for the regex query can be classified into positive factor and negative factor. We review three typical positive factors, including prefix, suffix, and necessary factor and show that negative factors can collaborate with positive factors to significantly improve the filtering ability.

Keywords: Regular expression · Filtering technique · Query efficiency

1 Introduction

Regular expression (regex) matching is a fundamental problem that exists in many applications, such as text editing, information extraction, protein sequence matching and instruction detection (IDS). For example, in the domain of bioinformatics, a regex query `TC(T|G)(C|T)A` has the language {`TCTCA`,`TCTTA`,`TCGCA`,`TCGTA`}, matching this regex is to find all matchings of the string in this language from a genome sequence.

The classical approaches to match a regex query in a text is that first transforming the regex into an equivalent automaton, and then running it from each position in the text to verify if the substring is an occurrence of the regex. An

The work is partially supported by the National Natural Science Foundation of China (Nos. 61572122, U1736104, 61532021).

The original version of this chapter was revised: For detailed information please see correction chapter. The correction to this chapter is available at https://doi.org/10.1007/978-3-319-91455-8_24

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 118–122, 2018.
https://doi.org/10.1007/978-3-319-91455-8_12

occurrence is found whenever a final state of the automaton is reached [1,5,8]. NFAThompson [8] and DFAClassical [1] are two typical automaton-based algorithms. NFAThompson is the pioneering work that proposes the Thompson NFA to match a regex with time complexity $O(mn)$, where m is the size of the regex query and n is the length of the text. DFAClassical realizes regex matching by simulating the DFA, which can guarantee a linear search time of $O(n)$. However, the automaton-based algorithms have to check every character in a text, the matching efficiency is largely limited.

To improve query efficiency, filtering techniques have been proposed for many applications which focus on producing a set of candidates which could be the final query results [3,4,7,10,13,14]. To alleviate the above issue in the regex matching problem, many algorithms have been developed under a filtering-and-verification framework, where candidate positions are generated using one or more filters and then verified by an automaton to find the true matching positions [7,11]. The filters can be divided into two types. The first one, called positive factor, utilizes the substrings extracted from the regex query, including prefix, suffix and necessary factor. MultiStringRE [9] computes a set of prefixes for all strings matching the regex query (i.e., the language of a regex), then uses a Commentz-Water-like algorithm to verify the text starting from each occurrence of these prefixes. NRGrep [6] gets the candidate positions using the reversed prefixes of the regex and verifies them using a reversed automaton. GNU grep [2] utilizes the necessary factors to get candidate positions, which are the substring must appear in a regex match. Since a necessary factor could divide a regex into a left and a right part, two automatons are constructed to verify a candidate position in forward and backward directions. The other one is called negative factor and initially proposed in [12], which is the substring that cannot appear in any matching string of a regex. Negative factors can further prune the candidate positions generated by the positive factors.

In this paper, we give a full specification on filtering techniques for the regex matching problem and show different filters of the regex can be used together to improve the filtering ability.

2 Filtering-Based Regular Expression Matching

Let Σ be a finite alphabet. A *regular expression* (regex) Q is a string over $\Sigma \cup \{\epsilon, |, \cdot, *, (,)\}$, in which $\{|, \cdot, *\}$ are the operators that represents disjunction, conjunction and Kleene closure (repeating unit), respectively. We use $L(Q)$ to represent the language of a regex Q. For a text T of the characters in Σ, we use $|T|$ to denote its length, $T[i]$ to denote its i-th character (starting from 0), and $T[i, j]$ to denote the substring ranging from its i-th character to its j-th character.

Regular Expression Matching. Given a regex Q and a text T, the *regex matching problem* is to find matching occurrences of the strings in $L(Q)$ from T.

In the following, we first review the filtering techniques with positive factors, then show negative factors can collaborate with positive factors.

2.1 Computing Candidate Positions Using Positive Factors

Recent techniques have utilized certain features of the regex Q to improve the performance of automaton-based methods [7]. Their main idea is to use *positive factors*, which are substrings of Q, to identify candidate positions of Q in T. Next, we present three typical positive factors, including *prefix*, *suffix*, and *necessary factor*.

A prefix *w.r.t.* a regex Q is defined as a prefix of a string in the language $L(Q)$. We use l_{pre} to denote the length of a prefix. A set of prefixes S_P can be used as the filters of a regex if and only if there is a prefix in S_P for any string in $L(Q)$ [9]. For example, for the regex $Q = \texttt{(A|G)T}^*\texttt{AT}^*\texttt{G}$, the prefixes with $l_{pre} = 2$ are $\{\texttt{AT}, \texttt{AA}, \texttt{GT}, \texttt{GA}\}$. Due to any matching string of Q must start with a prefix in S_P, then the matching positions of the prefixes in S_P on T are the candidate positions for Q. To compute all matches of Q, we only examine these matching positions of prefixes using the automaton of Q.

Similarly, a suffix *w.r.t.* a regex Q is defined as a suffix of a string in $L(Q)$, and the length is denoted by l_{suf}. We use S_S to represent the set of suffixes computed from Q, e.g., for the regex $Q = \texttt{(A|G)T}^*\texttt{AT}^*\texttt{G}$, the suffixes with $l_{suf} = 2$ are $\{\texttt{TG}, \texttt{AG}\}$. Different from prefixes, the ending matching positions of suffixes in S_S are candidate positions, which are verified by a reversed automaton in the backward direction [6].

In addition to prefixes and suffixes, the necessary factor is another type of positive factor, which is a substring that must appear in every matching string in $L(Q)$ [2]. For instance, for the regex $Q = \texttt{(A|G)T}^*\texttt{AT}^*\texttt{G}$, $\{\texttt{A}\}$ is a necessary factor of Q. To verify a candidate position where a necessary factor appears, we can divide Q into a left part and a right part with a corresponding automaton, e.g., two automatons are constructed for the left and right parts of the regex Q (i.e., $\texttt{(A|G)T}^*$ and $\texttt{AT}^*\texttt{G}$).

Instead of independently applying each positive factor, all three types of positive factors can also be leveraged together to further improve the filtering ability [11]. PS and PMS are two typical patterns used to identify candidate positions. PS pattern utilizes prefix and suffix which requires a candidate occurrence contains the matchings of a prefix and a suffix simultaneously in T. Likewise, PMS pattern requires a candidate occurrence contains all matchings of the three positive factors. Generally, PMS pattern can achieve better filtering ability than PS pattern since one more positive factor is considered, but it also needs more computational cost for filtering.

Consider the example in Fig. 1, there is a matching result $T[6, 10]$ for the regex $Q = \texttt{(A|G)T}^*\texttt{AT}^*\texttt{G}$. Using prefixes of Q as filters, there are 6 candidate occurrences needed to be verified. PS and PMS further prune the candidate occurrences when considering more positive factors, and obtain 5 and 4 candidate occurrences, respectively.

2.2 Further Pruning Candidate Positions Using Negative Factors

Although positive factors can be used together to compute candidate occurrences, compared to the single type of positive factors, using more than one type

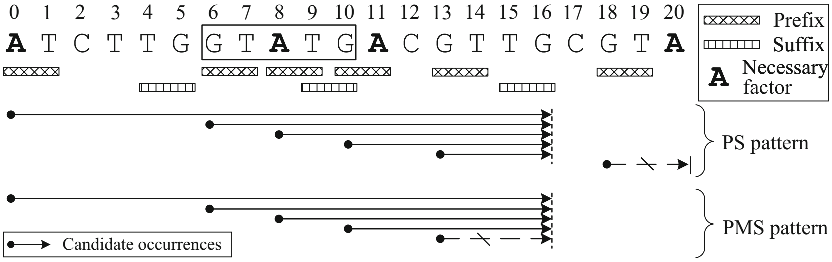


Fig. 1. An example of using positive factors to identify candidate occurrences for the regex $Q =$ (A|G)T*AT*G.

of positive factors obtains few improvements in the filtering ability. Negative factors solve this problem.

A *negative factor* (also called *N-factor*) *w.r.t.* a regex Q is a string w such that there is no string $\Sigma^* w \Sigma^*$ in $L(Q)$ [11,12]. For example, for the running example, C is an N-factor since any string in $L(Q)$ does not contain C. Essentially, N-factor is the substring that does not appear in any matching string of Q. Based on this property, given a set of N-factors of Q, a text T can be divided into several disjoint segments and we can get the matching result of Q can only appear within a segment.

At first, we show N-factors can be integrated into the PS pattern. According to the definition of N-factor, a candidate occurrence must start with a prefix and end with a suffix, and do not contain any matching of N-factor. We call such candidate occurrences satisfy the PNS pattern. For example, as shown in Fig. 2, candidate occurrences $T[0, 16]$ and $T[10, 16]$ obtained by PS pattern can be pruned by the PNS pattern since they contain the matching of N-factor C.

Similarly, we can get the PMNS pattern by integrating N-factors into PMS pattern, which requires a candidate occurrence contains the matchings of necessary factors based on the requirements of the PNS pattern. Because PMNS considers the requirements of all filters computed from the regex, it achieves the best filtering ability. For the example in Fig. 2, compared to the PNS pattern, the candidate occurrence $T[13, 16]$ can be further pruned by the PMNS pattern since it does not contain the matching of A.

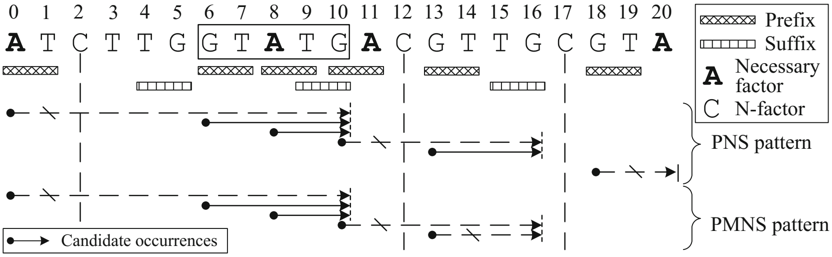


Fig. 2. Using negative factors to further prune candidates generated by positive factors.

3 Conclusion and Future Work

Regular expression matching is a fundamental problem existing in a diverse range of applications. In this paper, we introduced the filtering techniques for the regex matching problem, in which filters of the regex query can be classified into positive factor and negative factor. We reviewed three typical positive factors, including prefix, suffix, and necessary factor and showed they can be used together to compute candidate occurrences. Furthermore, we showed negative factors can collaborate with positive factors to significantly improve the filtering ability. As parts of future work, we will (i) further investigate the correlation between different filters extracted from the regex query; (ii) balance the filtering cost caused by different filters.

Acknowledgment. This paper constituted an invited talk, held at BDQM 2018, a DASFAA 2018 satellite workshop. The main techniques derive from our work cited in [11].

References

1. Aho, A.V., Sethi, R., Ullman, J.D.: Compilers - Principles, Techniques and Tools. Addison-Wesley, Reading (1986)
2. GNUgrep: ftp://reality.sgiweb.org/freeware/relnotes/fw-5.3/fw_gnugrep/gnugrep.html
3. Li, B., Yang, X., Wang, B., Cui, W.: Efficiently mining high quality phrases from texts. In: AAAI, pp. 3474–3481 (2017)
4. Li, B., Yang, X., Zhou, R., Wang, B., Liu, C., Zhang, Y.: An efficient method for high quality and cohesive topical phrase mining. TKDE (2018, to appear)
5. Mohri, M.: String matching with automata. Nord. J. Comput. **4**(2), 217–231 (1997)
6. Navarro, C.: NR-grep: a fast and flexible pattern matching tool. Softw. Pract. Exp. (SPE) **31**, 1265–1312 (2001)
7. Navarro, C., Raffinot, M.: Flexible Pattern Matching in Strings: Practical Online Search Algorithms for Texts and Biological Sequences. Cambridge University Press, Reading (1979)
8. Thomphson, K.: Regular expression search algorithm. Commun. ACM **11**, 419–422 (1968)
9. Watson, B.W.: A new regular grammar pattern matching algorithm. In: Diaz, J., Serna, M. (eds.) ESA 1996. LNCS, vol. 1136, pp. 364–377. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-61680-2_68
10. Yang, X., Liu, H., Wang, B.: ALAE: accelerating local alignment with affine gap exactly in biosequence databases. PVLDB **5**(11), 1507–1518 (2012)
11. Yang, X., Qiu, T., Wang, B., Zheng, B., Wang, Y., Li, C.: Negative factor: improving regular-expression matching in strings. ACM Trans. Database Syst. (TODS) **40**(4), 25 (2016)
12. Yang, X., Wang, B., Qiu, T., Wang, Y., Li, C.: Improving regular-expression matching on strings using negative factors. In: SIGMOD, pp. 361–372, June 2013
13. Yang, X., Wang, B., Yang, K., Liu, C., Zheng, B.: A novel representation and compression for queries on trajectories in road networks. TKDE **30**(4), 613–629 (2018)
14. Yang, X., Wang, Y., Wang, B., Wang, W.: Local filtering: improving the performance of approximate queries on string collections. In: SIGMOD, pp. 377–392. ACM (2015)

The 2nd International Workshop on Graph Data Management and Analysis (GDMA 2018)

Extracting Schemas from Large Graphs with Utility Function and Parallelization

Yoshiki Sekine and Nobutaka Suzuki(✉)

University of Tsukuba, 1-2 Kasuga, Tsukuba, Ibaraki 305-8550, Japan
ysekine@klis.tsukuba.ac.jp, nsuzuki@slis.tsukuba.ac.jp

Abstract. Unlike relational databases and XML documents, most of graphs are not given their own schemas. If we can extract a schema from a graph efficiently, we can take advantage of the extracted schema for query optimization, structure browsing, and so on. In this paper, we consider extracting schemas from large graphs by using *utility function*. Although reasonable schemas can be extracted by the utility function, the major problem of the utility function is its computation cost. In this paper, we propose a schema extraction algorithm based on (a) a novel utility function called local utility function and (b) parallelization. Experimental results show that our algorithm can extract schemas from graphs more efficiently without losing quality of schemas.

1 Introduction

Recently, various kinds of graphs are widely used, e.g., SNS graph, citation graph, RDF graph, and so on. Unlike relational databases and XML documents, most of graphs are not given their own schemas. Therefore, in many cases we cannot make use of schemas to manage graphs effectively. Since a schema of a graph is a concise representation of the graph, if we can extract a schema from a graph efficiently, we can take advantage of the extracted schema for query optimization, structure browsing, query formulation, and so on. A schema of a graph is also used to calculate ObjectRank scores [1].

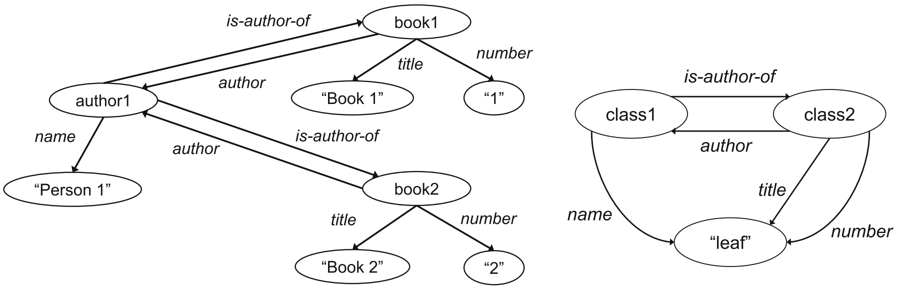
In this paper, we consider extracting schemas from large graphs by using *utility function*. Here, utility function is firstly proposed in the COBWEB system [2], and the utility function can successfully be applied to extract schemas from graphs [13]. In short, the utility function is used to select, for each node v in a given graph, which of the classes in the current schema is the "best" class to which v belongs. Although reasonable schemas can be extracted by the utility function, the major problem of the utility function is its computation cost; in order to calculate the utility function, we need to explore all the classes and all the labels in an extracted schema. Note that, if the input graph large and contains variety kinds of nodes, much more classes tend to be extracted due to the "richness" of the input graph. For such graphs, efficient schema extraction using the utility function becomes highly difficult.

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 125–140, 2018.
https://doi.org/10.1007/978-3-319-91455-8_13

To address the problem, we propose an algorithm for extracting schemas from large graphs, using a novel utility function called *local utility function*. Our local utility function can be calculated by using only the "local" information of a given node and a class, not requiring the entire classes nor the labels of the current schema. Thus, by using our local utility function, schemas can be extracted more efficiently. Another feature of our algorithm is that our algorithm extracts classes *in parallel*, aiming for further efficient schema extraction. Here, if classes were naively extracted in parallel, then extracted classes would conflict with each other and the quality of extracted schemas would decrease significantly. To avoid this problem, we incorporate an efficient "conflict resolution method" to our algorithm so that the quality of schemas are retained. In addition to the above two features, our algorithm is designed as an external memory algorithm, which is partly based on our previous work [11]. This makes it possible for our algorithm to deal with large graphs that do not fit in main memory. Experimental results show that our algorithm can extract schemas from graphs much more efficiently, and that the quality of schemas extracted by our algorithm is almost equal to schemas extracted by using the original utility function.

Related Work

Wang et al. proposed a schema extraction algorithm for graphs by using the original utility function [13]. The algorithm extracts reasonable schemas from graphs but requires too much computation cost for large graphs. DataGuide [4] extracts a schema by grouping nodes reachable from the root via the same path labels into the same class. The algorithm works efficiently for tree-like data but is highly inefficient for graphs containing cycles. ApproximateDataguite [5] is the approximate version of DataGuide, which proposed three methods to merge similar nodes: Object Matching, Suffix Matching, and Path-Cycle Matching. Suffix Matching merges nodes having the same incoming label regardless of their outgoing labels. Thus nodes having completely different outgoing edges may be merged into the same class. The other two methods are still inefficient and too many classes are extracted for large graphs. Nestorov et al. proposed an algorithm for extracting approximate typings by using a clustering approximation method [9]. However, the algorithm requires double-quadratic time and is hardly applicable to large graphs. Navlakha et al. proposed a graph summarization algorithm [8]. The algorithm is designed for unlabeled undirected graphs, while our algorithm is designed for labeled directed graphs. All the algorithms mentioned above are in-memory algorithms and cannot handle graphs that do not fit in main memory, and parallelization is not considered either. Luo et al. proposed an external memory algorithm for computing k-bisimulation of graph [7]. K-bisimulation is suitable for constructing exact indexes of graphs rather than extracting "approximate" schemas. Our previous work [11] proposed an external memory algorithm for extracting schemas from graphs. The algorithm is based on the original utility function and do not consider parallelization, and thus takes too much computation cost for graphs containing various kinds of nodes.



(a) Example of graph

(b) Example of schema

Fig. 1. Example of graph and schema

For XML documents, a number of schema extraction algorithms were proposed, e.g., [3,6,12]. However, these algorithm are designed for XML trees and cannot handle graphs containing cycles.

2 Preliminaries

Let L be a set of labels. A *labeled directed graph* (*graph* for short) is denoted $G = (V, E)$, where V is a set of *nodes* and $E \subseteq V \times L \times V$ is a set of *labeled directed edges* (*edges* for short). Let $e \in E$ be an edge labeled by $l \in L$ from a node $v \in V$ to a node $u \in V$. Then e is denoted (v, l, u), v is called *source*, u is called *target*, and we say that e is an *outgoing edge* of v. By $L(v)$, we mean the set of outgoing edge labels of v, that is, $L(v) = \{l \in L \mid (v, l, u) \in E\}$.

A schema is also represented as a graph. Formally, a *schema* is denoted $S = (C, E_s)$, where C is a set of nodes called *classes* and E_s is a set of edges between classes. For a node v in an (instance) graph, by $class(v)$ we mean the class that v belongs to. Any node in a graph is mapped to a class in a schema. We assume that every text node belongs to a single class denoted "leaf".

Example 1. Figure 1a and b illustrate a graph and its schema S, respectively. Book1 and book2 in Fig. 1a belong to class2 in Fig. 1b. Similarly, author1 belongs to class1 and the other nodes belong to "leaf". We have $S = (C, E_s)$, where $C = \{$class1, class2, "leaf"$\}$ and $E_s = \{$(class1, is-author-of, class2), (class2, author, class1), (class1, name, "leaf"), (class2, title, "leaf"), (class2, number, "leaf")$\}$.

In the following, we assume that a graph is stored in a graph file like N-Triples format. Each line of a graph file corresponds to an edge, namely, a line consists of $(source, label, target)$. Figure 2 shows a graph file that stores the graph of Fig. 1a. We also assume that a schema is stored in two files denoted schema_classes and schema_edges. The former stores pairs of a node and its class, namely, each line is of the form $(node, class)$. The latter stores edges between classes, in which

source	label	target
author1	name	"Person1"
author1	is-author-of	book1
author1	is-author-of	book2
book1	author	author1
book2	author	author1
book1	title	"Book1"
book2	title	"Book2"
book1	number	"1"
book2	number	"2"

Fig. 2. Graph file file 1 of the graph in Fig. 1a

node	class
author1	class1
book1	class2
book2	class2

(a) schema_classes

source class	label	target class
class1	is-author-of	class2
class1	name	"leaf"
class2	author	class1
class2	title	"leaf"
class2	number	"leaf"

(b) schema_edges

Fig. 3. Example of schema files

each line is of the form of (*source class*, *label*, *target class*). Figure 3 shows a pair of two schema files representing the schema of Fig. 1b.

3 Utility Function

In this section, we present two utility functions: original utility function [2,13] and our local utility function. The utility functions are used to select, for each node v, which class is the "best" for v by measuring the structural regularity of resulting classes. Let v be a node, c be a class, and C be a set of classes. We write $v \in c$ if v belongs to c. By $L(c)$, we mean the set of outgoing edge labels of the nodes in c, that is,

$$L(c) = \bigcup_{v \in c} L(v).$$

By $|c|$ we mean the number of nodes in c and by $c(l)$ we mean the set of nodes v in c such that v has an outgoing edge labeled by l. Then $P(l|c)$ is defined as the ratio of $|c(l)|$ to $|c|$, that is,

$$P(l|c) = \frac{|c(l)|}{|c|}.$$

By $C(l)$ we mean the set of nodes v in C such that v has an outgoing edge labeled by l, that is,

$$C(l) = \bigcup_{c \in C} c(l).$$

$P(c|l)$ is defined as the ratio of $|c(l)|$ to $C(l)$, that is,

$$P(c|l) = \frac{|c(l)|}{|C(l)|}.$$

$T(l, c)$, representing the strength of the relationship between label l and c, is defined as follows.

$$T(l, c) = P(l|c) \cdot P(c|l).$$

Then $E(c)$ is defined as the mean of $T(l, c)$ over $l \in L(c)$, that is,

$$E(c) = \frac{1}{|L(c)|} \sum_{l \in L(c)} T(l, c). \tag{1}$$

Now let us define the two utility functions. First, the *original utility function* is defined as the mean of $E(c)$ over $c \in C$, that is,

$$U(C) = \frac{1}{|C|} \sum_{c \in C} E(c). \tag{2}$$

By (2), to calculate $U(C)$ we need to calculate $E(c)$ for *every* $c \in C$, which also implies that $T(l, c)$ must be calculated for all label l by (1) in total. Here, a graph with variety kinds of nodes contains a large number of labels and brings a large number of extracted classes. For such graphs, the original utility function requires too much computation cost.

To address this problem, we propose another utility function, called *local utility function*. This is defined as the product of the dice coefficient and the mean of $P(c|l)$ over $l \in L(v)$, that is,

$$U_l(C, v, c) = Dice(L(v), L(c))^{\alpha} \frac{1}{|L(v)|} \sum_{l \in L(v)} P(c|l), \tag{3}$$

where

$$Dice(L(v), L(c)) = \frac{2|L(v) \cap L(c)|}{|L(v)||L(c)|},$$

and $\alpha > 0$ is a parameter. Here, α is adopted to the local utility function so that users can control which of the dice coefficient and the rest subexpression is emphasized. Since $Dice(L(v), L(c)) \leq 1$, as α gets larger, $Dice(L(v), L(c))^{\alpha}$ gets smaller. This implies that, as a larger value is set to α, a larger number of classes are extracted since each class contains only nodes similar to each other.

Note that in (3), both the dice coefficient and the mean of $P(c|l)$ over $l \in L(v)$ can be calculated by using only the "local" information of v and c, and no calculation over "every class" nor "every label" in a schema is required. Thus the local utility function can be computed efficiently, especially for graphs containing variety kinds of nodes. In spite of less computation cost, as shown in Sect. 5, the quality of schemas extracted by our algorithm is almost equal to that of schemas extracted by using the original utility function.

4 The Algorithm

In this section, we present our algorithm for extracting schemas from graphs. First, we give the outline of our algorithm, then present the details of our algorithm.

4.1 Outline of the Algorithm

Algorithm 1 shows the outline of our algorithm. Lines 3 to 12 extracts classes and lines 14 to 16 extracts edges between classes. In the class extraction part, for each node v in a given graph, the local utility function for v is calculated for each of the following classes:

- The existing classes extracted so far (lines 4 to 6), and
- a new class c_v having the same outgoing edges as v (lines 7 and 8).

Among the above classes, class c_{best} bringing the highest utility value is chosen as the "best" class for v and c_{best} is added to a current schema as an extracted class (lines 9 to 11). In the edge extraction part, for each edge $(v, l, v') \in E$, an edge $(class(v), l, class(v'))$ is extracted.

source	label	target
author1	is-author-of	book1
author1	is-author-of	book2
author1	name	"Person1"
book1	author	author1
book1	number	"1"
book1	title	"Book1"
book2	author	author1
book2	number	"2"
book2	title	"Book2"

Fig. 4. file 1'

4.2 Details of the Algorithm

We now present the details of our schema extraction algorithm. The algorithm is designed as an external memory algorithm to handle large graphs that cannot fit in main memory, in which the class extraction is parallelized. The algorithms consists of the following three parts.

Algorithm 1. Outline of the Algorithm

```
Input: graph $G = (V, E)$
Output: schema $(C, E_s)$ of $G$
$C \leftarrow \emptyset$, $E_s \leftarrow \emptyset$
// class extraction
for each node $v \in V$ do
    for each class $c \in C$ do
        Calculate $U_l(C, v, c)$
    end for
    Let $c_v$ be a new class having the same set of outgoing edge labels as $v$
    Calculate $U_l(C, v, c_v)$
    Let $c_{best}$ be the class such that the utility value is the highest among $C \cup \{c_v\}$
    $class(n) \leftarrow c_{best}$
    $C \leftarrow C \cup \{c_{best}\}$
end for
// edge extraction
for each edge $(v, l, v') \in E$ do
    $E_s \leftarrow E_s \cup \{(class(v), l, class(v'))\}$
end for
return $(C, E_s)$
```

1. Preprocessing: sort the input graph file by source node
2. Class extraction: extract classes by grouping similar nodes
3. Edge extraction: extract edges between classes

In the following, we give the details of the three parts. Our algorithm is designed as an external memory algorithm in which schema extraction is achieved by using only sequential read and external sort, in order to minimize random accesses to files on a disk.

Preprocessing. Let file 1 be the input graph file (Fig. 2). Each line represents an edge, namely, a line consists of the source, the label, and the target of an edge. We sort file 1 by source node and let file 1′ be the resulting file (Fig. 4). Since file 1′ is sorted, edges having the same source node appear consecutively in file 1′. Therefore, we can obtain the outgoing edges of each node by reading file 1′ sequentially.

Class Extraction. The outline of our class extraction part is as follows. By reading file 1′ sequentially, we obtain the set of outgoing edges of consecutive k nodes. We extract classes of the k nodes based on the local utility function in k parallel processes. After the completion of the parallel processes, some nodes are assigned to existing classes $c \in C$, and the other nodes v are (temporally) assigned to a new class c_v, where c_v is a new class having the same set of outgoing edges as v. Note that, such new classes are created *in parallel*, and thus classes having very similar or the same structure may be created. But such similar

Algorithm 2. Class Extraction

```
Input: file 1', positive integer $k$
Output: schema_classes
 1: Create an empty file schema_classes
 2: $C \leftarrow \emptyset$
 3: while file 1' does not reach EOF do
 4:     $V_{tmp} \leftarrow \emptyset$
 5:     Read the edges of $k$ consecutive nodes by reading file 1' sequentially. Let
        $v_1, v_2, \cdots, v_k$ be the $k$ nodes.
 6:     for each $i = 1, 2, \cdots, k$ do
 7:         $L(v_i) \leftarrow$ the set of labels of the outgoing edges of $v_i$
 8:         $V_{tmp} \leftarrow V_{tmp} \cup \{v_i\}$
 9:     end for
10:     $R \leftarrow \emptyset$
11:     Parallel for each $v_i \in V_{tmp}$
12:         $class(v_i) \leftarrow$ ClassDetermination$(C, v_i, L(v_i))$
13:         $R \leftarrow R \cup \{(v_i, class(v_i))\}$
14:     End Parallel
15:     $V_{one} = \{v \mid (v, class(v)) \in R, class(v) = c_v\}$
16:     if $|V_{one}| > 1$ then                                  ▷ conflict occurs
17:         $(R', C) \leftarrow$ ConflictResolution$(C, R, V_{one})$
18:     else
19:         $R' \leftarrow \{(v_i, class(v_i)) \mid 1 \leq i \leq k\}$
20:         $C \leftarrow C \cup \{class(v_1), class(v_2), \ldots, class(v_k)\}$
21:     end if
22:     Add each pair $(v_i, class(v_i)) \in R'$ to schema_classes
23: end while
```

classes should be merged into the same class. We call this recalculation process *conflict resolution*. By doing that, we finalize the classes of k nodes. This process is repeated until file 1' reaches the end of file.

Let us next present the details of the class extraction algorithm (Algorithm 2). The algorithm reads consecutive k nodes by reading file 1' sequentially (line 5). Let $v_1, v_2, \cdots, v_k$ be the k nodes. In lines 6 to 9, we obtain $L(v_i)$ for every $1 \leq i \leq k$, which is the set of labels of the outgoing edges of v_i. Lines 11 to 14 determines the classes of $v_1, v_2, \cdots, v_k$ in parallel. ClassDetermination in line 12 is an algorithm for extracting the class of v_i based on the local utility function. In ClassDetermination (Algorithm 3), for a given node v_i, the class that brings the most high utility value is chosen among the following classes.

C1: A new class c_{v_i} having the same outgoing edges as v_i, and
C2: the existing classes in C.

Note that C is not updated in the parallel block of Algorithm 2. After the parallel block is completed, we obtain a set R of pairs $(v_i, class(v_i))$ $(1 \leq i \leq k)$ by line 13. Then the set V_{one} of nodes v belonging to c_v (i.e., the nodes in the case C1) is collected (line 15). Note that V_{one} is the set of nodes whose class may conflict. If $|V_{one}| > 1$, then ConflictResolution (Algorithm 4) is called to

Algorithm 3. Class Determination

```
procedure ClassDetermination($C, v, L(v)$)
    for each class $c_i \in C$ do
        Calculate $U_l(C, v, c_i)$
    end for
    Let $c_v$ be the new class having the same set of outgoing edges as $v$
    Calculate $U_l(C, v, c_v)$
    Let $c_{best}$ be the class such that the value of $U_l$ is the highest among $C \cup \{c_{best}\}$
    return $c_{best}$
end procedure
```

Algorithm 4. Conflict Resolution

```
 1: procedure ConflictResolution($C, R, V_{one}$)
 2:     $C_{tmp} \leftarrow \emptyset$
 3:     $R' = \{(v, class(v)) \in R \mid v \notin V_{one}\}$        ▷ conflict resolution result
 4:     Remove the first node of $V_{one}$. Let $v$ be the first node.
 5:     $R' \leftarrow R' \cup \{(v, class(v))\}$
 6:     $C_{tmp} \leftarrow C_{tmp} \cup \{class(v)\}$
 7:     $C \leftarrow C \cup \{class(v) \mid (v, class(v)) \in R'\}$
 8:     for each $v \in V_{one}$ do
 9:         $class(v) \leftarrow$ ClassDetermination2($C, C_{tmp}, v, L(v)$)
10:         $R' \leftarrow R' \cup \{(v, class(v))\}$
11:         $C_{tmp} \leftarrow C_{tmp} \cup \{class(v)\}$
12:         $C \leftarrow C \cup \{class(v)\}$
13:     end for
14:     return $(R', C)$
15: end procedure
```

resolve conflicts (line 17, explained later). Otherwise, the conflict resolution is skipped and the obtained classes are set to R' and added to C (lines 19 and 20). After the conflict resolution, v_i and its class $class(v_i)$ $(1 \leq i \leq k)$ is written into the output schema file schema_classes (line 22). Repeating this process until the input file reaches EOF, we obtain the classes of all nodes.

Let us next explain ConflictResolution. C_{tmp} is the set of classes whose conflict resolution is completed, which is initially empty (line 1). R' denotes the set of pairs of a node v and its class $class(v)$ such that the class is not conflicted or its conflict resolution is completed. Initially, R' is the set of pairs of a node and its class such that the class is not in V_{one} (line 3). Then a node v is picked up from V_{one} (line 4). We regard the conflict of v as "resolved", and update R' and C_{tmp} accordingly (lines 5 and 6). Then the classes in R' are added to C, which is used for the subsequent calculation of the local utility function (line 7). Thus C consists of the classes extracted so far in Algorithm 2 and the classes of which conflict resolution is completed. Then for the nodes in V_{one}, conflict is resolved in lines 8 to 13, as follows. For each node $v \in V_{one}$, we recalculate the local utility function. To do this, we call ClassDetermination2 (Algorithm 5) instead of ClassDetermination to avoid duplicate calculation of the local

Algorithm 5. Class Determination in Conflict Resolution

```
procedure ClassDetermination2($C, C', v, L(v)$)
    for each class $c_i \in C'$ do
        Calculate $U_l(C, v, c_i)$
    end for
    Let $c_v$ be the new class having the same set of outgoing edges as $v$.
    Calculate $U_l(C, v, c_v)$
    Let $c_{best}$ be the class such that the value of $U_l$ is the highest among $C' \cup \{c_v\}$.
    return $c_{best}$
end procedure
```

utility function. That is, for a given node $v \in V_{one}$, the local utility function is calculated for the classes in $C_{tmp} \cup \{c_v\}$ instead of the classes in $C \cup \{c_v\}$. Specifically, ClassDetermination2 extracts the class of $v \in V_{one}$ as follows. We calculate the utility function in the following classes and choose the class for which the maximum utility value is obtained.

- The "conflict-resolved" classes in C_{tmp}, and
- a new class c_v having the same set of outgoing edge labels as v.

Note that we recalculate the utility function for c_v from the latest schema because the schema is updated and the resulting utility value may be different. Each time the class of v is determined, we update R', C_{tmp}, and C (lines 10 to 12 of Algorithm 4). When the class of every node in V_{one} is determined, (R', C) is returned to Algorithm 2.

Edge Extraction. The edge extraction part is similar to our previous work [11], and thus we explain this part briefly by an example. In short, schema_edges is obtained form the sorted input graph file by replacing each node in the graph file by its class. Here, consider file 1' in Fig. 4. First, by reading file 1' and schema_classes (Fig. 3a) sequentially and simultaneously, we replace each source node by its class (tmp_file1, shown in Fig. 5a). Then swap the source and the target, and externally sort the swapped file by the target (tmp_file2, shown in Fig. 5b). By reading tmp_file2 and schema_classes sequentially and simultaneously, replace each target node by its class (tmp_file3 in Fig. 5c). Finally, by removing duplicated edges of tmp_file3 and swapping the target and the source, we obtain schema_edges (Fig. 3b).

4.3 CPU and I/O Costs

Let us give the CPU and I/O costs of our algorithm. To compare the CPU costs in terms of the two utility functions, we present the CPU costs of the algorithm with the two utility functions by using the number of extracted classes.

Let L_v be the maximum number of labels owned by a node, that is, $L_v = \max_{v \in V} |L(v)|$. Let L_c be the maximum number of labels owned by a class, that

source	label	target
class1	is-author-of	book1
class1	is-author-of	book2
class1	name	“leaf”
class2	author	author1
class2	number	“leaf”
class2	title	“leaf”
class2	author	author1
class2	number	“leaf”
class2	title	“leaf”

(a) tmp_file1

target	label	source
“leaf”	name	class1
“leaf”	number	class2
“leaf”	number	class2
“leaf”	title	class2
“leaf”	title	class2
author1	author	class2
author1	author	class2
book1	is-author-of	class1
book2	is-author-of	class1

(b) tmp_file2

target	label	source
“leaf”	name	class1
“leaf”	number	class2
“leaf”	number	class2
“leaf”	title	class2
“leaf”	title	class2
class1	author	class2
class1	author	class2
class2	is-author-of	class1
class2	is-author-of	class1

(c) tmp_file3

Fig. 5. Intermediate files created in edge extraction

is, $L_c = \max_{c \in C} |\{l \in L(v) \mid v \in c\}|$. Moreover, by $|C|$ we mean the number of classes in C. For each node $v \in V$, the algorithm (with the local utility function) does the following.

- For each class $c \in C$, calculate the local utility function $U_l(C, v, c)$, which requires $O(L_v + L_c)$.
- After the class of v is determined, update $c(l)$ and $C(l)$. This requires $O(L_v)$.

Thus, the CPU cost of the algorithm is

$$O(|V|(|C|(L_v + L_c) + L_v)) = O(|V| \cdot |C| \cdot (L_v + L_c)).$$

On the other hand, to calculate the original utility function $U(C)$, we need $O(L_c \cdot |C|)$. Thus the CPU cost of the algorithm with the original utility function is

$$O(|V| \cdot |C| \cdot (L_c \cdot |C| + L_v)).$$

Thus, the algorithm with the local utility function runs in time linear to $|C|$, while with the original utility function the algorithm runs in time square of $|C|$. This implies that the cost of the algorithm with the original utility function becomes much higher if an input graph is large and consists of variety kinds of nodes, since such a graph tends to bring a large number of classes.

Finally, the I/O cost of the algorithm (with the original/local utility function) is

$$O\left(|V|/B + sort(|E|)\right),$$

where B is the block transfer size between external memory and main memory, and $sort(|E|)$ is the I/O cost of external merge sort (details are omitted because of space limitation).

5 Evaluation Experiment

In this section, we present experimental results on our algorithm. The algorithm was implemented in Ruby 2.4.2, and the parallelized class extraction was implemented by Ruby Gem parallel (version 1.12.0)[1]. All the evaluation experiments

[1] https://github.com/grosser/parallel.

Table 1. Dataset of the experiment

(a) SP^{2}Bench Graphs

$\|E\|$	$\|V^*\|$	$\|L\|$	size (GB)
1,000,009	187,066	24	0.10
10,000,457	1,730,250	26	1.04
100,000,380	17,823,525	26	10.35

(b) DBPedia Graphs

$\|E\|$	$\|V^*\|$	$\|L\|$	size (GB)
15,373,833	313,036	14,130	2.72
76,868,920	1,177,165	22,147	12.80
153,737,783	1,457,983	23,343	25.11

were executed on a machine with Intel Xeon E5-2623 v3 3.0 GHz CPU, 16 GB RAM, 2 TB SATA HDD, and Linux CentOS 7 64bit. We used GNU sort command in order to sort files externally in the preprocessing and the edge extraction, and we limited the maximum memory usage of the sort command to 1 GB by using option "-S".

In the experiments, we used the following two contrasting datasets.

SP^{2}Bench [10] *(SP2B, for short)* is a benchmark tool generating RDF (N-Triples) files based on DBLP. We generated three graphs of different sizes in Table 1a, where V^* denotes the non-leaf nodes (i.e., nodes for which classes are extracted) and L is the set of edge labels. The total number of unique RDF types in SP2B graph is 12.

DBPedia project extracts structured data from Wikipedia. We downloaded and used three benchmark dataset graphs[2] shown in Table 1b. In contrast to SP2B, DBPedia has a large number of unique edge labels. Moreover, DBPedia consists of much more variety kinds of nodes than SP2B; the total number of unique RDF types in the graph with $|E| = 15,373,833$ is 54,736, which is much larger than SP2B.

5.1 Execution Time and Memory Usage

Let us first present the execution time and the memory usage of our algorithm. We first give the results on the class extraction part. Then we give the results including the preprocessing and the edge extraction part briefly.

Execution Time of Class Extraction. Firstly, we present the execution time of the class extraction part. Tables 2 and 3 show the results. Table 2 shows the execution time of the class extraction for SP2B graphs. As shown in the table, for both utility functions the execution time is almost linear to the size of input graph. Moreover, the local utility function can be calculated more efficiently than the original utility function.

Table 3 shows the execution time of the class extraction part for DBPedia graphs. Compared to SP2B, DBPedia graphs contain a variety kinds of RDF

[2] http://benchmark.dbpedia.org/.

Table 2. Execution time (sec) of the class extraction for SP2B graphs

Utility function	$\lvert E\rvert$		
	1,000,009	10,000,457	100,000,380
Original ($k = 1$)	13.26	111.43	1065.99
Local ($k = 1$)	6.70	64.23	651.07

Table 3. Execution time (sec) of the class extraction for DBPedia graphs

Utility function	$\lvert E\rvert$		
	15,373,833	76,868,920	153,737,783
Original ($k = 1$)	-	-	-
Local ($k = 1$)	11,242.80	110,555.53	150,143.69
Local ($k = 4$)	4,803.85	42,071.79	58,026.58

types, and thus much more classes were extracted. Due to this, the class extraction part using the original utility function took longer than 24 h even for the smallest graph (these are treated as "do not finished" cases). On the other hand, as shown in the table, the class extraction part using the local utility function run much faster. These results imply that the combination of the local utility function and the parallelization works effectively.

Total Execution Time Including Preprocessing and Edge Extraction. We next give the total execution time including the preprocessing and the edge extraction part. Tables 4 and 5 show the details of the execution time. As shown in the tables, the execution time of the class extraction part is dominant for DBPedia. This implies that class extraction part is the most heavy process for extracting schemas from large and complex graphs, and thus the class extraction part is the most important part for improving the efficiency of our algorithm.

5.2 Memory Usage of the Algorithm

Let us give the memory usage of our algorithm. Table 6 shows the memory usage of the class extraction part for the largest graphs of SP2B and DBPedia.

Table 4. Total execution time (sec) of our algorithm (SP2B)

$\lvert E\rvert$	Preprocessing	Class extraction ($k = 1$)	Edge extraction	Total
1,000,009	8.85	6.70	5.46	21.01
10,000,457	84.16	64.23	56.97	205.36
100,000,380	856.52	651.07	577.51	2,085.10

Table 5. Total execution time (sec) of our algorithm (DBPedia)

$\vert E\vert$	Preprocessing	Class extraction ($k = 4$)	Edge extraction	Total
15,373,833	168.66	4,803.85	104.35	5,076.86
76,868,920	1,273.80	42,071.79	572.26	43,917.85
153,737,783	2,914.30	58,026.58	1559.73	62,500.61

Table 6. Memory usage of the class extraction part

Dataset	$k = 1$	$k = 4$
SP2B ($\vert E\vert = 100,000,380$)	11.1 MB	7.5 MB
DBPedia ($\vert E\vert = 153,737,783$)	116.6 MB	89.6 MB

This shows that the class extraction part requires only small amount of memory w.r.t. data size. The memory usage of preprocessing and the edge extraction part is as follows. For any graphs, we observed that the memory usage of the preprocessing and the edge extraction part is about 1.1 GB, of which 1 GB is used for external sorting since we limited the maximum memory usage of the sort command to 1 GB. Consequently, the memory usage of the class extraction part and the edge extraction part is fairly limited and the total memory usage of our algorithm mostly depends on external sorting.

5.3 Quality of Extracted Schema

Let us present the quality of schema extracted by our algorithm. To measure the quality of extracted schemas, we introduce two scores *Score1* and *Score2*. Both SP2B and DBPedia are RDF data and thus each node has RDF type(s). We use these RDF types as "correct answers" and calculate the scores by comparing the RDF type(s) and the extracted class of each node. In the following definition, class "leaf" is omitted.

Score1 is defined so that the score becomes larger as each extracted class contains smaller numbers of different RDF types. By $types(v)$ we mean the set of RDF types assigned to node v. The set of nodes in class c having RDF type t is denoted $nodes(t, c)$. Then $Score1$ is defined as follows.

$$Score1 = \frac{1}{|V^*|} \sum_{v \in V^*} \frac{1}{|types(v)|} \sum_{t \in types(v)} \frac{|nodes(t, class(v))|}{|class(v)|}.$$

Table 7. Class extraction scores for the SP2B graph ($|E| = 10,000,457$)

Utility function	Score 1	Score 2	Mean
Original	97.02	95.32	96.17
Local ($\alpha = 1$)	72.53	100.00	86.26
Local ($\alpha = 10$)	99.45	88.83	94.14

Table 8. Class extraction scores for DBPedia graphs

(a) Graph with $|E| = 50,000$

utility function	Score 1	Score 2	Mean
Original	43.07	81.69	62.38
Local ($\alpha = 1$)	67.30	73.87	70.58
Local ($\alpha = 10$)	89.74	27.77	58.76

(b) Graph with $|E| = 15,373,833$

utility function	Score 1	Score 2	Mean
Original	-	-	-
Local ($\alpha = 1$)	70.06	76.97	73.51
Local $\alpha = 10$)	85.12	3.77	44.44

Score2 is defined so that the score becomes larger as each RDF type is distributed to smaller numbers of different classes. Let $total(t)$ be the total number of nodes having RDF type t, and let $max(t) = \max_c nodes(t, c)$. Then $Score2$ is the mean of ratio of the two, that is,

$$Score2 = \frac{1}{|T|} \sum_{t \in T} \frac{max(t)}{total(t)}.$$

Tables 7 and 8a, b show the results. Table 7 shows the class extraction scores for the SP2B graph with $|E| = 10,000,457$. The result shows that both of the utility functions achieved high scores. The reason why such high scores are obtained is that in SP2B graphs $|L|$ is small and nodes having the same RDF type have a similar set of outgoing edge labels.

For DBPedia, since the algorithm using the original utility function took too much execution time even for the smallest graph ($|E| = 15,373,833$), we created a tiny graph with $|E| = 50,000$ by deleting edges from the smallest graph and calculated the scores for the tiny graph. Table 8a shows the class extraction scores for the tiny DBPedia graph. The maximum mean of score 70.58 is obtained at $\alpha = 1$, which is higher than the value 62.38 obtained by the algorithm using the original utility function. Table 8b shows the class extraction scores for a larger DBPedia graph with $|E| = 15,373,833$. The score with $\alpha = 1$ is better than that of tiny graph. Thus, regardless of data size, our algorithm can extract schemas with reasonable quality.

6 Conclusion

In this paper, we proposed an external memory algorithm for extracting a schema from a graph. Since class extraction is the most dominant part of schema extraction, we devised a new utility function called the local utility function and parallelized our class extraction part. Experimental results showed that the combination of the local utility function and the parallelization is effective to extracting schemas from large complex graphs efficiently.

However, we have a lot to do as future works. First, we need to compare our algorithm and other schema extraction algorithms experimentally. Second, we need to conduct experiments using other types of graphs, e.g., SNS graphs. Third, languages other than Ruby should be taken into consideration for implementing our algorithm.

Acknowledgments. This work was supported by JSPS KAKENHI Grant Number JP17K00150.

References

1. Balmin, A., Hristidis, V., Papakonstantinou, Y.: ObjectRank: authority-based keyword search in databases. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases, pp. 564–575 (2004)
2. Fisher, D.: Knowledge acquisition via incremental conceptual clustering. In: Shavlik, J., Dietterich, T. (eds.) Readings in Machine Learning. Morgan Kaufmann Publishers (1990)
3. Garofalakis, M.N., Gionis, A., Rastogi, R., Seshadri, S., Shim, K.: XTRACT: a system for extracting document type descriptors from XML documents. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 165–176 (2000)
4. Goldman, R., Widom, J.: DataGuides: enabling query formulation and optimization in semistructured databases. In: Proceedings of 23rd International Conference on Very Large Data Bases (VLDB 1997), pp. 436–445 (1997)
5. Goldman, R., Widom, J.: Approximate DataGuides. In: Proceedings of the Workshop on Query Processing for Semistructured Data and Non-standard Data Formats, vol. 97, pp. 436–445 (1999)
6. Hegewald, J., Naumann, F., Weis, M.: XStruct: efficient schema extraction from multiple and large XML documents. In: Proceedings of the 22nd International Conference on Data Engineering Workshops, ICDE 2006, p. 81 (2006)
7. Luo, Y., Fletcher, G.H., Hidders, J., Wu, Y., De Bra, P.: External memory k-bisimulation reduction of big graphs. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM 2013), pp. 919–928 (2013)
8. Navlakha, S., Rastogi, R., Shrivastava, N.: Graph summarization with bounded error. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 419–432 (2008)
9. Nestorov, S., Abiteboul, S., Motwani, R.: Extracting schema from semistructured data. In: Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 295–306 (1998)
10. Schmidt, M., Hornung, T., Lausen, G., Pinkel, C.: SP^{2}Bench: a SPARQL performance benchmark. In: Proceedings of the IEEE 25th International Conference on Data Engineering (ICDE 2009), pp. 222–233. IEEE (2009)
11. Sekine, Y., Suzuki, N.: An algorithm for extracting schemas from external memory graphs. In: Proceedings of the First Workshop on Big Network Analytics (in Conjunction with CIKM 2016) (2016)
12. Shanmugasundaram, J., Tufte, K., Zhang, C., He, G., DeWitt, D.J., Naughton, J.F.: Relational databases for querying XML documents: limitations and opportunities. In: Proceedings of 25th International Conference on Very Large Data Bases, pp. 302–314 (1999)
13. Wang, Q.Y., Yu, J.X., Wong, K.-F.: Approximate graph schema extraction for semi-structured data. In: Zaniolo, C., Lockemann, P.C., Scholl, M.H., Grust, T. (eds.) EDBT 2000. LNCS, vol. 1777, pp. 302–316. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-46439-5_21

FedQL: A Framework for Federated Queries Processing on RDF Stream and Relational Data

Guozheng Rao[1,3], Bo Zhao[1,3], Xiaowang Zhang[1,3](✉), and Zhiyong Feng[2,3]

[1] School of Computer Science and Technology,
Tianjin University, Tianjin 300350, China
xiaowangzhang@tju.edu.cn

[2] School of Computer Software, Tianjin University,
Tianjin 300350, China

[3] Tianjin Key Laboratory of Cognitive Computing and Application,
Tianjin 300350, China

Abstract. In this paper, we present a framework (FedQL) for processing RDF stream and relational data in a federal way. Firstly, we introduce a formalization of our federated query language by conjunction of continuous queries and SQL queries. Secondly, we present a white-box-based framework to separate query processing from query executing. The framework mainly includes three modules, namely, Query processor, Data transformer, and SPARQL query execution. Finally, we implement FedQL built on C-SPARQL and MySQL by employing three centralized SPARQL engines (e.g. Jena, RDF-3X, and gStore) and one distributed SPARQL engine (e.g. TriAD) in an adaptive way and evaluate FedQL on a real-world dataset. The experimental results show that FedQL is efficient and effective in processing RDF stream and relational data in a federal way.

Keywords: Federated query · RDF stream · Relational data
SPARQL · SQL

1 Introduction

RDF stream, as a dynamic type of dataset, can model real-time and continuous information in a wide range of applications, e.g., location tracking systems [15] and smart city. RDF streams have played an increasingly important role in many application domains such as sensors, feeds, and click streams. Despite the presence of a lot of streaming data applications, there are only a few RDF stream processing systems, such as the C-SPARQL [1], CQELS [2], EP-SPARQL [3]. These engines are centralized engines and are not designed for processing large-scale streaming data. A framework PRSP [5] is presented to process C-SPARQL queries [4] on RDF streams by exploiting various SPARQL query engines in a unified way. PRSP can handle large-scale RDF streaming data by using the

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 141–155, 2018.
https://doi.org/10.1007/978-3-319-91455-8_14

current state-of-the-art distributed SPARQL engine. In a real world, however, federated queries on RDF stream and relational database are important for many applications [6]. For example, in a car rental system, we use RDF stream to record GPS data and use relational database to store information of cars for tracking cars. As an important query, federated query is already recommended by W3C. A federated path querying language (FPQ) [7] based on conjunctive queries is defined to navigate RDF data and relational data in a combined way [14]. However, federated query on RDF stream is rarely researched.

In this paper, we present a framework (FedQL) for federated queries processing on RDF stream and relational data. We mainly discuss C-SPARQL queries to convey our idea simply. We argue that our proposed framework could support most of continuous query languages extending SPARQL [20], such as CQELS [2] and SPARQL$_{\text{stream}}$[13]. Our major contributions are summarised as follows:

- We define a formalization of our federated query language FQ based on the conjunction of continuous queries and SQL queries, where continuous queries and SQL evaluates RDF stream and relational data, respectively.
- We present a white-box-based framework to separate query processing from query executing. The framework contains four parts, namely, Query parser, RDF stream processor, SQL query execution, and Parallel joining. And RDF stream processor consists of three modules, namely, Query processor, Data transformer, and SPARQL query execution.
- We implement FedQL built on C-SPARQL and MySQL by employing four SPARQL engines (incl. centralized engines such as Jena [8], gStore [12], RDF-3X [11] and distributed engines such as TriAD [10]) in an adaptive way. Finally, we evaluate FedQL on a real dataset (i.e., car rental data) and the experimental results show that FedQL is efficient and effective to process the federal query of continuous queries and SQL query.

The remainder of this paper is structured as follows: Sect. 2 introduces RDF stream, C-SPARQL, and relational database. Section 3 describes our federated query language, and Sect. 4 introduces our framework FedQL. Section 5 presents experiments and evaluations. Section 6 summarizes our work.

2 Preliminary

In this section we introduce RDF stream, continuous queries, and relational database.

2.1 RDF Stream

RDF (Resource Description Framework) is the W3C-recommended data model for integrating and representing semantic information on the Web [17]. In RDF model, knowledge is decomposed into a set of unary/binary relations, and every relation is encoded into an RDF triple. RDF model offers a unified and machine-readable way to modify knowledge. Formally, assume three mutually disjoint sets

U, B, L to represent the URI set, blank Node set, and literal set, respectively. An RDF triple is a triple (s, p, o) from $(U \cup B) \times U \times (U \cup B \cup L)$, and an RDF graph is a set of RDF triples.

In order to maintain the interoperability between streaming information and relatively static knowledge, some efforts have been made to extend RDF for representing stream tuples. The streaming information is continuously updated and the content is periodically repeatable. For example, stream tuples are about status of red light. The content "red light is on/off" is continuously and repeatedly updated though, however, every stream tuple is a unique temporal statement about the status of red light at that time point. Compared with persistent knowledge, streaming information has relatively short valid time interval, thus the temporal correlations among them are extremely complex, and if not carefully modelled, will lead to completely different meanings. To accurately capture the temporal information, it is necessary to extend RDF triple with extra time annotation for representing stream tuple.

Example 1. Let us consider an RDF stream *carGPSLocation* coming from the car GPS location data stream. The data stream is about the real-time GPS information of the vehicle. Table 1 shows the pairs of *carGPSLocation*. Every record consists of one RDF triple and a timestamp represented as a 10-bit integer. There are two different ways to encode the time annotation of an RDF stream tuple: timestamp and time interval. In this paper, we use the timestamp encoding since timestamp suits the real-time processing feature of RDF stream better.

Table 1. An RDF stream of car GPS location

Subject (sub)	Predicate (pre)	Object (obj)	Timestamp
car1	isLongitude	116.3217389	1420074000
car1	isLatitude	39.9902739	1420074000
car2	isLongitude	116.3312150	1420074000
car2	isLatitude	40.0640069	1420074000
...	...	...	...
car100	isLongitude	116.4946730	1420074660
car100	isLatitude	40.0402890	1420074660
...	...	...	...

Formally, assume a set T disjoint with $U \cup B \cup L$, representing the timestamp set. Moreover, there is a linear order over the elements in T, denoted by $<_t$, such that $\forall t_i, t_j \in T \wedge i < j \rightarrow t_i <_t t_j$. An RDF stream tuple is a quadruple (s, p, o, t) from $(U \cup B) \times U \times (U \cup B \cup L) \times T$. An RDF stream is an infinite set of RDF stream tuples.

2.2 C-SPARQL

Since streaming information is continuously updated, traditional one-time-query approach is not suitable. In fact, queries about dynamic information should be re-executed as soon as new RDF stream tuples arrive. We take this kind of queries as continuous query.

As the firstly proposed and implemented RDF stream query language, C-SPARQL realizes the continuous SPARQL query functionalities over RDF stream [21]. C-SPARQL bridges the gap between dynamic RDF stream and static RDF graphs by combining the concept of SPARQL and continuous query language. Furthermore, C-SPARQL allows to refer to the most recently updated timestamp of certain kind of RDF stream tuples, to capture fine-grained temporal correlations between RDF stream tuples within window.

For example, the following is a C-SPARQL query Q query $Q_{CarLocation}$:

```
 REGISTER QUERY CarLocation AS
SELECT ?carID ?Latitude ?Longitude
FROM STREAM GPS [ RANGE 30s   STEP 30s ]
WHERE { ?carID <isLatitude> ?Latitude .
            ?carID <isLongitude> ?Longitude . }
```

Definition 1. *Formally, a C-SPARQL query Q can be taken as a 5-tuple of the form:*

$$Q = [\mathrm{Req}, S, \mathrm{w}, \mathrm{s}, \rho(Q)] \tag{1}$$

where

- Req*: the registration;*
- S*: the RDF stream registered;*
- w*:* RANGE*, i.e., the window size;*
- s*:* STEP*, i.e., the updating time of windows;*
- $\rho(Q)$*: a SPARQL query.*

Considering the RDF Stream GPS_{stream} in the Table 1, we can get the initial window data as shown in Table 2.

Table 2. The initial window data

Subject (sub)	Predicate (pre)	Object (obj)	Timestamp
car1	isLongitude	116.3217389	1420074000
car1	isLatitude	39.9902739	1420074000
car2	isLongitude	116.3312150	1420074000
car2	isLatitude	40.0640069	1420074000
...	...	...	...

2.3 Relational Database

Database is a collection of related data under unified management, which can be shared by users, with minimum redundancy, close data connection and high independence of programs. With the continuous development of information technology, the database has been widely used in various industries. MySQL is a very popular open source relational database, which has a multi-user, multi-threaded SQL database server, and it can run on different operating systems.

Consider a relational table that stores location information of all place. Table 3 shows the relational table data.

Table 3. The location information of all place

ID	Name	Latitude	Longitude
1	"25248787"	39.9061898	116.3894568
2	"25248788"	39.9902739	116.3217389
3	"25248789"	40.0640069	116.3312150
...	...	...	...
1000	"25585128"	39.9029396	116.3795085
...	...	...	...

3 Federated Queries

In this section we introduce the syntax and semantic of our federated query language FQ.

A FQ query is formally defined as follows:

$$Q = [Q_1, \ldots, Q_n] \tag{2}$$

where each Q_i is either a C-SPARQL query or a SQL query. Hence, Q may contain multiple C-SPARQL queries and SQL queries.

Regarding the semantics, let $Q = [Q_1, \ldots, Q_n]$ be a FQ query and D is the dataset of form: $D = [S_1, ..., S_m, R_1, ..., R_n]$, where S_j is an RDF stream data and R_k is a relational table data.

The semantics of Q over D is defined as follows:

$$[\![Q]\!]_D = [\![Q_1]\!]_D \bowtie [\![Q_2]\!]_D \bowtie \ldots \bowtie [\![Q_n]\!]_D \tag{3}$$

where $\Omega_1 \bowtie \Omega_2 = \{\mu_1 \cup \mu_2 \mid \mu_i \in \Omega_i \ (i = 1, 2) \text{ and } \mu_1 \sim \mu_2\}$ where $\mu_1 \sim \mu_2$ means $\mu_1(?x) = \mu_2(?x)$ for all common variable $?x$. For each Q_i, $[\![Q_i]\!]_D$ is defined as a set of mappings (as solutions). If Q_i is a SQL query, scheme$(Q_i) = \{?x_1, ..., ?x_n\}$, $[\![Q_i]\!]_R = \{(?x_1 \rightarrow a_1, \ldots, ?x_n \rightarrow a_n), \ldots\}$. If Q_i is SPARQL query, then Q_i is a pattern P of the form P_1 AND P_2 where P_1 and P_2 are

BGPs [18,19]. Now given an RDF graph G and a pattern P, then $[\![P]\!]_G := [\![P_1]\!]_G \bowtie [\![P_2]\!]_G$, where, for any two sets of mappings Ω_1 and Ω_2. Here, two mappings μ_1 and μ_2 are *compatible* [22], written by $\mu_1 \sim \mu_2$, if for every variable $?x \in \mathrm{dom}(\mu_1) \cap \mathrm{dom}(\mu_2)$, $\mu_1(?x) = \mu_2(?x)$.

For example, the following is a FQ query $Q_{CarNumber}$. Line 1 defines an FQ query and line 2 selects the returned result. Lines 3–8 are a C-SPARQL Query, which used to find the vehicle location of the current window. Lines 9–11 are a SQL query, which used to find the place of the given location. The purpose of this query is to continuously poll vehicles in a certain area every 30 s.

```
1. REGISTER QUERY CarNumber AS
2. SELECT ?carID name
3. {
4. { REGISTER QUERY CarLocation AS
5. SELECT ?carID ?Latitude ?Longitude
6. FROM STREAM GPS [ RANGE 30s   STEP 30s ]
7. WHERE { ?carID <isLatitude> ?Latitude .
8.              ?carID <isLongitude> ?Longitude . }   }
9. {SELECT name, Latitude, Longitude
10. FROM MapData WHERE
11. Longitude like '116.36%' and Latitude like '39.92%' }
12. }
```

4 The Framework of FedQL

In this section we mainly introduce the FedQL, a framework for federated queries processing on RDF stream data and relational data. The framework of FedQL is shown in Fig. 1, which contains six main modules: query parser, query processor, data transformer, SPARQL query execution, SQL query execution, parallel joining. Continuous FQ query, RDF stream data, and relational data as the input of the framework are used by the query parser module, data transformer module, and SQL query execution module respectively. The system can continuously generate the query results and feedback the results to the user.

Query Parser. Query parser module is responsible for parsing FQ query. FQ queries, as the input of query parser module, will be parsed and split into two types of queries, namely, continuous queries and SQL queries, which can be addressed in query processor module and SQL query execution module respectively.

Query Processor. The query processor module replies on the information captured by Denotational Graph which is defined as a view on the O-Graph, to obtain parameters of window selector and core SPARQL query $\rho(Q)$ from

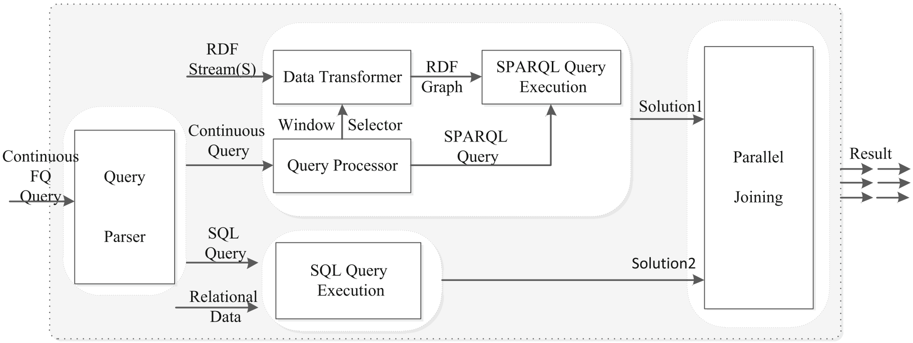


Fig. 1. The framework of FedQL

the input continuous query. The output of query processor module are a 4-tuple (i.e., Req, S, w, s) and a SPARQL query $\rho(Q)$, and they can be addressed in data transformer module and SPARQL query execution module respectively.

Data Transformer. The data transformer module manages the RDF streams via Data Stream Management System (DSMS) such as Esper. It transforms RDF streams into RDF graph data based on the window size and step size at window selector. The RDF graph data can be addressed in SPARQL query execution module.

SPARQL Query Execution. SPARQL query execution module receives the RDF graph data and SPARQL query obtained from the data transformer module and query processor module, then it calls the efficient SPARQL processing engine, such as Jena and TriAD, to execute the query processing in an adaptive mechanism and output the query result (i.e., solution1) to the parallel joining module.

SQL Query Execution. The SQL query execution module receives the SQL query obtained from the query parser module, then it uses the SQL query to query the relational data tables stored in the MySQL database and output the query result (i.e., solution2) to the parallel joining module.

Parallel Joining. The parallel joining module is responsible for performing the join operation on the input solution1 and solution2 to get the final query result, finally it outputs final result to the users.

Considering the FQ query q_1 mentioned in Sect. 3, RDF stream data and relational data in Sect. 2. The process is as follows:

Firstly, query parser module receives input FQ query q_1 and parses it into a C-SPARQL query q_2 mentioned in the Sect. 2 and a SQL query q_3 shown as follow:

```
{SELECT name, Latitude, Longitude
FROM MapData WHERE
Longitude like '116.36%' and Latitude like '39.92%' }
}
```

The query processor module receives the input C-SPARQL query q_2 and parse it into a 4-tuple (i.e., Req, S, w, s) and a SPARQL query q_4, which can be addressed in query processor module and SPARQL query execution module respectively.

Data transformer module periodically converts RDF stream data to static RDF graph data according to window selector obtained from the query processor module. Take the initial window as an example, the initial window data is shown in the Table 2. The SPARQL query execution module receives the RDF graph data and SPARQL query obtained from the data transformer module and query processor module, then it produces the query results (i.e., solution1) shown in the Table 4. The SQL query execution module receives the SQL query obtained from the query parser module and query the table data stored in the MySQL database. We can get the query results (i.e., solution2) shown in the Table 5.

Table 4. The query results of first window data.

No	?carID	?Latitude	?Longitude
1	car1	39.9902739	116.3217389
2	car2	40.0640069	116.3312150
...	...	...	...

Table 5. The query results of relational data.

No	Name	Latitude	Longitude
1	"25248787"	39.9061898	116.3894568
2	"25248788"	39.9902739	116.3217389
...	...	...	...

Finally, the parallel joining module receives the query solution1, solution2 to perform a join operation and output the final FQ query result shown in the Table 6.

Table 6. The query results of FQ query.

No	?carID	Name
1	car1	"25248787"
2	car2	"25248789"
...	...	...

5 Experiments and Evaluations

5.1 Experimental Setup

All centralized experiments were carried out on a machine running Linux, which has 4 CPUs with 6 cores and 64 GB memory, and 4 machines with the same performance for distributed experiments. We use one real-world dataset car rental data and converted the GPS data to RDF stream data. The GPS data contains four size: one day (RD1), 10 days (RD10), 20 days (RD20), 30 days (RD30). The data size increased from 42,000 to 1.6 million. For the relational data, we employ MySQL as the relational database and create a mapdata table. The table stores the location information. We choose the *FQ* query mentioned in the Sect. 3 and use Jena, gStore, RDF-3X and TriAD to process the federal data.

In our experiment, we mainly compare four indicators: Data Load Time (DLT), Query Response Time (QRT), Joining Time (JT), Total Execution Time (TET). Here we do not consider the dynamic update of the relational database. Since the SQL query time does not change with the increase of the stream data size, we do not separately consider the SQL query time, and we only considered it as part of the TET.

5.2 Experimental Results Analysis

To prove the excellent processing performance of our framework under different data sizes, we test the performance of our framework by comparing the time of four indicators with different data input rates. The experimental results are shown in Figs. 2, 3, 4, 5, 6, 7, 8 and 9. By Figs. 2, 3, 4 and 5, on the whole, with the exception of gStore, the SPARQL engine used in our framework is capable of handling RDF streaming data and relational data in real time. The experimental results show that our framework can effectively handle the federal query. In detail, we can find that the DLT and QRT indicators are increasing steadily except for the gStore when the data size is gradually increasing. Because gStore needs to spend a lot of time to build the index, resulting in its lower efficiency for processing RDF streams. Figure 4 is the join time of SPARQL query result and the SQL query result. All engines have almost the same joining time. Jena has the smallest execution time due to its store mechanism from the Fig. 5.

By Figs. 6, 7, 8 and 9, we can more clearly see the processing performance of the different processing engines in our framework. All metrics of Jena shows good performance at different data input rates. TriAD followed the Jena. As the data input rate continues to increase, the performance gap between TriAD and Jena is shrinking. Considering the storage mechanisms of Jena and TriAD, it is foreseeable that as the data loading rate continues to increase, the performance of TriAD will gradually exceed that of Jena.

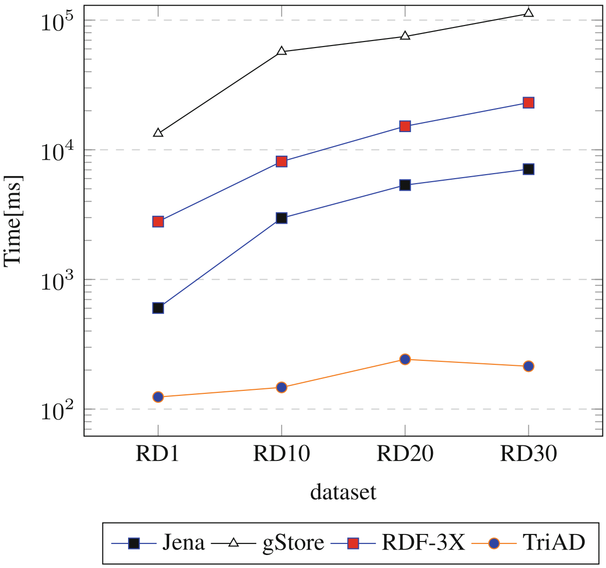


Fig. 2. The data load time within FedQL

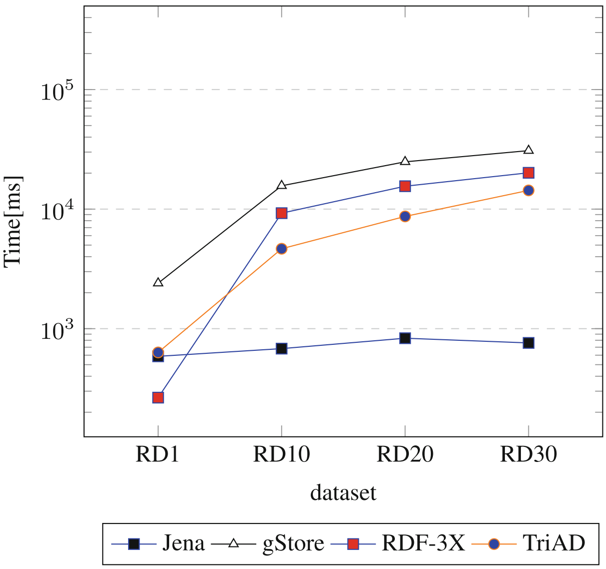


Fig. 3. The query response time within FedQL

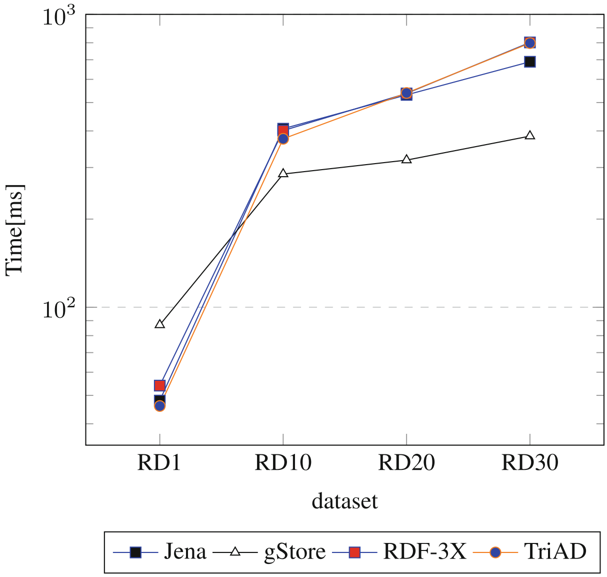


Fig. 4. The joining time within FedQL

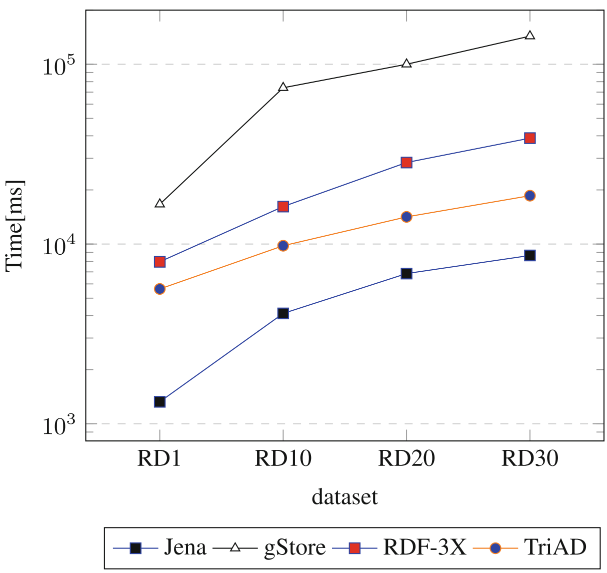


Fig. 5. The total execution time within FedQL

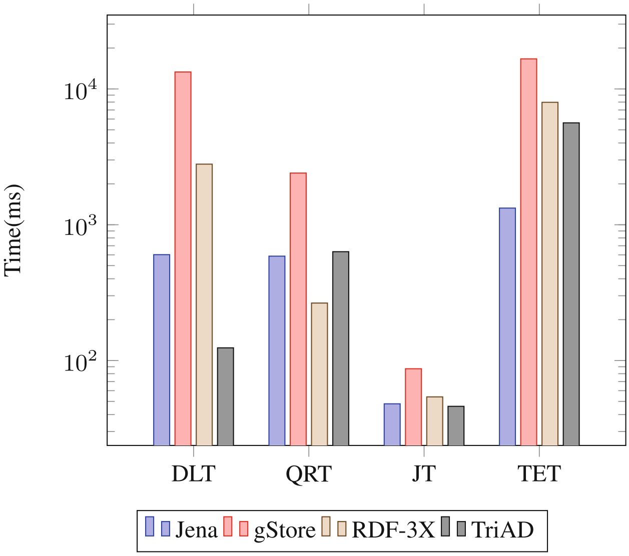


Fig. 6. The query processing time under RD1

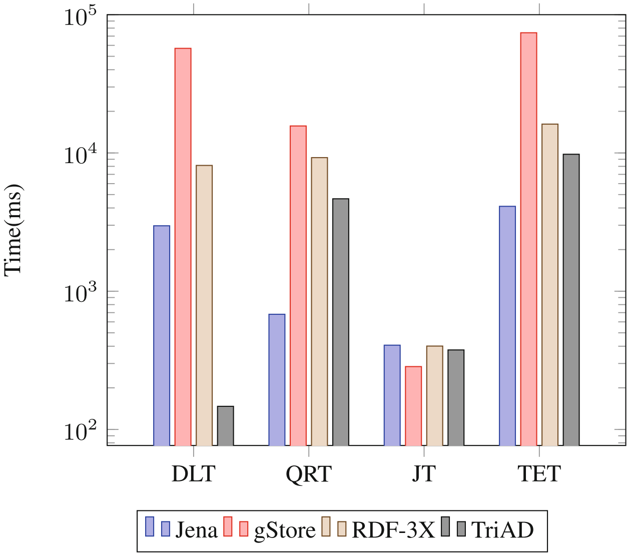


Fig. 7. The query processing time under RD10

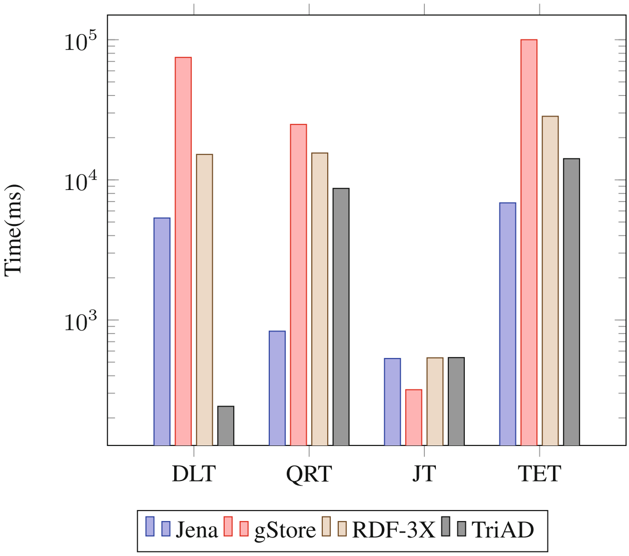


Fig. 8. The query processing time under RD20

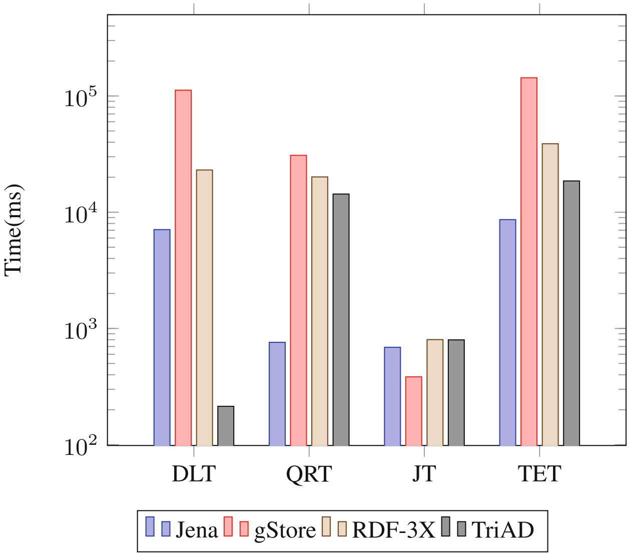


Fig. 9. The query processing time under RD30

6 Conclusions

In this paper, we present a framework for federated queries on RDF stream and relational data for richer querying services in many applications. Our proposal will provide an idea to represent and answer queries from different data models in a federal way.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (61373165, 61672377), the National Key Research and Development Program of China (2016YFB1000603), and the Key Technology Research and Development Program of Tianjin (16YFZCGX00210).

References

1. Barbieri, D.F., Braga, D., Ceri, S., Valle, E.D., Grossniklaus, M.: Querying RDF streams with C-SPARQL. ACM SIGMOD Rec. **39**(1), 20–26 (2010)
2. Le-Phuoc, D., Dao-Tran, M., Xavier Parreira, J., Hauswirth, M.: A native and adaptive approach for unified processing of linked streams and linked data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011. LNCS, vol. 7031, pp. 370–388. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25073-6_24
3. Anicic, D., Fodor, P., Rudolph, S., Stojanovic, N.: EP-SPARQL: a unified language for event processing and stream reasoning. In: Proceedings of WWW 2011, pp. 635–644 (2011)
4. Barbieri, D.F., Braga, D., Ceri, S., Grossniklaus, M.: An execution environment for C-SPARQL queries. In: Proceedings of EDBT 2010, pp. 441–452 (2010)
5. Li, Q., Zhang, X., Feng, Z.: PRSP: a plugin-based framework for RDF stream processing. In: Proceedings of WWW 2017, pp. 815–816 (2017)
6. Ngomo, A.C.N., Saleem, M.: Federated query processing: challenges and opportunities. In: Proceedings of ESWC 2016 (2016)
7. Zhang, J., Zhang, X., Feng, Z.: A path querying language for federation of RDF and relational database. In: Proceedings of WebDB 2017, pp. 41–46 (2017)
8. Carroll, J.J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K.: Jena: implementing the semantic web recommendations. In: Proceedings of WWW 2004 (Alternate Track Papers & Posters), pp. 74–83 (2004)
9. Peng, P., Zou, L., Özsu, M.T., Chen, L., Zhao, D.: Processing SPARQL queries over distributed RDF graphs. VLDB J. **25**(2), 243–268 (2016)
10. Gurajada, S., Seufert, S., Miliaraki, I., Theobald, M.X.: TriAD: a distributed shared-nothing RDF engine based on asynchronous message passing. In: Proceedings of SIGMOD 2014, pp. 289–300 (2004)
11. Neumann, T., Weikum, G.: The RDF-3X engine for scalable management of RDF data. VLDB J. **19**(1), 91–113 (2010)
12. Zou, L., Özsu, M.T., Chen, L., Shen, X., Huang, R., Zhao, D.: gStore: a graph-based SPARQL query engine. VLDB J. **23**(4), 565–590 (2014)
13. Calbimonte, J.-P., Corcho, O., Gray, A.J.G.: Enabling ontology-based access to streaming data sources. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010. LNCS, vol. 6496, pp. 96–111. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17746-0_7

14. Arasu, A., Babu, S., Widom, J.: The CQL continuous query language: semantic foundations and query execution. VLDB J. **15**(2), 121–142 (2006)
15. http://www.openbeacon.org/
16. Kolchin, M., Wetz, P., Kiesling, E., Tjoa, A.M.: YABench: a comprehensive framework for RDF stream processor correctness and performance assessment. In: Bozzon, A., Cudre-Maroux, P., Pautasso, C. (eds.) ICWE 2016. LNCS, vol. 9671, pp. 280–298. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-38791-8_16
17. Margara, A., Urbani, J., Van Harmelen, F., Bal, H.: Streaming the web: reasoning over dynamic data. J. Web Sem. **25**(1), 24–44 (2014)
18. Zhang, X., Feng, Z., Wang, X., Rao, G., Wu, W.: Context-free path queries on RDF graphs. In: Groth, P., Simperl, E., Gray, A., Sabou, M., Krötzsch, M., Lecue, F., Flöck, F., Gil, Y. (eds.) ISWC 2016. LNCS, vol. 9981, pp. 632–648. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46523-4_38
19. Zhang, X., den Bussche, J.V.: On the primitivity of operators in SPARQL. Inf. Process. Lett. **114**(9), 480–485 (2014)
20. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of SPARQL. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 30–43. Springer, Heidelberg (2006). https://doi.org/10.1007/11926078_3
21. Prud'hommeaux, E., Seaborne, A.: SPARQL query language for RDF. W3C Recommendation (2008)
22. Zhang, X.: On the primitivity of SPARQL 1.1 operators. In: Proceedings of WWW 2017, pp. 56–57 (2017)

A Comprehensive Study for Essentiality of Graph Based Distributed SPARQL Query Processing

Muhammad Qasim Yasin[1,3], Xiaowang Zhang[2,3(✉)], Rafiul Haq[1,3],
Zhiyong Feng[1,3], and Sofonias Yitagesu[1,3]

[1] School of Computer Software, Tianjin University,
Tianjin 300350, China

[2] School of Computer Science and Technology, Tianjin University,
Tianjin, China
xiaowangzhang@tju.edu.cn

[3] Tianjin Key Laboratory of Cognitive Computing and Application,
Tianjin, China

Abstract. Distributed SPARQL query processing frameworks are categorized on the bases of query computation into relation, graph and hybrid based distributed query computing. By exploring the historical achievements under these umbrellas we try to motivate the researchers, to define such a framework for Graph Based Distributed SPARQL Query Processing, which supports Full of SPARQL and also explains the principles for employing optimization. In this study we elaborate all popular existing frameworks for distributed query processing and organize a comparative study according to the facts and figures. We identify different limitations and discrepancies in all approaches e.g. only few support the Full of SPARQL, all these are optimized for different kind of benchmarks and all carries own partitioning strategy. We study some valuable query optimization techniques and their implementation. How these techniques are employed in distributed environment. Finally, some future work is highlighted on Graph Based Distributed SPARQL Query Processing which will support all features of SPARQL 1.1 and well optimized.

Keywords: Distributed query processing · Query optimization
Graph-based distributed query computing
Relation-based distributed query computing

1 Introduction

The Resource Description Framework (RDF) is a way to represent the information about the World Wide Web resources. It is recommended by W3C in 2004 [1]. Due to its flexibility and versatility it becomes very popular. With the continuous working of the researchers, latest recommendation by W3C, (the RDF 1.1) is published in 2014. RDF1.1 introduces many new serialization formats such as

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 156–170, 2018.
https://doi.org/10.1007/978-3-319-91455-8_15

Turtle, TriG, N-Triples, N-Quads JSON-LD and RDFa besides RDF1.0 serialized formats like RDF/XML, N-triples. As per (Fig. 1), JSON-LD, N-Quads, Trig supports multiple graphs. Moreover, RDF1.1 explains RDF Dataset as collection of RDF Graphs [2]. For these graphical datasets the W3C recommended SPARQL as a query langaug [3].

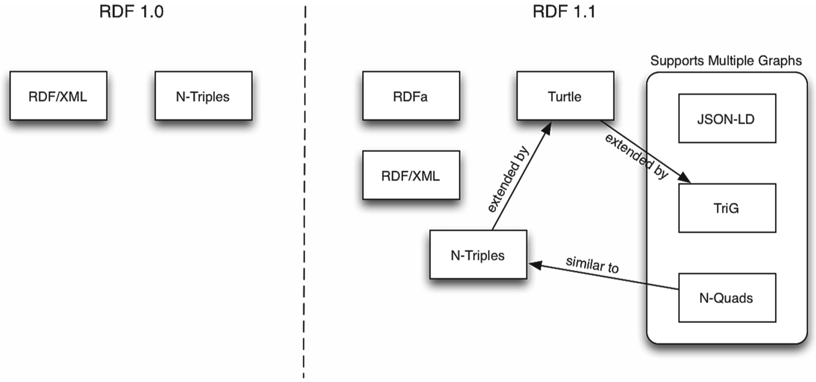


Fig. 1. RDF 1.0 and 1.1 serialization formats [2].

Many large public knowledge bases, including DBpedia [4], PubChemRDF [5], Bio2RDF [6], and UniProt [7], have billions of facts in RDF format. These databases are usually interlinked, and continuously increasing. With the rise in demand and flexible to be interlinked, RDF has been recommended by W3C as a framework to read and write link data on web in 2015. These public knowledge bases can be queried by SPARQL. W3C introduces SPARQL1.1 with novel features (the updated query language for RDF) in 2013. It keeps capabilities for querying required and optional graph patterns (BGP) along with their conjunctions and disjunctions. It also supports aggregation, subqueries, negation, property path, creating values by expressions, extensible value testing, and constraining queries by source RDF graph [8].

Continuous increase in frequent usage, size of data and development has made the processing tasks like (cleaning, sorting, validating, joining, sorting, aggregation) on RDF highly time consuming. To improve the performance of large RDF, researchers employ parallelism and distribution of computation over the hundreds of machines [9]. Some popular frameworks for distributed query processing are surveyed and their highlighted features are presented in Table 1.

Many multidimensional, surveys, comparative studies and analytic studies are carried out on RDF management systems. Most of them, consider data partitioning, query optimization, partitioning strategy and their adaptation towards distributed environment as a key point of their studies [21–25]. They forget to survey about e.g.

Table 1. Distributed query processing framework for RDF

Year	Framework	Supported SPARQL fragmentation	Query computation	Partitioning technique	Execution model
2013	Trinity.RDF [10]	BGP	Graph-based	Key-value store on graph	Graph Exploration
2013	PigSPARQL [11]	Full SPARQL1.0	Relation-based	Hash + Triple-based Files, VP and data parallel	SPARQL-to-PigLatin
2013	WARP [12]	BGP	Graph-based	METIS on query workload	RDF-3X (MapReduce)
2014	TriAD [13]	BGP+filter, union	Graph-based	Hash-based Sharding	Distributed Merge/Hash Join
2014	TriAD-SG [13]	BGP+filter, union	Graph-based	METIS + Horizontal Sharding	Distributed Merge/Hash Join
2014	SEMPALA [14]	Full SPARQL1.0	Relation-based	Unified Property Table	SPARQL to Impala SQL
2015	CliqueSquare [15]	BGP	Relation-based	Hybrid (Hash + VP)	MapReduce Join
2015	DREAM [16]	BGP	Hybrid	No partitioning	RDF-3X
2015	S2X [17]	BGP+filter, optional, orderby, limit, offset	Hybrid	GraphX partitioning strategy for BGP and Data paralllel	Vertex-Centric BGP matching (MapReduce)
2016	AdPart [18]	BGP+filter	Graph-based	Subject Hash + workload adaptive	Distributed Semi-Join
2016	AdPart-NA [18]	BGP+filter	Graph-based	Subject Hash	Distributed Semi-Join
2016	gStoreD [19]	BGP+filter, optional, union, aggregate	Graph-based	Partitioning Agnostic	gStore (novel index (VS∗-tree) with Partial evaluation)
2016	S2RDF-ExtVP [20]	Full SPARQL 1.0	Relation-based	Extended Vertical Partitioning	SPARQL-to-SQL (MapReduce)
2016	S2RDF-VP [20]	Full SPARQL 1.0	Relation-based	Vertical Partitioning	SPARQL-to-SQL (MapReduce)

- Will these RDF management systems support all features of SPARQL?
- Secondly if they support Full of SPARQL then what kind of query computation they employed.
- If query processing is graph based then, has these distributed SPARQL query frameworks employed optimization for all SPARQL algebraic operators?

These questions motivate us to conduct this comprehensive study to fix our future research direction. The rest of the paper is organized as followed Sect. 2 describes query processing and its types based on distributed query computation; Sect. 3 provides the overview of query optimization techniques; Sect. 4 presents the distributed query optimization; Sect. 5 briefed the need of optimized framework for Graph Based Distributed SPARQL Query Processing. Section 6 is about concluding, findings and recommendations.

2 Query Processing

Query processing for distributed environment has become a challenge for the researchers. It gives birth many questions like.

- How can complex queries be distributed over distributed RDF data?
- How can high level queries be transformed into low level query to execute them more efficiently.
- It directs towards the need of declarative query language.

The distributed query processing is very complex. It involves fragmentation /replication, additional communication cost and parallel query execution. As per Table 1 we can classified Different distributed framework for SPARQL on the basis of query computation into three types.

1. Relation-based distributed query computing
2. Hybrid distributed query computing
3. Graph-based distributed query computing

2.1 Relation-Based Distributed Query Computing

If we have a critic review of Table 1, the frameworks of distributed SPARQL query which support Full of SPARQL have implemented relational-based distributed query computing. Those have mapped RDF to some column based or row based Table storage and partitioned the RDF data by following predefine partitioned strategy. In this contrast they translated SPARQL into other well developed structured languages like PigLatin, ImpalaSQL, and Spark-SQL [11,14,20] etc.

In relation mapping, many RDF tripplestores commonly manage RDF in a big Triple Table [26]. It is very flexible but not efficient. It is significantly boils down when series of joins on this triple table are employed. It is not suitable for one-time processing of RDF data. Therefore, more optimized forms like vertical partitioning (VP) [27], and property tables [26], are come into existence with different kind of advantages and drawbacks e.g. In VP some partitions are become very large. Property table becomes weak when there is existence of multi-valued predicates in RDF [28].

2.2 Hybrid Distributed Query Computing

Some approaches followed a Hybrid way to process the distributed SPARQL query like Dream [16], and S2X [17]. S2X process BGP on GraphX [29], and uses Spark RDD (data parallel) for rest of the SPARQL operators like optional, orderby, limit, and offset etc. [17]. While Dream is composed of basically master, worker environment which did not believe on data partitioning [16]. It supports graph-based and relation-based computing/processing on workers sides. That is the reason we categories them under Hybrid query computing.

2.3 Graph-Based Distributed Query Computing

Thirdly, frameworks adopted graph-based approach for computation. Basically, RDF data is highly connected graph data [3], and SPARQL queries are like subgraph matching queries. Many sub graph queries (e.g., community detection) on entity/relationship data only rely on graph operations. Beside, these approaches only insure BGP (Basic Graph Pattern, the fundamental fragment of SPARQL)) the distributed Graph-based computing methodologies cannot be ignored as RDF is inherited from graph. These approaches are proved as efficient and well optimized, especially under large workload. In coming sections we discussed them in details.

3 Query Optimization

Query optimization is a crucial part of the overall query processing. It involves optimal and efficient query evaluation plan with lowest costs. We need to minimize the following *cost function*:

$$\text{cost function} = \text{I/O}_{cost} + \text{CPU}_{cost} + \text{Communication}_{cost}.$$

Any query optimizer module is mostly composed of three main components, e.g., Search space, Cost model, and Searching strategy.

Search space is query optimization plan abstracted by operator trees, which define the execution order of the operations. For a given query the search space is defined as set of equivalent operator trees which are produced by transformation rules as in (Fig. 1). Joining order plays an important role in query optimization as different operator has different costs. For complex queries alternative simple or optimized queries can be attained by applying commutativity and associativity rules. But for some complex queries, query optimizers investigate a large sized search space and make actual execution more expensive. Therefore, query optimizers restricted the size of the search space. One more restriction is about the join tree as there are two kinds of trees, linear tree and bushy tree. In distribution and parallelism, bush tree is more useful.

The equivalent query execution plan is passed to the Search strategy. The most popular strategy is dynamic programing which is deterministic. All possible plan are built, breadth-first and the best plan is chosen. Sometimes it becomes more expensive to build all possible plans. In this contrast greedy algorithm builds only one plan, depth first. Furthermore, for complex queries, randomized strategies are suggested as they perform better than deterministic strategies.

While building search strategy the cost model is incorporated. It predicts the cost of operators statistics and base data. It formulates how the size of intermediate results can be evaluated [25,30].

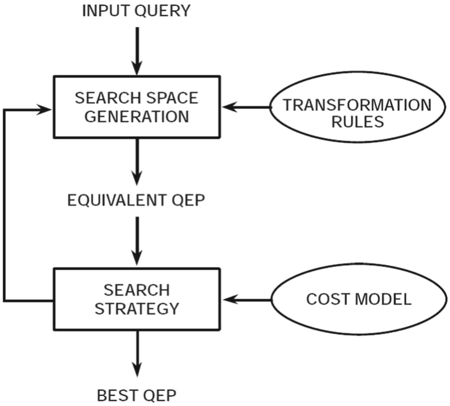


Fig. 2. Query optimization process [30].

For graph based SPARQL query processing some popular optimization techniques are as followed.

1. Query Rewriting Based on Transformation Rules
2. Selectivity Based Query Optimization
3. Mixed Strategy for Query Optimization
4. Using Graph Traversal Algorithms for Query Optimization
5. Query Analysis Based Query Optimization

3.1 Query Rewriting Based on Transformation Rules

For SPARQL query rewriting based optimization algorithm is presented in [31]. The optimization process works in two phases. In first phase SPARQL query is translated into a SQGM (SPARQL Query Graph Model) [31], which is involved in all phases of query processing. Second phase rewrites the query based on generated SQGM to reduce the query execution plan.

In details, during first phase, the SPARQL query is presented in shape of tuple (DS, GP, SM, R) to an algorithm that returns the corresponding SQGM. Where DS, GP, SM, R, are referred as queried RDF dataset, graph pattern, set of solution modifiers and results [3]. In the second phase to achieve a better execution strategy generated, SQGM is transformed into a semantically equivalent. Firstly it defines the semantic equivalence of two SQGMs and then defines transformation rules. Transformation rule merges the join of two graph pattern operators to a single operator. A heuristic is generated with a set of preconditions and a set of rewrite rules. Consequently, it reduces query execution time if all preconditions are fulfilled and rewrite rules are applied.

3.2 Selectivity Based Query Optimization

Selectivity algorithm's first component is BGP (Basic Graph Pattern) Abstraction. SPARQL query is abstracted as an undirected graph. Second component is core optimization algorithm which uses minimum selectivity approach to generate the query execution plan. Third component is set of heuristics that help the optimization algorithm in selectivity estimation. Heuristics without pre-computed statistics and heuristics with pre-computed statistics are presented for triple patterns selectivity and joined triple patterns selectivity. For the static query optimization this algorithm reduces the number of execution plans [32]. It is totally based on static BGP (Basic Graph Pattern). Therefore, they concoct many heuristics for joined triple pattern. In [33] authors employ two kinds of statistic selectivity estimations. One is histogram which can be applied to any kind of triples patterns. The second computes frequent join paths in data. Above mentioned approaches takes joins uniformity assumptions. They do not take into dependencies of the properties. For joined triples Bayesian network and chain histogram are proposed respectively for selectivity estimation. The algorithm uses precomputed statistics for star paths and chain paths [34].

3.3 Mixed Strategy for Query Optimization

In [35], mixed strategy for query optimization is proposed. It is combination of top down [36], and bottom up strategy [37]. Mostly, top down and bottom up approaches are separately adopted. But the mixed approach give advantage of runtimes like triple pattern results, join pattern results and links to results, indexes in computing matrices. In this way it refines the old information and referred as corrective source ranking. It is used for lessening the problem of busy waiting in a loop, stream based approach with an operator symmetric hash join (SHJ). It is 42% faster than bottom up strategy.

3.4 Graph Traversal Algorithms

Graph traversal algorithms are popular for SPARQL query optimization due to its unique feature of representing the whole query as a graph [3,38]. Basic Graph Pattern (BGP) is exemplified as a directed graph of subjects as node and predicate as edge. The weight of each edge can be calculated as the cost of evaluating the corresponding triple pattern. Graph traversing algorithms generate the optimal query plan, corresponding to the minimum spanning tree [38]. In [39], Edmonds algorithm and prims algorithm are applied to optimize the SPARQL query. Firstly, static query execution plan is generated before the query execution by using prims algorithm [40], or (Edmonds algorithm) [41]. The potential solution is followed as an independent query plan, the execution plan generated by phase one is altered by an adaptive approach using prims algorithm.

In [39], the methodology have shortcoming like it does not keep separate data copies for different iterations, new binding retrieval process could not produce right results. These shortcomings are overwhelmed and presented in [38]. In spite of all the above overcoming, method of [38], suffers from new challenges like it is consumed unexpected time while executing a complex query having many triple patterns. It cannot execute multiple triple patterns in parallel.

There are many opportunities for further work in the area of distributed SPARQL optimization, as a comprehensive solution has not yet been proposed by the research community. One such opportunity is the use of parallelized algorithms, given that each source may be queried independently. Surprisingly, there has been no mention of this idea in the literature regarding SPARQL systems [39].

3.5 Query Analysis Based Query Optimization

In [42], SPAQRL query is presented as directed graph and it is traversed by the algorithms fetching of classes is discouraged that could not contribute to answer of the query. It works in two phases. Query analysis is done before query execution is the first step. The classes that cannot contribute to the result of query are ignored. In second step a context graph [42], is formed as model for execution. This pattern is used by heuristic to analyze more patterns that can only be discovered at run time and it further reduces the amount of data fetched from web to answer the query results. This approach reduces the query execution time up to some extent and improves query performance.

4 Optimization for Distributed SPARQL Query

Presently it is the core issue to engage the above revealed techniques for distributed SPARQL query. But it is difficult to apply because SPARQL owns different data representation (i.e. relations and triples). Current approaches only optimize some part of the query evaluation process [21]. The accuracy of the query optimization framework (see Fig. 2). For distributed environment depends upon good knowledge of cost model about the distributed execution environment. It helps in defining the order and effeteness of the execution plan. As a result, it affects the search space as search strategy explores the search space [30].

4.1 Distributed Cost Model

A cost function of a distributed execution strategy can be expressed with respect to total time or response time. Cost is generally expressed in unit time but can be expressed into other units.

Total Time. It is the sum of all times and can be expressed as

$$Total\_time = T_{CPU} * \#insts + T_{I/O} * \#I/O_s + T_{MSG} * \#msgs + T_{TR} * \#bytes \quad (1)$$

Where T_{CPU} = time of a CPU instruction, $T_{I/O}$ = time of a disk I/O, T_{MSG}= fixed time of initiating and receiving message and T_{TR} =time taken to transmit data in terms of bytes from one location to other.

Communication Time. It is the time to transfer bytes of data from one location to other and expressed as

$$CT\,(\#bytes) = T_{MSG} + T_{TR} * \#bytes \quad (2)$$

In local area Network it is considered as same for all. But on wide area networks it is considered as dominant time factor. As SPARQL is query language of public knowledge bases which are with scattered data on internet. So communication time for distributed SPARQL query optimization is domineering.

Response Time. It is duration time from the initiation time of the query to the completion time.

$$\begin{aligned} Response\_time = {} & T_{CPU} * seq\_\#insts + T_{I/O} * seq\_\#I/O_s \\ & + T_{MSG} * seq\_\#msgs + T_{TR} * seq\_\#bytes. \end{aligned} \quad (3)$$

Here if seq_#y then y, can be instruction, I/O, messages or bytes and y which must done in execution sequences. Parallel processing is ignored at the time being [30].

Cost function is to reduce total time (Eq. 1) and response time (Eq. 3). When we reduce the response time by employ the parallel processing but sometimes total time is increased. Total time is reduced by decreasing all of its component and intelligent use of resources.

In addition, the primary cost factor is size of intermediate relations. Which produce during execution and need to be transmitted over network? To estimate the size intermediate relations global statistics of relations and fragments are to be used. Selectivity factor, cardinality of different intermediate results and joining order are also affective implement in relational calculus [30].

4.2 Query Optimization in Graph Based Query Computing

The distributed SPARQL query engines like AdPart-NA [17], and TriAD [12], which employed query optimization techniques and they shows high performance and out class others in comparative study. In this study they take lowest query run time. AdPart-NA is proved as the best choice for reducing end to end workload runtime by adopting dynamically its data distributions as workload.

In rest of this section we will discuss what kind of distributed cost model and other optimization techniques they have adopted for execution plan [24]. The both [12,17], follows the locality aware query planner. TriAD constructs summary graph to maintain the partition information while Adpart-Na [17], exploits hash based data locality. These both reduce the communication overhead. In consecutive paragraphs we take into account the overview of optimization during query processing for both approaches TriAD and AdPart-NA simultaneously.

In [12], SPARQL query is translated into a labeled directed multi graph GQ $(V_Q, E_Q, L, Vars, \varphi_Q)$ where V$_Q$, is set of query nodes, E$_Q$ is the set of edges connecting nodes in V$_Q$, L is the set of edge and node labels, and φ_Q is a labeling function with $\varphi_Q : V_Q \cup E_Q \rightarrow Vars \cup L$. Unique Id for each distinct Vars from the forward dictionary is assigned in L by replacing the constant. E$_Q$ is referred as a set of triple pattern that capture cunjective queries. It refers two stages query optimization. In first stage it employs exploratory [10,43], based algorithm over conventional joins. It finds the supernode binding for each query variable for facilitating the Join-ahead pruning at actual permutation indexes. Exploratory plan uses first DP-based optimizer over summary graph for best ordering. In Second stage TriAD follows relational style of processing uses second DP-based Algorithm [44], in combination with distribution aware cost model as objective function. Supernode bindings obtained in pruning, incorporated into the cost model used in re-estimation of cardinality. The global query planner passed entire summary graph to the slaves. At each slave, the local query processor executes the plan by asynchronously sending and/or receiving intermediate join results to/from the other nodes. Each slave passed subquery results to master which finally merged them.

Adpart-NA [17], uses cost based optimizer based on Dynamic Programming (DP) for finding best subquery ordering. It uses statistic for cost calculation. It collects and aggregates all statistic form workers during adaptive process. On the base of this statistic it plans the global query planning. It focuses on storing unique predicates to avoid the data size overhead. It calculates the cordiality of unique subject and object using predicate and computes the subject and object score for unique predicate and then calculates the average. It calculates the cardinality of the subqueries. The master consult works to be updated about the cordiality of subqueries patterns. Beside cost based optimizer Adpart-NA also introduced pinned subject approach. Under subject hash portioning, combing right-deep tree planning and Distributed semi-join algorithm causes the intermediate and final result to be local to the subject of the first executed subquery pattern refers red as pinned subject. Typically, it also employs the log-based recovery and introduced hierarchical heat map to monitor the workload. Consequently, we can says that success of the adPart-Na depends upon employment of multiple optimization technique, log recovery and heat map to monitor the workload. In comparison we also discuss the query optimization techniques adopted by relation-based query framework.

4.3 Query Optimization in Relation Based Query Computing

Sempala [14], and S2RDF [19], most frequently discussed research in relation based query computing did not take optimization as a core factor. S2RDF elevate the dangling tuples by implementing Extended Vertical portioning (ExtVP) over vertical partitioning VP. ExtVP defines the correlation of semi joins as subject to subject (SS), subject to object (SO), Object to object (OO) and object to subject (OS). ExtVP query processing have the main idea is to divide query into subqueries for every triple. It involved heuristic algorithm for identifying the smaller join inputs which leads to smaller output. Joining order is defined according to the size of the ExtVP Table. The subquery with smaller ExtVP is considered first. The triple with most bounded components is prioritized. Ultimately, joining of the all subqueries provided the result. In a recent comparative study [24], S2RDF has higher preprocessing overhead and failed to answer the query for Bio2RDF in 24 h. While Sempala implemented Unified property table which is a single property table consisting of all RDF data. It is mainly designed to entertain the star shape queries in efficient manner. When it translated SPARQL query into SPARQL algebra it only applied filter pushing technique for optimization.

5 Discussion

As RDF is inherited from graph and SPARQL is the recommend language for RDF. It raised a question why researchers translating SPARQL to SQL. The answer, it is easy to use the expressive power of SQL for distributed environment. We reviewed some articles about expressive power of SPARQL, it is discovered that expressive power of SPARQL and its fragmentation are the core issues for researchers. It is well known, SPARQL as a whole has the same expressive power as first-order logic and relational algebra [45–47], and most of the SPARQL operators are primitive that are not expressible via each other [48]. The core SPARQL algebra is composed of operators like Join, Union, Filter, Projection and Optional. First four are corresponding to positive relational algebra with inequalities [49]. Relational calculus can successfully be fragmented. Its fundamental fragment is Conjunctive queries (CQs) and then it extended as union of CQs and CQs of inequalities. Only optional (opt) is a distinctive feature of SPARQL. It motivated us to define the different fragments of SPARQL in future, which will ensure the efficient resolution of the complex queries for distributed environment. Graph based distributed SPARQL query frameworks are efficient for large dataset. It is because of their employment of query optimization techniques, e.g. query analysis intermediate result pinning vs. sharding and right-deep vs. bushy tree planning. These systems are well aware about their distribution have no dependencies like cloud based systems [24]. But most optimized system AdPart-Na [17], and TriAD [12], only support BGP fragment, not Full of SPARQL 1.1 [50]. They are only optimized the BGP queries. They did not calculate the cost function for different SPARQL algebraic operators. They calculated the only cardinality for BGP triples.

6 Conclusions

In this study we categories Distributed SPARQL query processing frameworks according to query computation. These categories are critically reviewed for finding some weaknesses and discrepancies like: (i) All distributed query processing framework who adopted relation based computing support Full of SPARQL but these are not workload tolerant. These consume more than 24 h to answer the complex query for large datasets [40]. These are not well optimized. They translated SPARQL to SQL and RDF to some relation based form. It is observed that if we mapped RDF to some relation based form then SQL querying language is available for querying relational mapping of RDF. There is no need to employed extra computation for translating SPARQL to SQL. (ii) Graph based query computing approaches are well optimized for BGP only (see Table 1) and specific type of benchmark. These approaches do not support Full of SPARQL. (iii) Every approach for query processing follows its own partition strategy except [19].

Therefore, we believe that researchers should emphasis on defining a framework for Graph Based Distribute SPARQL Query Processing which support Full of SPARQL. It can be accomplished by defining fragments of SPARQL by describing the complexity level of SPARQL operators. Secondly optimization techniques need to be incorporated in such frameworks. It desires to explain a distributed cost model for different SPARQL algebraic operators like optional, negation and join etc. Thirdly, these frameworks required to be partition tolerant for distributed SPARQL query processing. In future we will propose the optimized framework for graph based distribute SPARQL query processing.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (61502336, 61672377), the National Key Research and Development Program of China (2016YFB1000603), and the Key Technology Research and Development Program of Tianjin (16YFZCGX00210).

References

1. W3C: RDF Primer. http://www.w3.org/TR/rdf-primer/. Accessed 1 Mar 2018
2. W3C: RDF 1.1. https://www.w3.org/TR/rdf11-new/. Accessed 4 Mar 2018
3. Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF. W3C Recommendation (2008)
4. DBpedia. http://dbpedia.org/. Accessed 3 Mar 2018
5. PubChemRDF. http://pubchem.ncbi.nlm.nih.gov/rdf/. Accessed 26 Feb 2018
6. Bio2RDF. http://bio2rdf.org/. Accessed 20 Feb 2018
7. UniProt. http://www.uniprot.org/. Accessed 21 Feb 2018
8. SPARQL1.1. https://www.w3.org/TR/sparql11-query/. Accessed 4 Mar 2018
9. Koutris, P.: Query processing for massively parallel systems, University of Washington, pp. 2–5 (2015)
10. Zeng, K., Yang, J., Wang, H., Shao, B., Wang, Z.: A distributed graph engine for web scale RDF data. Proc. VLDB Endow. **6**, 265–276 (2013)
11. Schätzle, A., Przyjaciel-Zablocki, M., Lausen, G.: PigSPARQL: mapping SPARQL to Pig Latin. In: Proceedings of SWIM 2011, pp. 4:1–4:8 (2011)

12. Hose, K., Schenkel, R.: WARP: workload-aware replication and partitioning for RDF. In: Proceedings of ICDE 2013 Workshops (2013)
13. Gurajada, S., Seufert, S., Miliaraki, I., Theobald, M.: TriAD: a distributed shared-nothing RDF engine based on asynchronous message passing. In: Proceedings of SIGMOD (2014)
14. Schätzle, A., Przyjaciel-Zablocki, M., Neu, A., Lausen, G.: Sempala: interactive SPARQL query processing on hadoop. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8796, pp. 164–179. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11964-9_11
15. Kaoudi, Z., Manolescu, I., Zampetakis, S.: CliqueSquare: flat plans for massively parallel RDF queries. In: Proceedings of ICDE 2015, pp. 771–782 (2015)
16. Hammoud, M., Rabbou, D.A., Nouri, R., Beheshti, S.-M.-R., Sakr, S.: DREAM: distributed RDF engine with adaptive query planner and minimal communication. Proc. VLDB **8**(6), 654–665 (2015)
17. Schätzle, A., Przyjaciel-Zablocki, M., Berberich, T., Lausen, G.: S2X: graph-parallel querying of RDF with GraphX. In: Wang, F., Luo, G., Weng, C., Khan, A., Mitra, P., Yu, C. (eds.) Big-O(Q)/DMAH -2015. LNCS, vol. 9579, pp. 155–168. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41576-5_12
18. Harbi, R., Abdelaziz, I., Kalnis, P., Mamoulis, N., Ebrahim, Y., Sahli, M.: Accelerating SPARQL queries by exploiting hash-based locality and adaptive partitioning. VLDB J. **25**(3), 355–380 (2016)
19. Peng, P., Zou, L., Özsu, M.T., Chen, L., Zhao, D.: Processing SPARQL queries over distributed RDF graphs. VLDB J. **25**(2), 243–268 (2016)
20. Schätzle, A., Przyjaciel-Zablocki, M., Skilevic, S., Lausen, G.: S2RDF: RDF querying with SPARQL on Spark. Proc. VLDB **9**(10), 804–815 (2016)
21. Dadhaniya, D.R., Makwana, A.: Survey paper for different SPARQL query optimization techniques. MJSRE J. **2**(8), 83–85 (2016)
22. Özsu, M.T.: A survey of RDF data management systems. Front. Comput. Sci. **10**(3), 418–432 (2016)
23. Ma, Z., Capretz, M.A.M., Yan, L.: Storing massive resource description framework (RDF) data: a survey. Knowl. Eng. Rev **31**(4), 391–413 (2016)
24. Abdelaziz, I., Harbi, R., Khayyat, Z., Kalnis, P.: A survey and experimental comparison of distributed SPARQL engines for very large RDF data. Proc. VLDB **10**(13), 2049–2060 (2017)
25. Aljanaby, A., Abuelrub, E., Odeh, M.: A survey of distributed query optimization. Int. Arab J. Inf. Technol. **2**(1), 48–57 (2005)
26. Wilkinson, K., Sayers, C., Kuno, H., Reynolds, D.: Efficient RDF storage and retrieval in Jena2. In: Proceedings of SWDB, pp. 131–150 (2003)
27. Abadi, D.J., Marcus, A., Madden, S.R., Hollenbach, K.: Scalable semantic Web data management using vertical partitioning. In: Proceedings of VLDB 2007, pp. 411–423. (2007)
28. Schätzle, A.: Distributed RDF querying on hadoop, University of Freiburg, pp. 124–127 (2016)
29. Gonzalez, J.E., Xin, R.S., Dave, A., Crankshaw, D., Franklin, M.J., Stoica, I.: GraphX: graph processing in a distributed dataflow framework. In: Proceedings of 11th USENIX OSDI 2014, pp. 599–613 (2014)
30. Özsu, M.T., Valduriez, P.: Optimization of distributed queries. In: Özsu, M.T., Valduriez, P. (eds.) Principles of Distributed Database Systems, 3rd edn, pp. 245–295. Springer, New York (2011). https://doi.org/10.1007/978-1-4419-8834-8_8

31. Hartig, O., Heese, R.: The SPARQL query graph model for query optimization. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 564–578. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72667-8_40
32. Stocker, M., Seaborne, A., Bernstein, A., Kiefer, C.: SPARQL basic graph pattern optimization using selectivity estimation. In: Proceedings of WWW 2008, pp. 595–604 (2008)
33. Neumann, T., Weikum, G.: RDF-3X: a RISC-style engine for RDF. Proc. VLDB **1**(1), 647–659 (2008)
34. Huang, H., Liu, C.: Estimating selectivity for joined RDF triple patterns. In: Proceedings of CIKM 2011, pp. 1435–1444 (2011)
35. Ladwig, G., Tran, T.: Linked data query processing strategies. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010. LNCS, vol. 6496, pp. 453–469. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17746-0_29
36. Harth, A., Hose, K., Karnstedt, M., Polleres, A., Sattler, K.-U., Umbrich, J.: Data summaries for on-demand queries over linked data. In: Proceedings of 19th WWW 2010 (2010)
37. Hartig, O., Bizer, C., Freytag, J.-C.: Executing SPARQL queries over the web of linked data. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 293–309. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04930-9_19
38. Wang, X., Tiropanis, T., Davis, H.C.: Evaluating graph traversal algorithms for distributed SPARQL query optimization. In: Pan, J.Z., Chen, H., Kim, H.-G., Li, J., Wu, Z., Horrocks, I., Mizoguchi, R., Wu, Z. (eds.) JIST 2011. LNCS, vol. 7185, pp. 210–225. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29923-0_14
39. Vandervalk, B.P., McCarthy, E.L., Wilkinson, M.D.: Optimization of distributed SPARQL queries using Edmonds algorithm and Prims algorithm. In: Proceedings of CSE 2009, pp. 330–337 (2009)
40. Prim, R.C.: Shortest connection networks and some generalizations. Bell Syst. Tech. J. **36**(6), 1389–1401 (1957)
41. Edmonds, J.: Optimum branchings. J. Res. Natl. Bur. Stand. **71B**, 233–240 (1967)
42. Reddy, B.R.K., Kumar, P.S.: Optimizing SPARQL queries over the web of linked data. In: Proceedings Workshop on Semantic Data Management (VLDB) (2010)
43. Atre, M., Chaoji, V., Zaki, M.J., Hendler, J.A.: Matrix bit loaded: a scalable lightweight join query Processor for RDF data. In: Proceedings of WWW 2010, pp. 41–50 (2010)
44. Neumann, T., Weikum, G.: The RDF-3X engine for scalable management of RDF data. VLDB J. **19**(1), 91–113 (2010)
45. Polleres, A., Peter, J.: On the relation between SPARQL 1.1 and answer set programming. J. Appl. Non-Class. Logics **23**(1–2), 159–212 (2013)
46. Angles, R., Gutierrez, C.: The expressive power of SPARQL. In: Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 114–129. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88564-1_8
47. Kostylev, E.V., Reutter, J.L., Romero, M., Vrgoč, D.: SPARQL with property paths. In: Corcho, O., et al. (eds.) ISWC 2015. LNCS, vol. 9366, pp. 3–18. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25007-6_1
48. Zhang, X.: On the primitivity of SPARQL 1.1 operators. In: Proceedings of WWW 2017, pp. 875–876 (2017)

49. Kontchakov, R., Kostylev, E.V: On expressibility of non-monotone operators in SPARQL. In: Proceedings of KR 2016, pp. 369–378 (2016)
50. Feng, J., Meng, C., Song, J., Zhang, X., Feng, Z., Zou, L.: SPARQL query parallel Processing: a survey. In: Proceedings of BigData Congress 2017, pp. 444–451 (2017)

Developing Knowledge-Based Systems Using Data Mining Techniques for Advising Secondary School Students in Field of Interest Selection

Sofonias Yitagesu[1,2](✉), Zhiyong Feng[1,2], Million Meshesha[3],
Getachew Mekuria[4], and Muhammad Qasim Yasin[1,2]

[1] School of Computer Software, Tianjin University,
Tianjin 300350, People's Republic of China
sofoniasyitagesu@yahoo.com
[2] Tianjins Key Laboratory of Cognitive Computing and Application,
Tianjin 300350, People's Republic of China
[3] School of Informatics, Addis Ababa University, Addis Ababa, Ethiopia
[4] College of Computing, Debre Berhan University, Debre Berhan, Ethiopia

Abstract. Ethiopia gives a highly emphasis on a secondary school and reform program of impressive expansion. In doing this, students interest towards the fields they are assigned to needs to be taken into consideration. This has been put into practice when the Ministry of Education and preparatory schools have assigned students in fields of studies based on their performance at secondary schools. However, they used only the students grade 10 Ethiopian General School Leaving Certificate Examination result to assign them. The objective of this study is to develop a knowledge-based systems using machine learning (data mining) techniques that consults the students in their field of study selection process. In this study, the hybrid model that was developed for academic research is used. To build the predictive model, 9364 sample students data from selected secondary schools are used. The sample data is preprocessed for missing values, outliers, noisy and errors. Then the model is experimented using decision tree (j48) and rule induction (PART) algorithms. In this study as compared to j48, the PART unpruned decision list algorithm has 98.003% predictive performance. Thus, the knowledge discovered with this algorithm is further used to build the knowledge-based systems. Hence, the Java program is used to integrate data mining results to knowledge-based systems. As a result, the developed knowledge based-systems is used to predict students field of study based on their performance at secondary school. The study concludes that, to build the accurate knowledge-based systems discovering knowledge using data mining techniques is significant.

Keywords: Data mining techniques · Knowledge-based systems
Mapping · Use of knowledge

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 171–183, 2018.
https://doi.org/10.1007/978-3-319-91455-8_16

1 Introduction

1.1 Background

In governmental and private secondary school in Ethiopian education systems, field of interest is selected by students based on the score of the student in Ethiopian General School Leaving Certificate Examination (EGSLCE), the capacity of the preparatory school and the governments policies and strategies which influence more of students to select science fields. However, the existing systems is based on the point of view of preparatory schools to receive new students and students interest based on their results, but not on the basis of secondary schools that are sending their students to pursue higher, and the preparatory schools that knows very little about the applicant, the secondary school knows a great deal more about their student. That is, those scoring higher results than others can have the chance to be assigned to field of study based on their choice [1].

As per the revised manual of higher and preparatory education students placement, students have the right to choose and study any field of study in order of applicant's interest (Ministry of Education, 2011). Even though ministry of education consider students' interest, there are always students who are assigned to field of study which is not their choice or interest. Such students complain about and become disappointed with the field they are assigned to. Moreover, according to Ministry of Education of Ethiopia, student who have completed their secondary school education and fulfilled minimum result of the year in EGSLCE will choose their field of interest based on their interest to join their preparatory education. However, a major problem student face is selecting a field of interest that is appropriate for them to follow in preparatory school which otherwise will affect their future education.

Most of the students are not clear to know about each field of interest and make the selection without compressive advice from expert. They only follow advice from families, their senior and graduated students from universities and they look output of future job engagement. However these did not help them in selecting a field of study that is best fit to them. To overcome this problem, students need expert advice that enables them to decide which field of stream is appropriate for them for their future education based on different background factors and their performance to the specific field of interest using their historical data.

Currently, data mining and knowledge base systems could have been found in many applications both commercially and the research community [2]. Knowledge based systems (KBS) can help link and integrate all available knowledge sources, including explicit knowledge (various kinds of databases stored in existing information systems) and in-explicit knowledge (practical experience, skills, thought and thinking method in the brain of the experts) to form knowledge databases of various kinds [3].

Data mining (DM) is the use of algorithms to discover hidden knowledge. It has attracted a great deal of attention in the information industry and in society as a whole in recent years; wide availability of huge amounts of data and the imminent need for turning such data into useful information and Knowledge Market analysis, fraud detection, and customer retention, production control and science exploration. Important decisions are often made based not on the information-rich data stored in data repositories, but rather on a decision makers intuition. The decision maker does not have the tools to extract the valuable knowledge embedded in the vast amounts of data.

Rule induction is one of the major classification techniques of data mining and is perhaps the most common method of knowledge discovery in unsupervised learning systems. It is the process of extracting useful rules in the form of if then from data based on statistical significance. A Rule based system constructs a set of if-then-rules. Rule induction knowledge representation has the following form

IF conditions THEN conclusion

This rule consists of two parts. The rule antecedent (the IF part or left part) contains one or more conditions about value of predictor attributes whereas the rule consequent (THEN part or right part) contains a prediction about the value of a goal attribute. The authors further explained that, an accurate prediction of the value of a goal attribute will improve decision-making process. IF-THEN prediction rules are very popular in data mining; they represent discovered knowledge at a high level of abstraction. In the field of study selection system, it can be applied as follows:

(Background performance in secondary school) implies (field of interest)

Example: If then rule induced in the field of study Natural Science: when the rules are mined from the database the rules can be used either for better understanding of the business problems that the data reflects or for performing actual predictions against some predefined prediction target and it is helpful for decision making in the field of study selection system.

Moreover, decision tree training algorithms (J48) have been used for classification in different application areas, such as medicine, manufacturing and production, financial analysis, astronomy, and molecular biology.

Therefore, in this study the main goal is firstly, the students result will be predicted and the predicted model will used for ministry of education to adjust the intake capacity of the preparatory schools and students field of study. Then integrate the output model to knowledge based systems to recommend or advice the students in their field of study selection early.

Hence, to address this problem the researcher use integrated model of DM with knowledge base systems to predict model and integrate it with knowledge based systems to advise students in their field of study selection that will appropriate to them, considering all back ground factors and students secondary school scores.

To this end, this study investigates and addresses the following research questions.

- What are the relevant attributes used for prediction model?
- How to develop an integrated prototype model for field of study selection?
- To what extent the integrated prototype model provides appropriate advice for students in their field of study selection?

2 Materials and Methods

In this study a hybrid knowledge discovery process model is followed. This model takes lessons both from the industrial and academic models. It was developed based on the CRISP-DM model by adopting it to academic research [4]. A hybrid data mining process model is a six step process which includes understanding the business problem, understanding the data, data preparation, data mining, evaluation of the discovered knowledge and use of discovered knowledge [5].

2.1 Problem Understanding

To improve the education planning and strategies of Ethiopian secondary education, in focusing on field of study selection, domain problem understanding is assessed and investigated in depth with domain experts for constructing field of study selection predictive model.

In the study, the researcher identified the core domain experts and purposively selected domain areas from Amhara, Oromia, Addis Ababa and Debub such as secondary schools, Preparatory Schools students and Ministry of Education to define the business problem and determine the data mining goal. In addition to this, observations, review of different articles and pilot study was done about how the business process was performed so as to understand the problem area noticeably. To understand clearly, the researcher has attempted to discuss the issue by classifying into three major tasks and consequently identify what are the inputs used in the area? How inputs are processed? What outputs are expected in return?

Firstly, admission to preparatory schools based on secondary school performance is therefore a topic of importance [6]. How a student chooses a field of study, and conversely how a Preparatory school places a student in appropriate field of study, determine the success of both sides in carrying through the future higher education [6].

Thus, studying appropriate field within students interest according to their historical performance is important [1]. In this regard, a novel prediction model that can provide recommendations for students in their decision making about which field of study a student should apply to, taking not only the students secondary school scores but also considering other background factors like gender and school completed into account [6].

Secondly, according to MOE, as cited in [1,7], the intake capacity and availability of field of study in the preparatory school and higher education institutions at large affects the MOE in field of study placement process to assign students with their interest. Placing students according to the intake capacity and available fields in the preparatory is one of the important points of the MOE that aims to place students with their interest so as to resolve the problems arisen in field of study selection processes.

Thirdly, though few, past researcher has suggested that students backgrounds and other factors correlate to the performance of their preparatory and tertiary education [6]. In this regard, active supports on secondary school students were given to address the problem by collaborative actions with secondary school teachers, families and school administrators.

Therefore, in the MOE and students side the secondary school have a great attention in field of study selection so as to avoid factors which may cause wrong field of study selection. Potential interactions among predictors and field of study selection were assessed from various sources to put forth the predictors affecting field of study selection and to create a model that is useful for prediction.

2.2 Data Understanding

The data employed for this study was collected from purposively selected secondary schools students from Amhara, Oromia, Addis Ababa and Debub. From the whole data set, only 2014–2015 is considered in order to suit the research goal in the study. This data were collected from both natural science and social students in the secondary schools randomly. The intent of this study is to discover hidden knowledge based on primary data that was collected from selected secondary students form filled by their own hand to input to the knowledge bases.

Initially, the data set were collected in paper file format, so the researcher prepared a form in MS-Excel 2010 program by selecting the attributes defined in the problem domain. During further processing the relevant attributes are selected. The importance of attributes in field of study predictive modeling is checked by maximum gain ratio, GainratioAttributeEval search method and WEKA attribute selection ranker method. Hence, experiments are built based on the selected attributes. List of raw data variables in the initial data set is shown in Table 1.

2.3 Data Preprocessing

According to Han and Kamber [8], the data processing task of data mining includes data cleaning, data integration, data reduction, and data transformation. Before feeding data to Data Mining we have to make sure the quality of data. Well-accepted multidimensional data quality measures such as accuracy, completeness, consistency, timeliness, believability and interpretability are preprocessed.

Table 1. List of raw data variables in the initial data set.

Category	Field	Attribute	Data Type	Descriptions
Students Grade 9 result	1	ID Number	Nominal	Identification number
	2	Name	Nominal	Name of students
	3	Sex	Nominal	Gender of student
	4	Age	Nominal	Age of student
	5	School Completed	Nominal	Secondary school completed
	6	English	Nominal	Secondary School Result
	7	Mathematics	Nominal	Secondary School Result
	8	Physics	Nominal	Secondary School Result
	9	Chemistry	Nominal	Secondary School Result
	10	Biology	Nominal	Secondary School Result
	11	Civic & ethics	Nominal	Secondary School Result
	12	Geography	Nominal	Secondary School Result
	13	History	Nominal	Secondary School Result
Students Grade 10 result	14	English	Nominal	Secondary School Result
	15	Mathematics	Nominal	Secondary School Result
	16	Physics	Nominal	Secondary School Result
	17	Chemistry	Nominal	Secondary School Result
	18	Biology	Nominal	Secondary School Result
	19	Civic & Ethics	Nominal	Secondary School Result
	20	Geography	Nominal	Secondary School Result
	21	History	Nominal	Secondary School Result
Students Grade 10 EGSLCE	22	English Grade	Nominal	Result point in EGSLCE
	23	Math Grade	Nominal	Result point in EGSLCE
	24	Physics Grade	Nominal	Result point in EGSLCE
	25	Chemistry Grade	Nominal	Result point in EGSLCE
	26	Biology Grade	Nominal	Result point in EGSLCE
	27	Civics Grade	Nominal	Result point in EGSLCE
	28	Geography Grade	Nominal	Result point in EGSLCE
	29	History Grade	Nominal	Result point in EGSLCE
	30	EGSLCE GPA	Nominal	Total point in EGSLCE
Other factor	31	School Type	Nominal	Government or private
	32	Field of Study	Nominal	Field of study assigned

3 Result and Discussion

3.1 Experimentation

To discover knowledge from the preprocessed data for predicting field of study, two data mining classification algorithms have been experimented. Based on the hybrid model used in this study, after preparation and preprocessing of the data, the next step is mining or modeling process followed by model evaluation. A data set with a total of 9364 records is used for training and testing the predictive model constructed in this study.

3.2 Attribute Selection

The importance of attributes in field of study predictive modeling is checked by maximum gain ratio using WEKA ranking optimal attributes. Evaluating on all training data, attributes selection has been involved through all possible combinations of attributes in the data to find which subset of attributes works best for field of study prediction. To do this, determining CfSubsetEval method is used to assign a value to each subset of attribute by searching best first method technique in WEKA. With these regards, attributes selected using best first techniques in WEKA are SEX, SCHOOL COMPLETED, FAMILY INTEREST and STUD INTEREST.

The researchers also experiments WEKA attribute selection method using GainratioAttributeEval search method and WEKA attribute selection ranker method. The result of the ranker based on information gain ratio, Best first method gives 4 attributes whereas ranker method gives 31 attributes. Thus, before preceding the experiments the researcher evaluates the accuracy of the model in WEKA with the selected attribute in both best first and ranker methods. The experimental result shows that the ranker method is better than the best first method. Thus, the subsequent experiments, regarding selected attributes are performed based on attributes selected in ranker method as show in Table 2.

3.3 Experimental Setup

For experimentation, two classification algorithms, PART decision list and J48 decision tree, have been employed by considering different parameters for model building such as pruning, unpruning and testing model performance with selected attributes and all attributes in the sample data sets. The importance of attributes in field of study predictive modeling is checked by maximum gain ratio, GainratioAttributeEval search method and WEKA (Waikato Environmental for Knowledge Analysis) attribute selection ranker method.

The experiments are conducted on two setups with pruned and unpruned parameters, both contains all the attributes. Thus, the models are compared using different performance measures like accuracy, TN Rate, TP Rate, F-Measure, ROC Area and execution time as shown in Table 3.

Table 2. Attributes ranked with information gain.

Rank	Attribute	Weight	Representation Descriptions
11	STUD INTEREST	0.206508	STUDENT INTEREST (SS, NS)
7	FAMILY INTEREST	0.140479	FAMILY INTEREST(SS, NS)
4	SCHOOL COMPLETED	0.10635	SCHOOL COMPLETED (AA, OR, DB, AM)
10	FAMILY ED UBG	0.057187	FAMILY EDUCATIONAL BACKGROUND
29	EGSLCE GEO	0.056475	EGSLCE GEOGRAPHY (A, B, C, D, F)
19	SEC SCH BIO	0.04726	SECONDARY SCHOOL BIOLOGY
27	EGSLCE BIO	0.045974	EGSLCE BIOLOGY (A, B, C, D, F)
16	SEC SCH MATH	0.038681	SECONDARY SCHOOL MATHEMATICS
1	SEX	0.037591	GENDER (F, M)
20	SEC SCH CIV	0.03473	SECONDARY SCHOOL CIVICS
12	STUD SPECIAL SKIL	0.03284	STUDENTS SPECIAL SKILL
18	SEC SCH CHEM	0.027585	SECONDARY SCHOOL CHEMISTRY
5	SCHOOL TYPE	0.026087	GOVERNMENT OR PRIVATE SCHOOL
8	LANG PROBLEM	0.021487	LANGUAGE PROBLEM (YES, NO)
21	SEC SCH GEO	0.017276	SECONDARY SCHOOL GEOGRAPHY
26	EGSLCE CHEM	0.016969	EGSLCE CHEMISTRY (A, B, C, D, F)
31	EGSLCE CGPA	0.014092	GSLCE COMMUTATIVE AVERAGE
24	EGSLCE MATH	0.013919	EGSLCE MATHEMATICS (A, B, C, D, F)
22	SEC SCH HIS	0.013496	SECONDARY SCHOOL HISTORY
28	EGSLCE CIV	0.011518	EGSLCE CIVICS (A, B, C, D, F)
17	SEC SCH PHY	0.009404	SECONDARY SCHOOL PHYSICS
14	CURRENTLY LIVE	0.00938	CURRENTY LIVE WITH (ALON, WITH F)
15	SEC SCH ENG	0.009304	SECONDARY SCHOOL ENGLISH
30	EGSLCE HIS	0.008649	EGSLCE HISTORY (A, B, C, D, F)
25	EGSLCE PHY	0.008468	EGSLCE PHYSICS (A, B, C, D, F)
9	FAMILY JOB	0.006388	FAMILY JOB (PRVT, GOVT, FARM)
23	EGSLCE ENG	0.005362	EGSLCE ENGLISH (A, B, C, D, F)
3	LIVING BG	0.004409	LIVING BACKGROUND (CITY, RURAL)
13	FAMILY CLASS	0.000341	FAMILY CLASS (MID, HIGH, LOW)
2	AGE	0.000305	AGE (LOW, MID, HIGH)
6	DISABLITY	0.000142	DISABILITY (YES, NO)

As presented in Table 3, all classification algorithms performed nearly equal. The highest accuracy is 98.00% while the lowest accuracy score is 97.8%. Unpruned PART rule induction classifier which was implemented on all attributes achieved the highest accuracy (98.00%) while an pruned PART tree classifier which was implemented on all attributes came out to be the second with classification accuracy of 97.99%.

A model built from pruned PART rule induction with all attributes scored the highest TP Rate while the other model built scored the lowest TP Rate.

Table 3. Performance summary of J48 and PART classification algorithms.

Model	Accuracy	TP Rate	TN Rate	F-Measure	ROC Area	Time (Sec)
j48 unpruned	97.78%	97.8%	97.7%	97.8%	98.3%	0.11 s
j48 with pruned	97.88%	97.8%	98.2%	98%	98.4%	0.14 s
PART with unpruned parameters	98.00%	97.8%	98.2%	98%	98%	0.45 s
PART with pruned parameters	97.99%	98.2%	97.8%	98%	98.7%	0.38 s

It was easier for the unpruned rule induction with all attributes to identify negative cases correctly compared to the other models. In contrast, a model built from others with all attributes straggled a little bit to identify negative case correctly compared to the Unpruned PART rule induction classifier.

3.4 Performance Analysis of the Selected Model

Based on the experimental result shown in Table 4 PART rule induction classification algorithm with unpruned all attributes is selected as the best predictive model for this study and the rules generated by this model are further used for developing the intended knowledge-based systems.

Table 4. Performance analysis of the selected PART rule induction algorithm.

Model	Rule	Accuracy	Confisun Matrix
PART unpruned with all attributes	72	98	a b classified as 4575 102 a= SOCIAL SCIENCE 85 4602 b = NATURAL SCIENCE
PART pruned with all attributes	106	97.99	a b classified as 4592 85 a= SOCIAL SCIENCE 103 4584 b = NATURAL SCIENCE

Experimenting PART decision list rule induction classification algorithm with different parameters, PART unpruned decision list model is selected with a performance of 98.00% of accuracy.

Table 4 shows that the model built with PART unpruned decision list with all attributes classified 9177 (98.003%) of instances correctly while 187 (1.997%) of the instances were classified incorrectly. Moreover, the model identified 4575 of social science instances correctly out of 4677 instances that were social science and the remaining 102 instances were classified incorrectly as natural science while they are actually social science. The model also identified 4602 natural science instances correctly out of 4687 instances that are natural science and the remaining 85 instances were identified to social science while they are actually natural science. As a result, the overall accuracy rate of the model is highly successful.

3.5 Error Rate (Misclassification) of the Selected Model

Error rates are used to make actual decisions about which parts of the tree to replace or raise. One of the methods in the knowledge discovery tasks is to evaluate the performance of the system about how correctly the model classifies tuples into different labeled classes. Though the predictive performance of the selected model (PART) is promising 98.003% of accuracy for field of study prediction, the model commits 1.997% of the cases to classify wrongly (misclassified) to some other class. The learning algorithm made bias to the majority class (social science) in this case in all the modes the predictive performance in identifying True Positive or social science cases of model is higher than identifying True Negative or natural science cases. This is because there is imbalance between the two classes in the data set. Consequently, the model tends to misclassify instances to some other class. The other reason for misclassification is due to the fact that field of study is based on the values of other attributes i.e. taking the similarity of the other attributes as a predominant predictive values.

3.6 Rule Extraction

From the entire models that are built, the model developed with unpruned all PART rule induction classifier was selected as the best model for this study. The rules provided by PART models can be easily assimilated by human without any difficulty. PART decision list generated 72 significant rules that are useful for field of study prediction. Thus, the researcher discusses with domain experts about the significance of the rules. Therefore, all 72 best rules are selected that cover most of the data points in the study in consulting with domain experts. Some of the interesting rules generated by PART unpruned tree model with all attributes are presented below.

RULE 1: IF STUD INTEREST = SS AND SEC SCH CIV = C AND FAMILY CLASS = MID AND EGSLCE GEO = B AND SEC SCH BIO = C THEN FIELD OF STUDY = SOCIAL SCIENCE (249.0) The first rule selected from the rules generated by the PART algorithm gave a correct result for all 149 cases that it covers; thus, its success is 100%. This rule is a very strong rule for predicting students field of study. The domain expert accepted this rule.

RULE 2: IF STUD INTEREST = SS AND SEC SCH CIV = B AND EGSLCE GEO = B AND FAMILY INTEREST = SS AND SCHOOL COMPLETED = AM AND FAMILY CLASS = MID AND SEC SCH MATH = C AND EGSLCE BIO = B THEN FIELD OF STUDY = SOCIAL SCIENCE (800.0) This rule selected from the rules generated by the PART algorithm gave a correct result for all 800 cases that it covers; thus, its success is 100%. This rule is a very strong rule for predicting students field of study. The domain expert accepted this rule.

RULE 3: IF STUD INTEREST = NS AND SCHOOL COMPLETED = AM AND EGSLCE BIO = A AND FAMILY INTEREST = NS AND FAMILY CLASS = MID AND STUD SPECIAL SKIL = STEM THEN FIELD OF STUDY= NATURAL SCIENCE (939.0) This rule selected from the rules generated by the PART algorithm gave a correct result for all 939 cases that it covers; thus, its success is 100%. This rule is a very strong rule for predicting students field of study. The domain expert accepted this rule.

RULE 4: IF STUD INTEREST = SS AND STUD SPECIAL SKIL = ART AND EGSLCE MATH = C AND EGSLCE GEO = B THEN FIELD OF STUDY = SOCIAL SCIENCE (718.0) This rule selected from the rules generated by the PART algorithm gave a correct result for all 718 cases that it covers; thus, its success is 100%. This rule is a very strong rule for predicting students field of study. The domain expert accepted this rule.

RULE 5: IF STUD SPECIAL SKIL = STEM THEN FIELD OF STUDY = NATURAL SCIENCE (3.0) This rule selected from the rules generated by the PART algorithm gave a correct result for all 3 cases that it covers; thus, its success is 100%. This rule is a very strong rule for predicting students field of study. The domain expert accepted this rule.

3.7 Mapping Predictive Model to Knowledge-Based Systems

In this study Java programming has played a great role in integrating the data mining to knowledge-based systems. In order to explore alternative program representations a parser that translates a Java program from a text file representation to a Prolog representation is implemented. A parser transforms a flat file to a tree representation, the parse tree. Therefore, the researcher develop a tool using Java Programming Language and runs the Java code against the Java source file to produce file containing PART decision list rules and Prolog file that contains facts and rules used by the knowledge bases.

Finally, the hidden knowledge discovered using data mining techniques (PART decision list classification algorithm) is being rules and facts used in building knowledge-based systems that predict students field of study. Then the knowledge based systems is used for prediction field of study by inferring from the inference engine of the knowledge based system. Moreover, in addition to resealing mechanisms, the developed knowledge based systems could have a self-learning capability that new rules can be updated automatically.

4 Conclusions and Future Work

Joining to preparatory school to study fields with students background performance issue and giving emphasis to factors that affect secondary school students are given less emphasis throughout the Ethiopian education systems. The major challenge for field of study selection with students performance in the country is lack of skill. In Ethiopia, due to wrong field of study selection which may lead to students dropout, the number of tertiary students and their parents are disproportionate. Due to this the placement of students to a given field of study is greatly unfair. Lacks of knowledge among secondary school students, the allocation of budgets for preparatory school, and the lack of awareness about field of study selection based on students performance are the other challenges that become obstacle to address the problem.

In this study the hybrid methodology was employed. In order to discover knowledge from the data collected from selected secondary schools of Oromia, Amhara, Debub and Addis Ababa, a total of 9364 students record from 2014–2015 years were taken for both classes (natural science and social science) using stratified simple random sampling technique. The findings noticeably show that the PART decision list algorithm is selected based on its highest accuracy of 98.00% and the discovered knowledge using this algorithm has been automatically connected to knowledge-based systems using Java. Based on which it provides the recommended field of study together with the probable reasons being assigned to the recommended field of study.

As a result, the following future works are given based on the opening opportunities and uncovered areas by this study.

- This study only gives advice for the best one field of study. For future work researchers would develop and improve this system by advising more than one field of study selection.
- This study is limited to advising the students side during their field of study selection. Therefore, further investigation should be done in the integration of both students side and Ministry of Education side by considering all the available criteria such as, disability, sex as affirmative action, availability of fields and developing regions.
- This study is attempts to integrate data mining discovered knowledge to rule based knowledge-based systems. But to enhance the performance of the proposed knowledge based systems, the integrated approaches should be investigated in which the case-based reasoning with rule based systems is incorporated.
- To reach the advising in every ones hand there is a need to incorporate with web based mobile application for easy accessibility of the system using mobile internet.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (61502336, 61672377), the National Key Research and Development Program of China (2016YFB1000603), and the Key Technology Research and Development Program of Tianjin (16YFZCGX00210).

References

1. Biazen, C.: Application of case based recommender system to advise students in field of study selection at higher education in Ethiopia. Addis Ababa University, Addis Ababa, Ethiopia (2014)
2. Micheline, K.: Data mining: concepts and techniques. University of Illinois at Urbana-Champaign, San Fransisco (2006)
3. Mihaela, O.: On the use of data-mining techniques in knowledge-based systems. Econ. Inform. **4**, 21–41 (2006). University Petroleum-Gas of Ploieşti, Department of Informatics
4. Two Crows Corporation Introduction to Data Mining and Knowledge Discovery. http://www.twocrows.com. Accessed 25 Dec 2017
5. Pedrycz, F., Kurgan, K.G.: Data Mining for Knowledge Discovery. Laxmi, New Delhi (2007)
6. Fong, S., Biuk-Aghai, R.: An automated admission recommender system for secondary school student. In: The 6th International Conference on Information Technology, pp. 14–24 (2009)
7. Getachew, F.: Higher education entrance student placement processing and retrieval system. Addis Ababa University, Addis Ababa, Ethiopia (2008)
8. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann Publishers, Francisco (2006)

Template-Based SPARQL Query and Visualization on Knowledge Graphs

Xin Wang[1,2](✉), Yueqi Xin[1], and Qiang Xu[1]

[1] School of Computer Science and Technology, Tianjin University, Tianjin, China
{wangx,xinyueqi,xuqiang3}@tju.edu.cn
[2] Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin, China

Abstract. With the popularity of Linked Open Data, a large amount of RDF data have been published and developed in the form of knowledge graphs, which can be publicly accessible via SPARQL endpoints. The efficiency of SPARQL querying on large-scale knowledge graphs has attracted increasing research efforts. In this paper, we propose a template-based query approach, which involves temporal, spatial, and domain-specific constraints to focus on certain resources of interest. Furthermore, query results which include a set of RDF triples are visualized in graph format to display entities and relationships in a user-friendly manner. We also analyze the visualized graph with ranking, partitioning, filtering, and statistics. Various template-based queries are designed and evaluated on the knowledge graph of DBpedia. It can be observed that template-based queries with temporal-spatial and domain-specific constraints can effectively facilitate users to obtain target answers by filtering out irrelevant information.

Keywords: Template query · Temporal-spatial · Visualization
SPARQL

1 Introduction

With the popularity of *Linked Data*, a series of systematic methods to organize and publish RDF graphs have been developed [6], which aim to build large-scale *knowledge graphs*. As a flexible graph-like data model, an RDF graph is a set of triples, where each triple, consisting of a subject, predicate, and object, can be viewed as an edge in a directed graph from a subject to an object with the predicate as the edge label. SPARQL is the standard query language, endorsed by W3C, to retrieve data from RDF graphs. Since RDF has been gradually recognized as a major representation format by the knowledge graph community, in recent years, increasing importance has been attached to SPARQL for querying large-scale knowledge graphs more effectively.

To provide effective SPARQL query experience, there have been some research efforts on large-scale knowledge graphs. With the interface of TriniT [15], users need to input a complete SPARQL query, which is difficult

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 184–200, 2018.
https://doi.org/10.1007/978-3-319-91455-8_17

for end users. On the other hand, in RelFinder [8,13], users need to provide two resources in the interface. Then, from a starting node, RelFinder explores its neighboring nodes and properties, which form the edges in the paths between the two user-specified resources. Meanwhile, Fusion [1] is also designed to offer discovery of relationships between two given resources. As a graph-based query method, NAGA [11] provides a user interface and focuses on a novel scoring model to rank results. Shekarpour et al. [14] propose basic graph pattern templates to generate SPARQL queries, where users merely need to provide some words of interest. However, the above methods are designed for professional programmers in most cases and the query results are returned in form of triples or paths, which are not visible and usable for users.

In this paper, we propose an approach to evaluating template-based SPARQL queries, which are defined as general SPARQL queries with Basic Graph Patterns (BGPs) and FILTER clauses. End users just need to provide several specific keywords or values to replace the *placeholders* in BGPs or FILTER clauses without knowing the syntax and semantics of SPARQL. Since the attributes about time and space are common and essential for the resources in knowledge graphs, we define the *basic template query* by adding temporal-spatial constraints that are provided by users. However, in the real world, users' requests not only focus on the temporal and spatial attributes of resources but also other various domain-specific properties. If users need to obtain precise relationships, they can choose the *refined template query*, which replaces the variables with certain values or adds triple patterns with extra constraints based on the basic template query.

Despite the proliferation of knowledge graphs, there still exist a number of obstacles, which hinder the large-scale deployment of knowledge graphs [2]. In general, for end users, the query result on knowledge graphs is a set of triples that are not yet sufficiently visible and usable. The *visualization* for knowledge graphs is commonly displayed in form of a labeled graph, such as the Paged Graph Visualization [4]. It turns out that the efficient innate human capabilities can be inspired to perceive and process data when knowledge graphs are presented visually [10]. Therefore, we convert the result of the template-based query into a series of nodes and edges and display them in graph format using the following three steps: (1) the result is loaded into the R tool; (2) *igraph* package is applied to construct an adjacency graph of the result in R; and (3) the adjacency graph can be visualized as a labeled graph in Gephi. We designed and evaluated 8 template-based queries on the knowledge graph of DBpedia. Moreover, We demonstrate these queries in several typical case studies via a SPARQL endpoint on a single machine.

Our main contributions include: (1) We propose the basic template SPARQL query with temporal-spatial constraints, where the keywords or values are provided by users to extract entities and relationships of interest. (2) Further we design the refined template query, which is defined by adding constraints with certain domain-specific values based on the basic one. (3) Various queries are designed and conducted as different case studies on knowledge graphs, such as the DBpedia dataset. Moreover, we realize a user-friendly visualization of each query result in graph format.

The rest of the paper is organized as follows. We discuss the related work in the areas of SPARQL query and visualization on knowledge graphs in Sect. 2. Section 3 provides the fundamental definitions of background knowledge. In Sect. 4, we describe in detail the template-based query. We also present the universal procedure of visualization in Sect. 5. Queries based on the basic and refined templates are evaluated in case studies in Sect. 6, and we conclude in Sect. 7.

2 Related Work

The existing query and visualization methods on knowledge graphs can be classified into the following four categories:

Language-Based Query. TriniT search engine [15] provides a user interface for querying, where users need to input a complete SPARQL query with knowing the syntax of SPARQL. Elbassuoni et al. [5] present a structured query mechanism on RDF graphs, which shows the inter-relationships between entities based on a language model. Despite with rich expressiveness, the above language-based query methods are designed for professional programmers in most cases. Thus, it is difficult for end users to use it effectively.

Keyword-Based Query. Heim et al. [8] propose an approach, called RelFinder, to searching the relationships between two user-specified nodes and displaying all the edges between the two nodes as a graph. Users can choose a starting node and incrementally explore a knowledge graph. The found resources are visualized as nodes connected by the edges labeled with the relationships. In addition, Lohmann et al. [13] present an approach with relationships filtering in four dimensions based on RelFinder. Fusion [1] implements a path discovery algorithm, which can find a path represented in form of a triple between two known resources given by users in a Web interface. However, the above methods only focus on the paths satisfying between the two given nodes labeled with keywords, which result in ignoring the global information that is vital for data statistics and analysis.

Graph-Based Query. NAGA [11] is a semantic search engine, which is built based on a knowledge base consisting of millions of entities and relationships. It presents a graph-based query language that allows the formulation of queries with semantic information, which can be more expressive than those standard keyword-based search approaches. The query result of NAGA is ranked using a scoring model, while it is not visualized in form of a graph and not intuitive for end users.

Template-Based Query. Shekarpour et al. [14] propose a set of predefined basic graph pattern templates to generate SPARQL queries with the user-supplied keywords. Users just need to provide several keywords, then the corresponding SPARQL query can be generated and the query result can be returned. LESS [2] presents a language, called LESS Template Language (LeTL), which can define arbitrary text-based output representations and support the integration of information. It provides a Web interface for users to edit the template with user-defined parameters, then returns the query result for users, whereas it cannot display the result in form of a graph.

Actually, most of the above methods simply automatically display the results in the Web applications without graphics visualization. Unlike the above methods, we combine keyword-based and template-based queries with a balance between flexibility and expressiveness to design the basic and refined template queries. In our method, the query result is displayed as a graph with interactive features and filter options considering the global information in different granularity and dimensions.

3 Preliminaries

In this section, we introduce the definitions of relevant background knowledge.

RDF data is a collection of triples denoted as (s, p, o), which states that the resource s has a relationship p to the resource o, where s is called the subject, p the predicate (or property), and o the object, which can be formally defined as follows:

Definition 1 (RDF graph). *Let U and L be the disjoint infinite sets of URIs and literals, respectively. A tuple $(s, p, o) \in U \times U \times (U \cup L)$ is called an* RDF triple*. A finite set of RDF triples is denoted as $G = (V, E, \Sigma)$, called an RDF graph, where V is a set of vertices that correspond to all subjects and objects; $E \subseteq V \times V$ is a set of directed edges that correspond to all triples; and Σ is a set of edge labels.*

SPARQL is the standard RDF query language, in which Basic Graph Pattern (BGP) queries are fundamental building blocks [7]. The BGP queries can be easily extended to general SPARQL queries with FILTER, UNION, and OPTIONAL.

Definition 2 (Basic graph pattern (BGP)). *Assume there exists an infinite set Var of variables disjoint from U and L, and every element in Var starts with the character ? conventionally, e.g., $?v \in Var$. A triple $(s, p, o) \in (Var \cup U) \times (Var \cup U) \times (Var \cup U \cup L)$ is called a* triple pattern. Basic graph pattern (BGP) *is denoted as a finite set of* triple patterns. *For a triple pattern t, let $vars(t)$ be the set of variables occurring in t.*

In order to help users query on knowledge graphs without knowing the syntax and semantics of SPARQL, we design the *basic template query*. First, we predefine a series of *placeholders*, denoted by $K = \{K_1, K_2, \ldots, K_n\}$, where $K_i \in \mathcal{P}(U \cup L)$ and K_i denotes the resources that belong to some specific domains. Then, as a placeholder, each element $k_i \in K_j$ can denote a subject, a predicates, or an object. For example, given a certain triple pattern (s, k_i, o), users can specify $k_i \in K_j$ to denote a predicate in the triple pattern. In particular, we define K_{pt} as a set of predicates for restricting the temporal attributes, such as $K_{pt} = \{$`birthYear,birthDay,...`$\}$, K_{ot} as a set of objects for representing certain temporal values, such as $K_{ot} = \{$`1990,1990-01-01,...`$\}$, K_{ps} as a set of predicates for restricting the spatial attributes, such as $K_{ps} = \{$`nationality,birthPlace,...`$\}$, and K_{ot} as a set of objects for representing certain spatial values, such as $K_{os} = \{$`United_Kingdom, London,...`$\}$. Therefore, we define the *basic template query* as follows:

Definition 3 (Basic template query). *Assume that a SPARQL query, denoted as $Q_t = \{t_1, \ldots, t_n\}$, includes a set of triple patterns and additional FILTER statements. There exist several triple pattern statements, such as, $t_p = (s, k_{pt}, k_{ot})$ satisying $k_{pt} \in K_{pt} \wedge k_{ot} \in K_{ot}$, $t_f = (s, k_{pt}, o)$ satisying $k_{pt} \in K_{pt}$ and o is restricted by FILTER in a range, and $t_s = (s, k_{ps}, k_{os})$ satisying $k_{ps} \in K_{ps} \wedge k_{os} \in K_{os}$. If $(t_p \in Q_t \vee t_f \in Q_t) \wedge t_s \in Q_t$, then Q_t is a* basic template query*. The placeholders in Q_t can be replaced by proper values specified by users.*

The properties related to temporal-spatial attributes of the resources in knowledge graphs are general in most cases. For Q_t, users just need to provide the values to replace the placeholders. However, in the real world, users'requests not only focus on the temporal-spatial attributes, but also other attributes in different domains. If users would like to obtain more specific or detailed information, they can choose the *refined template query*, which is defined as follows.

Definition 4 (Refinement relation). *Let S_q and S_r denote two sets of basic template queries, a binary relation from S_q to S_r, denoted as $R \subseteq S_q \times S_r$, is called a* refinement relation *if and only if $\forall (Q_t, Q_r) \in R$ the following conditions hold, for a triple pattern $t \in Q_t$: (1) $t \in Q_r$ or $t \notin Q_r \wedge \exists\ t_r \in Q_r \wedge vars(t_r) \subset vars(t)$; (2) $vars(Q_r) \subset vars(Q_t)$.*

Definition 5 (Refined template query). *Given a basic template query $Q_t = \{t_1, \ldots, t_n\}$, where t_i is a triple pattern or a FILTER statement, we design a query $Q_r = \{t_1, \ldots, t_n, t_{n+1}, \ldots, t_{n+m}\}$, where t_{n+i} $(i \geq 1)$ denotes a triple pattern in general* SPARQL*. If Q_t and Q_r satisfy a refinement relation, i.e., $(Q_t, Q_r) \in R$, then Q_r is called a* refined template query *w.r.t. Q_t.*

Obviously, for a *refinement* relation R, $\forall (Q_t, Q_r) \in R$, the result set of Q_r is a subset of that of Q_t [12]. For instance, in Fig. 1, the SPARQL expressions highlighted in orange represent temporal-spatial constraints, in gray represent extra refined triple patterns. The query result is a set of triples, i.e., a series of subjects, predicates, and objects, which need to be visualized explicitly. To this end, we visualize these triples in graph format.

SELECT *	**SELECT** *
WHERE {	**WHERE**{
`?x ?z ?m .`	`?x type MusicalArtist .`
`?x ?u ?y .`	`?x associatedMusicalArtist ?y .`
?x k_{pt} k_{ot} .	`?x birthYear 1989 .`
?x k_{ps} k_{os} .	`?x birthPlace United_Kingdom .`
`?y ?v ?n .` }	`?y type Person .` }

Fig. 1. Examples of the basic and refined template queries

Definition 6 (Visualization of RDF triples)**.** *We define the visualization of a set of RDF triples as an labeled undirected graph $G = (V, E)$, called the* visualized graph*. Then, for $\forall$ a triple (s, p, o), the subject s (or the object o) denotes a vertex $v \in V$ (or $v' \in V$) labeled with s (or o), and the predicate p denotes an undirected edge between v and v'. The nodes are sorted in different colors and sizes and the graph is shown in proper layouts.*

4 Template-Based Query

In this section, we describe two template-based queries in detail. Moreover, we present how to evaluate the queries to obtain the meaningful target relationships and information of rich semantics.

4.1 Basic Template Query

There exist several predicates that are restricted to the sets K_{pt} and K_{ps} in the basic template query. The basic template query applies temporal and spatial constraints to the resources, which contributes to an important influence on the visualized graph.

Given a basic template query Q_t, users need to provide k_{ot} and k_{os} to complete the query and execute it against a SPARQL endpoint. Since the predicates in K_{pt} and K_{ps} are determined by the knowledge graph, we can evaluate the basic template query on a knowledge graph in the real world, such as DBpedia. For example, when asking the query "to search the one that belongs to `dbo:Preson` and return the person and his related attributes", two triple patterns t_1 = (`?x rdf:type dbo:Person`) and t_2 = (`?x ?p ?y`) can return the target answers. In basic template query, we add temporal and spatial constraints based on t_1, which is shown in template Q_t. It aims to find all the resources that have the property `rdf:type` with `dbo:Person`, `dbo:nationality` with k_{os}, and `dbo:birthYear` with the value that is larger than k_{ot}. When we increase (or reduce) the range of k_{os} or k_{ot}, the number of the results can change dramatically, which can be clearly observed in the visualized graph.

```
Template $Q_t$:
PREFIX rdf: <http://www.w3.org/1992/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX db: <http://dbpedia.org/resource/>
SELECT ?x ?y
WHERE {
   ?x rdf:type dbo:Person.
   ?x ?p ?y.
   ?x $k_{ps}$ (e.g., dbo:nationality) $k_{os}$ (e.g., db:United_Kingdom).
   ?x $k_{pt}$ (e.g., dbo:birthYear) ?birthYear .
   FILTER(?birthYear >= $k_{ot}$ (e.g., 1990)) .
}
```

In this paper, we mainly focus on the resources and their relationships involving persons, i.e., social networks. We also consider the resources in other domains to reveal the meaningful relationships in the real world. When the domain varies, the corresponding temporal and spatial attributes also change.

4.2 Refined Template Query

In the basic template query, we only add temporal and spatial constraints. However, the resources in knowledge graphs have covered various domains and the query result is too large to be analyzed directly due to the massive knowledge graphs. We propose the refined template query to specify and restrict the resources further to search more meaningful relationships. For example, we specify ?p in t_2 = (?x ?p ?y) of Q_t as dbo:parent and add new triple patterns (?x ?z ?m), (?y ?v ?n), ..., to restrict ?x and ?y, where ?z, ?m, ?v, ?n, etc. are all specified by users.

```
Template $Q_r$:
SELECT ?x ?y
WHERE {
   ?x rdf:type dbo:Person.
   ?x dbo:parent ?y.
   ?x $k_{ps}$ (e.g., dbo:nationality) $k_{os}$ (e.g., db:United_Kingdom).
   ?x $k_{pt}$ (e.g., dbo:birthYear) ?birthYear .
   FILTER(?birthYear >= $k_{ot}$ (e.g., 1990)) .
   ?x ?z ?m .
   ?y ?v ?n .
   ...
}
```

Obviously, there exists a refinement relation R from Q_t to Q_r, where the result set of Q_r is a subset of that of Q_t.

5 Template-Based Visualization

In this paper, the result of the template-based query is a set of RDF triples. Therefore, we can transform these RDF triples into an undirected labeled graph.

As a statistical analysis software, R combines statistical analysis and graph visualization well. The *igraph* package in R can easily convert the result into an adjacent graph in *GraphML* format. Gephi is a very powerful software for processing graphs with many functions, such as sorting, partitioning, statistics, filtering, layout, and so on. To this end, we process the query results with R and realize visualization of RDF triples in Gephi.

5.1 General Data Transform Algorithm

Given a template-based query Q, we execute the query in R against a specified SPARQL endpoint. Once receiving the result, we transform it to an adjacency graph in Algorithm 1, which includes the general steps of visualization in R.

Algorithm 1. `visualizeInR`

```
   Input  : The template-based query $Q$.
   Output: A visualized graph.
1  $R_e \leftarrow$ the result of $Q$ against a specified SPARQL endpoint;
2  Sort $R_e$ as a table $T_r$;
3  Select two columns $\overrightarrow{x}$, $\overrightarrow{y}$ in $T_r$;
4  $m, n \leftarrow$ the number of values in $\overrightarrow{x}$, $\overrightarrow{y}$, respectively;
5  $M_{xy} \leftarrow$ construct a matrix with $m$ rows and $n$ columns;
6  if i-th value in $\overrightarrow{x}$ relates to j-th value in $\overrightarrow{y}$ then
7      $M_{xy}[i][j] = 1$;
8  else $M_{xy}[i][j] = 0$;
9  if visualizing a direct relationship then
10     $xy \leftarrow graph.incidence(M_{xy})$;
11 else if visualizing an indirect relationship then
12     $M_{xy} \leftarrow M_{xy} * M_{xy}$;
13     $diag(M_{xy}) \leftarrow 0$;
14     $xy \leftarrow graph.incidence(M_{xy})$;
15 Attach the labels to the corresponding nodes in $xy$;
16 return xy in GraphML format;
```

The query result in R includes several columns of different entities with certain attributes. We select two columns of entities to construct an adjacency matrix (lines 2–8), which can be transformed into an adjacency graph. There are two cases: (1) when visualizing a direct relationship, we transform the adjacency matrix into an adjacency graph directly (lines 9–10); (2) when visualizing an indirect relationship, we conduct multiplication with the adjacency matrix itself to build a new matrix, then we transform the new one into an adjacency graph (lines 11–14). For clarity of the visualized graph, we attach the labels to the nodes in the graph. Finally, the output is a labeled graph in GraphML format, which can be further demonstrated in Gephi.

5.2 Visualization and Analysis

Gephi is a common visualization tool, which is mainly used for exploratory data analysis, link analysis, social network analysis, and biological network analysis. As a powerful software for processing graphs, it provides systematic analysis with ranking, partitioning, filtering, and statistics for graph analysis [3]. In this paper, the input of Gephi is an adjacency graph from R, shown in a random layout, which can be sorted and displayed in a proper layout as output to intuitively provide useful information for end users.

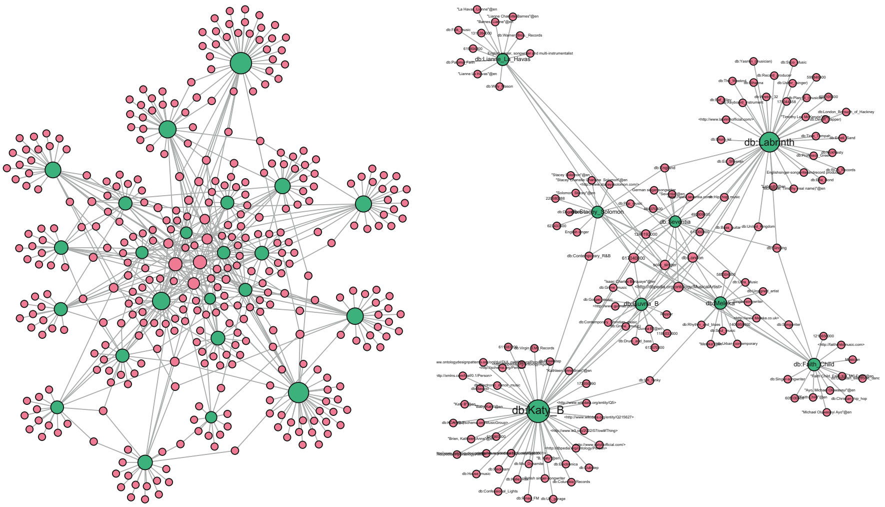
The process of visualization in Gephi can be realized in the following steps. First, we can allocate different colors to the nodes in accordance with the parameters of *betweenness centrality* or *PageRank* and sort the size of each node by its degree. Then we can choose an appropriate layout strategy to display the nodes. There are a variety of layout strategies which consider the gravitational and repulsive forces between each node. Furthermore, to obtain more refined information, we can select nodes or edges with thresholds, ranges, and other properties to filter out irrelevant information.

6 Case Study

The SPARQL queries were executed against a SPARQL endpoint provided by *Virtuoso* on a PC machine. Eight template-based queries were designed and evaluated on the knowledge graph of DBpedia, shown in four case studies. The datasets we employed are four subsets in DBpedia, called `instance_types_en`, `labels_en`, `mappingbased_literals_en`, and `mappingbased_objects_en`, respectively. We also displayed the visualized graph with sorted nodes in terms of colors and sizes. In most cases, a strategy, called *ForceAtlas2*, was chosen to layout the nodes in the graph, which aims to generate a readable shape of the graph [9].

Case Study 1. Without temporal and spatial constraints, the result includes a large number of triples which cannot be analyzed intuitively when visualized. We carried out 4 template-based queries as follows, for selecting the entities and relationships in the real world, in three granularities: (i) template query with temporal and spatial constraints, (ii) refinement by replacing variables with constants, and (iii) refinement by adding extra triple patterns.

When asking the query *"which musical artists have a genre?"*, in template query users just need to provide `MusicalArtist`, `genre`, and temporal and spatial constraints instead of a complete SPARQL query. For example, we select musical artists and corresponding resources who were born in `United_Kingdom` and between 1987 and 1997, shown as Q_1. The result of Q_1 is shown in Fig. 2(a). Although the nodes that represent `dbo:MusicalArtist` are highlighted in green



(a) The result of Q_1 **(b) The result of Q_2**

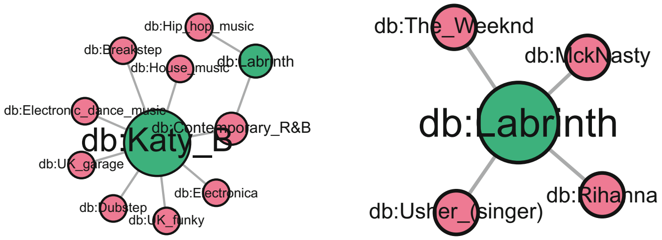
Fig. 2. The visualization of the query results of Q_1 and Q_2 (Color figure online)

and its related nodes are marked in red, the whole results are still difficult to be analyzed intuitively.

```
Q1: SELECT ?x ?y
WHERE {
?x rdf:type dbo:MusicalArtist .    ?x dbo:genre ?z .
?x ?p ?y .       ?x dbo:birthPlace db:United_Kingdom .
?x dbo:birthYear ?birthYear.    FILTER(?birthYear >= 1987) .
FILTER(?birthYear <= 1997) .
}
```

Furthermore, we can narrow the range of spatial and temporal values, such as `London` and 1989, shown as Q_2. As shown in Fig. 2(b), the size of the result decreases in comparison with the result of Q_1. We can observe that the node labeled with `db:Katy_B` and `db:Labrinth` have the larger outdegree than the other nodes. However, temporal and spatial constraints are inadequate, the above results include a certain number of resources that users are not interested in.

```
Q2: SELECT ?x ?y
WHERE {
?x rdf:type dbo:MusicalArtist .    ?x dbo:genre ?z .
?x ?p ?y .              ?x dbo:birthPlace db:London .
?x dbo:birthYear 1989.
}
```



(a) The result of Q_3 **(b) The result of Q_4**

Fig. 3. The visualization of the query results of Q_3 and Q_4

To obtain more specific information, more constraints need to be added based on the basic template query. For example, if users are interested in the musical artists who can play `Bass guitar`, or the people who are associated with the musical artists, then variables can be replaced with constants or extra new triple patterns can be added. Suppose Q_3 be a query that satisfies the refinement relation $R(Q_2, Q_3)$. Based on Q_2, Q_3 is formed by replacing the triple pattern `(?x ?p ?y)` with `(?x dbo:instrument db:Bass_guitar)`, whose result is shown in Fig. 3(a). As we can see, the size of the result of Q_3 decreases significantly compared with that of Q_2. We use Q_4 to select the persons who have the relationship called `associatedMusicalArtist` with musical artists, shown as follows.

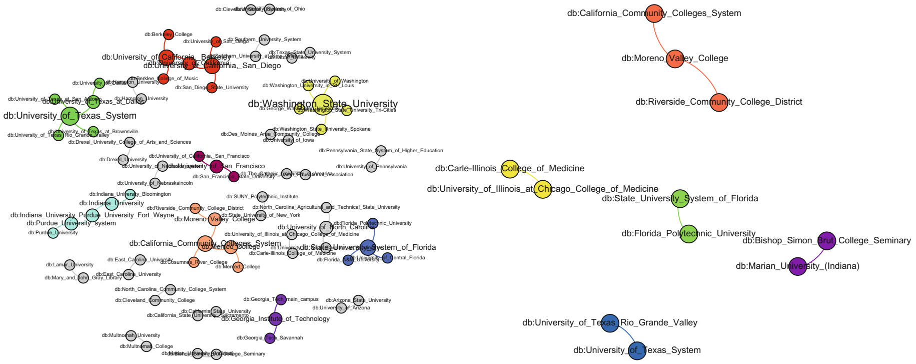
```
Q4: SELECT ?x ?y
WHERE {
?x rdf:type dbo:MusicalArtist .    ?x dbo:genre ?z .
?x dbo:birthPlace db:London .       ?x dbo:birthYear 1989.
?x dbo:associatedBand ?y .          ?y rdf:type dbo:Person .
}
```

The query Q_4 also satisfies $R(Q_2, Q_4)$, whose result is shown in Fig. 3(b). It can be observed that the answers to Q_3 and Q_4 are both subsets of the answers to Q_2.

Case Study 2. When querying the universities that have direct relationships, we add corresponding temporal and spatial constraints, i.e., `dbo:foundingDate` and `dbo:country`. We design Q_5 and vary the value of k_{ot} in temporal dimension, i.e., 1800-01-01 and 2000-01-01 to analyze the query results, as shwon in Fig. 4.

```
Q5: SELECT ?x ?y
WHERE {
?x rdf:type dbo:University .    ?y rdf:type dbo:University .
?x ?p ?y .                  ?x dbo:country db:United_States .
?x dbo:foundingDate ?date .
FILTER(?date >= 1800-01-01 (or 2000-01-01)) .
}
```

Obviously, the number of nodes in the result of Q_5 declines as the range of temporal constraint being narrowed, as shown in Fig. 4(a) and (b). The main relationships between two universities are that one has `dbo:affiliation` with the another. We allocate the color of each node using the *community detection* algorithm and sort the size of each node with the value of the degree. Thus, different communities consisting of several universities are highlighted in different colors. In particular, the less important nodes are marked in gray. It can be observed that all the universities are divided into several groups, which can be analyzed further.


db:Berkeley_College
db:University_of_San_Diego
db:Southern_University_System
db:Texas_State_University_System
db:University_of_California,_Berkeley
db:University_of_California,_San_Diego
db:Berklee_College_of_Music
db:San_Diego_State_University
db:University_of_Washington
db:Washington_University_in_St._Louis
db:Washington_State_University
db:Washington_State_University_Tri-Cities
db:Washington_State_University_Spokane
db:Des_Moines_Area_Community_College
db:University_of_Iowa
db:University_of_Texas_at_San_Antonio
db:University_of_Texas_at_Dallas
db:Hampton_University
db:University_of_Texas_System
db:University_of_Texas_at_Brownsville
db:University_of_Texas_Rio_Grande_Valley
db:Drexel_University_College_of_Arts_and_Sciences
db:Drexel_University
db:University_of_California,_San_Francisco
db:University_of_San_Francisco
db:San_Francisco_State_University
db:University_of_Nebraskaincoln
db:Pennsylvania_State_System_of_Higher_Education
db:University_of_Pennsylvania
db:Indiana_University_Bloomington
db:Indiana_University
db:Indiana_University_Purdue_University_Fort_Wayne
db:Purdue_University_system
db:Purdue_University
db:SUNY_Polytechnic_Institute
db:North_Carolina_Agricultural_and_Technical_State_University
db:State_University_of_New_York
db:Riverside_Community_College_District
db:Moreno_Valley_College
db:University_of_North_Carolina
db:Florida_Polytechnic_University
db:University_of_Illinois_at_Chicago_College_of_Medicine
db:California_Community_Colleges_System
db:Merced_College
db:State_University_System_of_Florida
db:Carle-Illinois_College_of_Medicine
db:Florida_A&M_University
db:University_of_Central_Florida
db:Cosumnes_River_College
db:East_Carolina_University
db:Lamar_University
db:Mary_and_John_Gray_Library
db:North_Carolina_Community_College_System
db:Cleveland_Community_College
db:Georgia_Tech_main_campus
db:Arizona_State_University
db:University_of_Arizona
db:California_State_University
db:California_State_University,_Sacramento
db:Georgia_Institute_of_Technology
db:Georgia_Tech_Savannah
db:Multnomah_University
db:Multnomah_College
db:California_Community_Colleges_System
db:Moreno_Valley_College
db:Riverside_Community_College_District
db:Carle-Illinois_College_of_Medicine
db:University_of_Illinois_at_Chicago_College_of_Medicine
db:State_University_System_of_Florida
db:Florida_Polytechnic_University
db:Bishop_Simon_Bruté_College_Seminary
db:Marian_University_(Indiana)
db:University_of_Texas_Rio_Grande_Valley
db:University_of_Texas_System

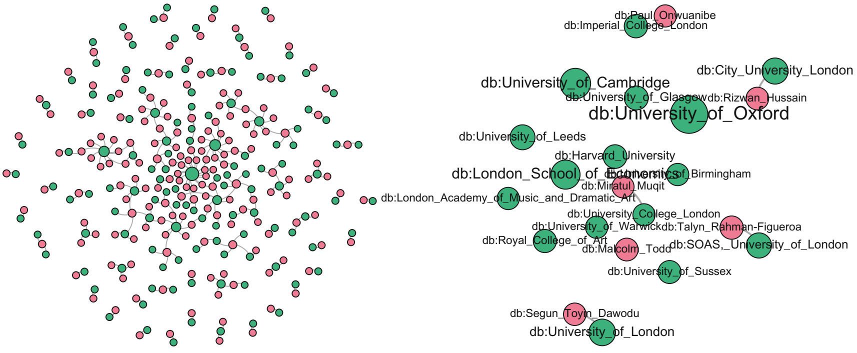
(a) $\geq$ **1800-01-01** **(b)** $\geq$ **2000-01-01**

Fig. 4. The visualization of the query result of Q_5 (Color figure online)

Case Study 3. In Q_5, it searches the universities that have direct relationships, where each node in Fig. 4 represents a university. Now we look for a person and his/her related university. Temporal-spatial constraints are added to the person rather than the university, as shown in Q_6.

```
Q6: SELECT ?x ?y
WHERE{
?x ?p ?y .    ?x dbo:nationality db:United_Kingdom .
?x rdf:type dbo:Person .    ?y rdf:type dbo:University .
?x dbo:birthYear ?birthYear .    FILTER(?birthYear >= 1900) .
}
```

The result of Q_6 contains the persons and the related universities, including 296 nodes and 236 edges in total, among which 169 persons marked in red and 127 universities in green in Fig. 5(a). We allocate color of each node by its kind; rank the size of each node by *PageRank* algorithm in Gephi, where the rank of a node is higher, the size is larger. Based on the result of Q_6, we just display the nodes whose rank are greater than or equal to 0.59, then we obtain several



(a) Persons and universities **(b) The nodes with rank $\geq$ 0.59**

Fig. 5. The visualization of the query result of Q_6 (Color figure online)

significant nodes as shown in Fig. 5(b). Further, the modularity class is applied to allocate the color of each node in order to build 9 main communities consisting of the persons and the related universities in Fig. 6.

In particular, we take advantage of the above direct relationships between the persons and the universities to mine the indirect connections, i.e., the schoolfellow relationship. After using matrix multiplication operation in Algorithm 1 to process the result of Q_6, we can obtain the social relationship between the persons who are studying or working in the same university, as visualized in Fig. 7, which is a complex social network and different from the other visualized graphs. If a person has complex relationships with other persons, then the color of the node that labeled with this person is closer to red. We use the *K-core* algorithm to refine the social network further and employ *between_centrality* to filter out the nodes, as shown in Fig. 8(a) and (b).

Case Study 4. In this subsection, we focus on the relationships that belong to some specific domains. For example, based on Q_1, we change the value of object correspond with to `rdf:type` in the first triple pattern, which can limit the subject to a certain class of people, such as swimmer, writer, soccerplayer, or tennisplayer. Then we design Q_7 to specify the relationship called `dbo:influencedBy`, which is a unique attribute in `dbo:Writer`. Similarly, we desgin Q_8 to search the relationship called `dbo:soccerPlayer` between a soccerplayer and a team. For obtaining specific results, we search the soccerplayers and their teams, who are born after 1800 s and have British citizenship.

The result of Q_7 includes 29 nodes and 18 edges in Fig. 9(a), which means that there are 18 relationships satisfy the conditions. We sort color of each node by the modularity class and rank the size of each node by PageRank. In Fig. 9(a),

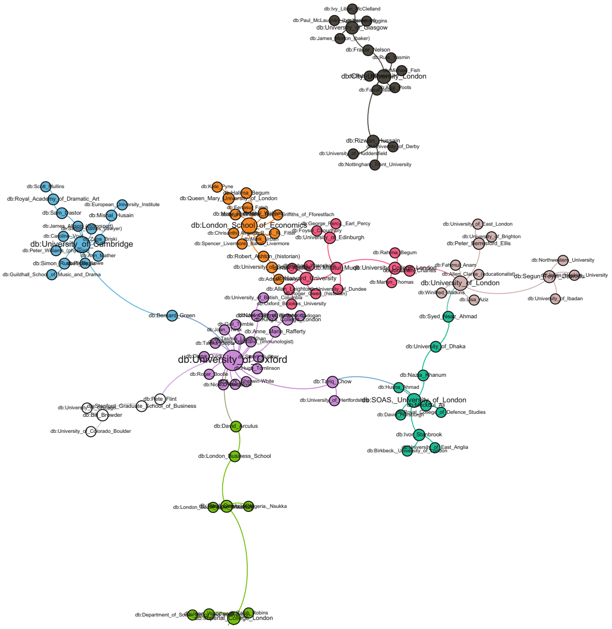

db:Ivy_Lilian_McClelland
db:University_of_Glasgow
db:James_Gordon_Baker)
db:Fraser_Nelson
db:City,_University_of_London
db:Rizwan_Hussain
db:University_of_Derby
db:University_of_Huddersfield
db:Nottingham_Trent_University
db:Scott_Mullins
db:Royal_Academy_of_Dramatic_Art
db:European_University_Institute
db:Sam_Pastor
db:Mishal_Husain
db:University_of_Cambridge
db:Guildhall_School_of_Music_and_Drama
db:Kate_Pyne
db:Queen_Mary_University_of_London
db:London_School_of_Economics
db:Spencer_Livermore,_Baron_Livermore
db:Robert_Ashton_(historian)
db:University_of_Edinburgh
db:Rahima_Begum
db:University_of_East_London
db:University_of_Brighton
db:Peter_Beresford_Ellis
db:Northwestern_University
db:University_College_London
db:University_of_London
db:University_of_Ibadan
db:Syed_Neaz_Ahmad
db:University_of_Dhaka
db:Nazia_Khanum
db:SOAS,_University_of_London
db:Royal_College_of_Defence_Studies
db:Ivor_Stanbrook
db:University_of_East_Anglia
db:Birkbeck,_University_of_London
db:Bernard_Green
db:University_of_Oxford
db:Anne_Marie_Rafferty
db:Hugh_Tomlinson
db:Roger_Bootle
db:Tariq_Chow
db:University_of_Hertfordshire
db:Pete_Flint
db:Stanford_Graduate_School_of_Business
db:Bill_Browder
db:University_of_Colorado_Boulder
db:David_Arculus
db:London_Business_School
db:Department_of_Science
db:Imperial_College_London

Fig. 6. The visualization of the main communities in Q_6 (Color figure online)

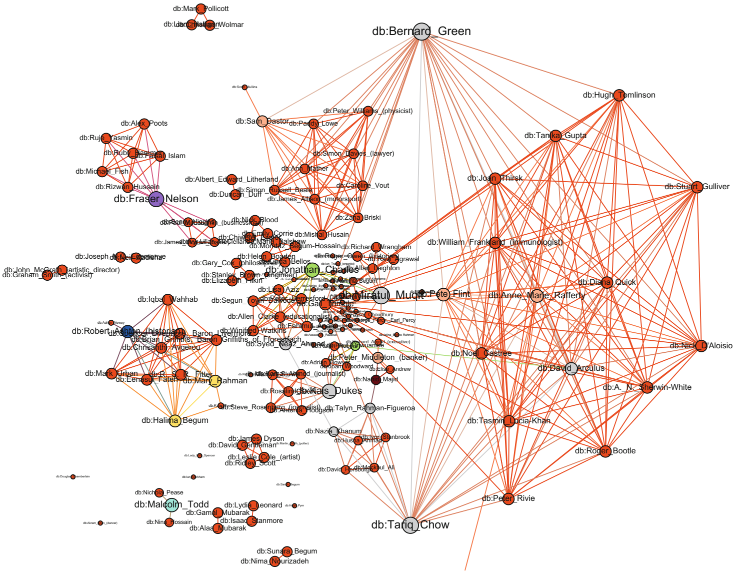
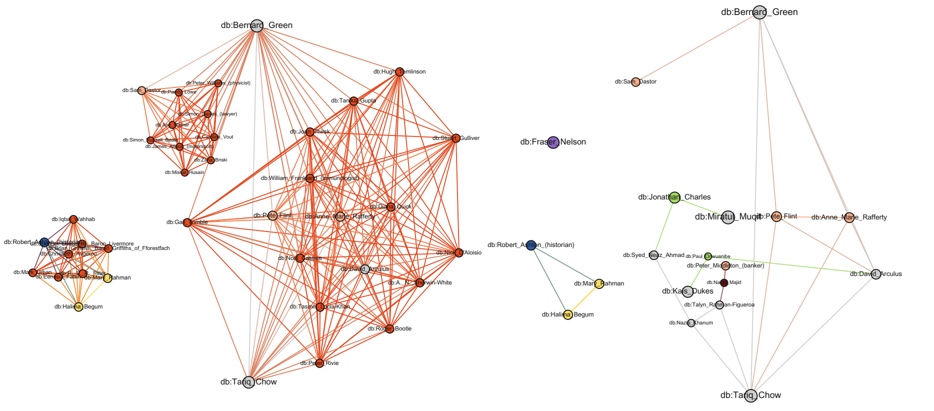
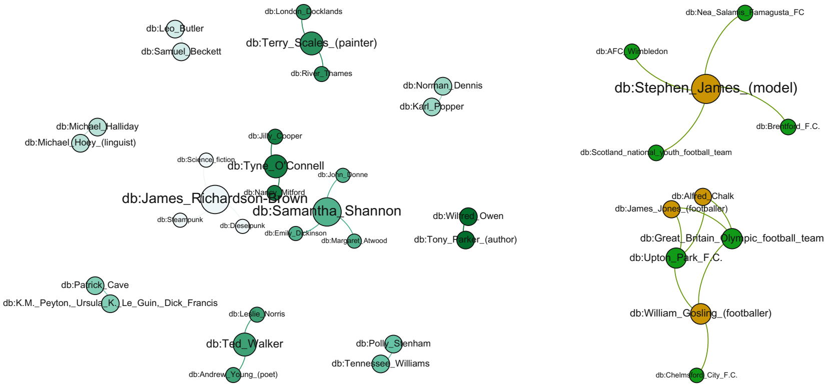


Fig. 7. The visualization of the schoolfellow relationship based on Q_6 (Color figure online)



(a) K-core (k=9) (b) $1.08 \leq$ between_centrality ≤ 1080

Fig. 8. The visualization of the schoolfellow relationship with filtering (Color figure online)



(a) The influencedBy relationship (b) The soccerPlayer relationship

Fig. 9. The relationships of some specific domains in Q_7 and Q_8 (Color figure online)

the size of a node is proportional to the rank of the node. The relationship `soccerPlayer` in Q_8 involves soccer players and teams, which are marked in yellow and green, respectively, as shown in Fig. 9(b). Since the above two relationships belong to the specific domains, the number of result is relatively fewer and it can be easily visualization.

7 Conclusion

We present template-based queries on knowledge graphs, which is a query method based on temporal, spatial, and domain-specific constraints. We mainly propose two template queries, i.e., the basic and refined template queries, to extract valuable information on knowledge graphs, whose results are visualized in undirected labeled graph. Our experimental results are well displayed in graph format for data analysis. With different constraints, knowledge graphs are visualized in different granularity. Our future work includes the visualization of more knowledge graphs and implementation of a more user-friendly interface.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (61572353) and the Natural Science Foundation of Tianjin (17JCYBJC15400).

References

1. Araujo, S., Houben, G.-J., Schwabe, D., Hidders, J.: Fusion – visually exploring and eliciting relationships in linked data. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010. LNCS, vol. 6496, pp. 1–15. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17746-0_1
2. Auer, S., Doehring, R., Dietzold, S.: LESS - template-based syndication and presentation of linked data. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010. LNCS, vol. 6089, pp. 211–224. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13489-0_15
3. Bastian, M., Heymann, S., Jacomy, M., et al.: Gephi: an open source software for exploring and manipulating networks. In: ICWSM, vol. 8, pp. 361–362 (2009)
4. Deligiannidis, L., Kochut, K.J., Sheth, A.P.: RDF data exploration and visualization. In: Proceedings of the ACM First Workshop on CyberInfrastructure: Information Management in eScience, pp. 39–46. ACM (2007)
5. Elbassuoni, S., Ramanath, M., Schenkel, R., Sydow, M., Weikum, G.: Language-model-based ranking for queries on RDF-graphs. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 977–986. ACM (2009)
6. Fernández, J.D., Martínez-Prieto, M.A., Gutierrez, C.: Compact representation of large RDF data sets for publishing and exchange. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010. LNCS, vol. 6496, pp. 193–208. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17746-0_13
7. Harris, S., Seaborne, A., Prudhommeaux, E.: SPARQL 1.1 query language. W3C Recommendation **21**(10) (2013)
8. Heim, P., Hellmann, S., Lehmann, J., Lohmann, S., Stegemann, T.: RelFinder: revealing relationships in RDF knowledge bases. In: Chua, T.-S., Kompatsiaris, Y., Mérialdo, B., Haas, W., Thallinger, G., Bailer, W. (eds.) SAMT 2009. LNCS, vol. 5887, pp. 182–187. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10543-2_21

9. Jacomy, M., Venturini, T., Heymann, S., Bastian, M.: Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. PloS One **9**(6), e98679 (2014)
10. Kapler, T., Wright, W.: Geotime information visualization. Inf. Visual. **4**(2), 136–146 (2005)
11. Kasneci, G., Suchanek, F.M., Ifrim, G., Ramanath, M., Weikum, G.: NAGA: Searching and ranking knowledge. In: 2008 IEEE 24th International Conference on Data Engineering, ICDE 2008, pp. 953–962. IEEE (2008)
12. Kolaitis, P.G., Vardi, M.Y.: Conjunctive-query containment and constraint satisfaction. J. Comput. Syst. Sci. **61**(2), 302–332 (2000)
13. Lohmann, S., Heim, P., Stegemann, T., Ziegler, J.: The relfinder user interface: interactive exploration of relationships between objects of interest. In: Proceedings of the 15th International Conference on Intelligent User Interfaces, pp. 421–422. ACM (2010)
14. Shekarpour, S., Auer, S., Ngonga Ngomo, A.C., Gerber, D., Hellmann, S., Stadler, C.: Generating SPARQL queries using templates. Web Intell. Agent Syst. Int. J. **11**(3), 283–295 (2013)
15. Yahya, M., Barbosa, D., Berberich, K., Wang, Q., Weikum, G.: Relationship queries on extended knowledge graphs. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, pp. 605–614. ACM (2016)

The 5th International Symposium on Semantic Computing and Personalization (SeCoP 2018)

A Corpus-Based Study on Collocation and Semantic Prosody in China's English Media: The Case of the Verbs of *Publicity*

Qunying Huang[1], Lixin Xia[2](✉), and Yun Xia[3]

[1] Faculty of English Language and Culture,
Guangdong University of Foreign Studies, Guangzhou 510420, China
[2] Laboratory of Language Engineering and Computing,
Guangdong University of Foreign Studies, Guangzhou 510420, China
cdhuiyi@aliyun.com
[3] Nanfang College of Sun Yat-Sen University, Guangzhou 510420, China

Abstract. This paper explored extensively the collocability and semantic prosody of the verbs of *publicity* in China's English media, and then contrasted them with those in American newspapers. The purpose of the study is to disclose the attitudinal meanings from their linguistic performance. To that end, the Corpus of China's English Media (CCEM) was built with 29,151,028 tokens, and the newspaper branch of COCA was used as a comparable corpus. All the col-locational patterns of the verbs were extracted from the two corpora, and they were further analyzed in great detail. The major findings of the study can be summarized as below: (1) The verbs of *publicity* are generally used more frequently in CCEM. (2) Chinese journalists tend to use some unique collocational patterns to describe some political, cultural or social concepts that are specific to China. (3) Chinese journalists have a preference for collocational patterns with a positive or neutral semantic prosody over those with a negative semantic prosody. These findings suggest that the attitudinal meanings could be worked out from the differences in collocational behaviors and semantic prosody between the texts in CCEM and COCA.

Keywords: Collocation · Semantic prosody · Semantics · Social media
Corpus linguistics · Linguistic theories

1 Introduction

English media has achieved a substantial growth in China since its introduction of the reform and opening policy. It is reported that there are in all more than 20 English newspapers and periodicals alone in Mainland China (Zhang 2012).

China's English media has become one of the major means to present the story of China to the world. Most of the English language reports in Chinese media are written by Chinese journalists. Guo and Huang (2012: 227) reported that around 70% of the news articles in the three newspapers they investigated were written by the publication's Chinese staff.

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 203–217, 2018.
https://doi.org/10.1007/978-3-319-91455-8_18

Collocation and semantic prosody are two closely related terms in linguistics, and mainly deal with the structure and meaning of word combinations. By examining the differences in collocational behavior and semantic prosody between texts in China's English language media and the English language media in the United States, it will disclose the levels of meaning from their use in language. This study, from the perspective of corpus linguistics, will attempt to answer the following three research questions:

(1) Are there any differences between China's English media and American media in the use of the verbs of *publicity*? If so, what are they?
(2) Are there any differences between China's English media and American media in the use of the collocates occurring to the right of the verbs of *publicity*? If so, what are they?
(3) Are there any differences between China's English media and American media in terms of semantic prosody of the verbs of *publicity*? If so, what are they?

2 Relevant Studies

The linguistic features of English media in China have been a core subject for many researchers. Most of the studies have been conducted at either the lexical or grammatical level.

2.1 Studies at the Lexical Level

Wen and Yu (2003) conducted a study on English expressions with Chinese characteristics and their intelligibility in China's English newspapers with a sample of around 150 news articles in the publication, *the 21st Century*. They parsed the articles and picked up 88 English expressions found to have distinct Chinese characteristics. With the help of English native speakers, they concluded that 96.6% of the expressions are intelligible, and that localization of English occurs only at the lexical level in China's English newspapers.

Gao (2006a) extracted 500 key words from China's English News Articles Corpus (CENAC), about 54% of which are unique Chinese borrowings, coined words, and words with semantic shift. She further analyzed their linguistic features, and concluded that nativized words and expressions are common in China's English media. In the same way, Gao (2006b) studied 1,736 titles from CENAC, and reached the same conclusion that the lexical units found in the article titles are nativized to express Chinese politics, culture, and social life.

Zhang (2016) extracted 1,231 English words and expressions found to have Chinese characteristics from the Corpus of English on China (CEC). Their distributive features and frequencies of use were analyzed and calculated. Through in-depth analyses, Zhang found that there are significant differences in the use and treatment of these words and expressions between Chinese and international English media.

2.2 Studies at the Grammatical Level

Li (2007) examined multi-word units found in CENAC. The results of Li's study show that Chinese news reporters tend to use longer word clusters in their reporting when compared with their British counterparts. Moreover, these word clusters are ex-pressive, with fixed structure and specific linguistic and social meanings, which is indicative of the nativization of multi-word units in China's news articles.

Hu and Zhang (2015) collected 1,681 news articles published in the *China Daily* during the period from July to October 2011, and built a Micro Corpus of the China Daily (MCCD) with about 1 million tokens. By comparing the semantic, collocational, and grammatical patterns of high-frequency verbs of transformation in the MCCD and the Corpus of Contemporary American English (COCA), they found that the verbs under investigation tend to co-occur with certain types of nouns which mirror the current status of Chinese society, and that colligation patterns more closely resemble those found in the Chinese language.

Yu (2006) compared the collocational patterns of the word foreign in CENAC and the NBNC (the news subcorpus of the BNC), and found that the collocates of the base *foreign* are more diversified in CENAC than those in the NBNC. The use of the word *foreign* in China's newspapers shows an implicit tendency toward nativization. Yu and Wen (2010) used the same corpora and examined the collocational patterns of 20 high frequency evaluative adjectives. They concluded that the use of English in Chinese media outlets displays systematic nativization, which is both grammatical and intelligible.

To sum up, all these studies employ corpus-based research methodology to examine the nativization of English in China. They either used existing corpora (Gao 2006a, b; Yu 2006; Li 2007; Yu and Wen 2010) or built their own (Wen and Yu 2003; Hu and Zhang 2015; Zhang 2016) to achieve their research objectives.

However, these studies have the following limitations. Firstly, they only concentrate on the lexical or collocational localization of English, but ignore the importance of looking at the attitudinal meanings of these collocations. Secondly, and with the exception of the CEC, all the corpora used in the aforementioned studies are at a scale of around 1 million tokens, which is not large enough to reveal the linguistic behavior under investigation sufficiently. The size of Zhang's (2016) corpus is larger, but his study only covers lexical units with Chinese characteristics and forgoes investigating collocation or semantic prosody. Thirdly, the corpora are not fully representative of English language media in China. For example, the data for the tailored corpora comes from a single source: *The 21st Century* for Wen and Yu's corpus (2003), and the *China Daily* for Hu and Zhang's corpus (2015). CENAC only covers three newspapers: the *China Daily*, the *Beijing Weekend*, and the *Shanghai Star*. Finally, although researchers agree that nativization of English in China has occurred at the lexical level, there does not seem to be a consensus that it has occurred at the syntactic level. Wen and Yu (2003: 8) argue that localization of the English language is not common in China's officially published English periodicals, and it does not occur at the syntactical level, even though most scholars hold the opposite view.

In contrast to past studies on the topic, this paper will focus on collocation and semantic prosody in order to explore further the attitude and evaluation of China's

English media. In order to achieve the research objectives, a larger corpus of English media in China will be collected, and the issue of representativeness has been tackled more carefully by including more English language newspapers and periodicals at various levels.

3 Methods

It is generally acknowledged that collocation and semantic prosody are “inaccessible to a speaker’s conscious introspection” (Xiao and McEnery 2006: 106). Therefore, this study will follow a corpus-based research methodology.

3.1 Source of the Data

The linguistic data used in this study come from the Corpus of China’s English Media (CCEM) and the COCA. The former was specially built for this study based on the domestic news branch of the CEC. The latter is utilized in this study as a reference corpus. The CCEM collected 87,658 texts from six major English language newspapers and periodicals published in China from 2006 to 2013. The six include *China Daily, People’s Daily, Global Times, Shanghai Daily, China Today*, and *Beijing Review*. The corpus has about 29,151,028 tokens, covering a wide range of topics in politics, economy, culture, and social affairs in China. The reference corpus utilizes the newspaper branch of the COCA, which consists of 105,963,844 tokens out of its sum total 533,788,932 tokens.

3.2 Procedures

Data extraction and analysis followed the procedures numerated below:

First, all the data in the CCEM was tagged using the software TagAnt 1.2.0 so that all the words in the corpus were given a part-of-speech tag. This enables one to distinguish uses of the verbs indicating publicity in its verb form.

Second, the software AntConc. 3.4.4 was used to extract the collocational patterns of the verbs of *publicity* from the CCEM. For example, by imputing the query expression “publicize_vv *_nn” into AntConc, all the collocations of publicize + nouns were listed. The collocational patterns of the verbs in the newspaper branches of the COCA were queried one by one.

Third, all the collocations from both the CCEM and the COCA were sorted manually and analyzed in detail. Focus was placed on finding the differences between China’s English language news media and American media in terms of collocational behavior and semantic prosody.

Fourth, the log-likelihood ratio calculator developed by Liang Maocheng was used. This is a tool which enables easy calculation of log-likelihood and chi-square values needed in corpus contrastive studies. By imputing the following four values into the calculator, one can get the log-likelihood and significance values of the lexical item in the two corpora: the token numbers of the two corpora, and the original frequency numbers of the lexical item in the two corpora. If the significance value is less than 0.05, then the result can be considered statistically significant.

4 Results

To accomplish the objectives of the study, the following verbs of publicity were chosen: *publicize, propagate, disseminate, preach, circulate*, and *propagandize*, which are examined in the context of meaning to "make something known to the public".

4.1 Frequency

The frequency of the verbs in the two corpora is shown in Table 1.

Table 1. The frequency of the verbs in the two corpora

Verbs	Original/Norm. freq. in CCEM	Original/Norm. freq. in COCA	Log-likelihood	Significance
Publicize	673/23.2	833/7.9	398.43	0.000***+
Propagate	35/1.2	137/1.3	0.15	0.694−
Disseminate	121/4.2	236/2.2	28.66	0.000***+
Preach	127/4.4	1510/14.2	230.27	0.000***−
Circulate	414/14.3	964/9.1	53.85	0.000***+
Propagandize	11/0.4	20/0.19	3.14	0.077+

From Table 1, one can see that three out of the six verbs are significantly overused in China's English news media compared with those in American news media, as their p-values are less than 0.05. The three verbs are *publicize, disseminate*, and *circulate*. The verb *propagandize* is overused in China's English news media, but not to a significant level. However, the verb *preach* is significantly underused in China's English news media. The verb *propagate* has nearly the same frequency in both corpora.

4.2 Verb + Noun Collocation

Collocations can be approached with different methods, such as Sinclair's statistical method (Sinclair 1991) or Cowie's phraseological method (Cowie 1998). For the purpose of this study, the head nouns co-occurring to the right of the verbs functioning as an object, and those to the left of the verb functioning as a subject when the verbs are used in a passive form, are considered as a verb + noun collocations. The most frequent of these are listed in Table 2.

The word *publicize* is highly overused in the CCEM. 17 out of the 20 collocations listed in Table 2 are significantly overused in China's English news media. The following collocations are uniquely Chinese, as they occur only in the English texts

Table 2. Collocational patterns of the verb publicize

V + N collocation	Original freq. in CCEM	Original freq. in COCA	Log-likelihood	Significance
Publicize information	71	6	178.55	0.000***+
Publicize case	19	53	0.94	0.333+
Publicize list	19	1	50.82	0.000***+
Publicize result	18	4	36.29	0.000***+
Publicize budget	13	2	29.07	0.000***+
Publicize report	12	3	23.25	0.000***+
Publicize plan	12	3	23.25	0.000***+
Publicize statement	11	1	27.34	0.000***+
Publicize detail	9	3	15.57	0.000***+
Publicize knowledge	8	1	18.75	0.000***+
Publicize policy	8	0	Invalid	Invalid
Publicize data	8	0	Invalid	Invalid
Publicize asset	8	0	Invalid	Invalid
Publicize regulation	7	0	Invalid	Invalid
Publicize warning	6	2	10.38	0.001**+
Publicize spirit	6	0	Invalid	Invalid
Publicize law	6	0	Invalid	Invalid
Publicize expenditure	6	0	Invalid	Invalid
Publicize shooting	0	6	Invalid	Invalid
Publicize trial	1	12	1.85	0.174−

written by Chinese news reporters: *publicize policy/data/asset/regulation/spirit/law/expenditure*. By contrast, only one collocation is not found in the CCEM.

As the verb *propagate* has a low frequency in both corpora, there are only a small number of collocational patterns (see Table 3), none of which is significantly overused or underused in the CCEM. The exception is the collocation propagate view.

The case of *disseminate* is more complicated. As shown in Table 4, some collocational patterns appear only in Chinese English news texts, such as *disseminate rumor*. However, some collocations common in American news texts are not found in the text for their Chinese counterparts, such as *disseminate news/report/lie/propaganda*.

Table 3. Collocational patterns of the verb propagate

V + N collocation	Original freq. in CCEM	Original freq. in COCA	Log-likelihood	Significance
Propagate myth	1	7	0.44	0.506−
Propagate value	1	3	0.03	0.870+
Propagate hatred	1	2	0.22	0.639+
Propagate idea	2	2	1.56	0.211+
Propagate view	2	0	Invalid	Invalid

Table 4. Collocational patterns of the verb disseminate

V + N collocation	Original freq. in CCEM	Original freq. in COCA	Log-likelihood	Significance
Disseminate information	20	54	1.23	0.267+
Disseminate rumor	12	0	Invalid	Invalid
Disseminate culture	7	1	15.93	0.000***+
Disseminate knowledge	6	3	8.40	0.004**+
Disseminate technology	4	1	7.75	0.005**+
Disseminate line	3	1	5.19	0.023*+
Disseminate message	1	12	1.85	0.174*−
Disseminate news	0	12	Invalid	Invalid
Disseminate report	0	8	Invalid	Invalid
Disseminate idea	1	4	0.01	0.931−
Disseminate lie	0	4	Invalid	Invalid
Disseminate data	1	3	0.03	0.870+
Disseminate propaganda	0	3	Invalid	Invalid
Disseminate view	1	3	0.03	0.870+

Different from the other verbs denoting publicity, the verb *preach* is underused in the CCEM as shown in Table 5, which leads to fewer collocational patterns in the English texts of the CCEM. Some frequently used collocations in the COCA cannot be found in the CCEM. They are *preach sermon/message/patience/hatred/word/defense.*

In Table 6, the following four collocational patterns are found to be significantly overused in the CCEM: *circulate draft/picture/photo/report*. Additionally, the collocation circulate petition is significantly underused in the CCEM. At the same time, some collocational patterns occur exclusively in the Chinese texts (*circulate notice/post/story/article*) or exclusively in the American texts (*circulate memo/proposal/email*).

Table 5. Collocational patterns of the verb *preach*

V + N collocation	Original freq. in CCEM	Original freq. in COCA	Log-likelihood	Significance
Preach benefit	3	2	3.44	0.063+
Preach virtue	2	10	0.18	0.670−
Preach theory	2	0	Invalid	Invalid
Preach gospel	2	73	23.17	0.000***−
Preach Buddhism	2	0	Invalid	Invalid
Preach value	2	10	0.18	0.670−
Preach sermon	0	24	Invalid	Invalid
Preach message	0	22	Invalid	Invalid
Preach patience	0	12	Invalid	Invalid
Preach hatred	0	10	Invalid	Invalid
Preach word	0	10	Invalid	Invalid
Preach defense	0	9	Invalid	Invalid
Preach importance	1	9	0.94	0.332
Preach Christianity	1	8	0.68	0.411

Table 6. Collocational patterns of the verb *circulate*

V + N collocation	Original freq. in CCEM	Original freq. in COCA	Log-likelihood	Significance
Circulate draft	6	4	6.89	0.009**+
Circulate picture	5	1	10.42	0.001**+
Circulate notice	5	0	Invalid	Invalid
Circulate post	4	0	Invalid	Invalid
Circulate photo	4	1	7.75	0.005**+
Circulate story	3	0	Invalid	Invalid
Circulate rumor	3	3	2.34	0.126+
Circulate report	3	1	5.19	0.023+
Circulate information	3	3	2.34	0.126+
Circulate article	3	0	Invalid	Invalid
Circulate letter	2	6	0.05	0.817+
Circulate statement	2	5	0.19	0.664+
Circulate petition	1	31	9.24	0.002**−
Circulate memo	0	9	Invalid	Invalid
Circulate plan	1	7	0.44	0.506−
Circulate proposal	0	6	Invalid	Invalid
Circulate email	0	4	Invalid	Invalid

Table 7. Collocational patterns of the verb *propagandize*

V + N collocation	Original freq. in CCEM	Original freq. in COCA	Log-likelihood	Significance
Propagandize theory	2	0	Invalid	Invalid
Propagandize idea	1	0	Invalid	Invalid
Propagandize extremism	1	0	Invalid	Invalid
Propagandize democracy	1	0	Invalid	Invalid
Propagandize partnership	1	0	Invalid	Invalid
Propagandize clash	1	0	Invalid	Invalid
Propagandize communism	0	1	Invalid	Invalid
Propagandize history	0	1	Invalid	Invalid
Propagandize idealism	0	1	Invalid	Invalid
Propagandize issue	0	1	Invalid	Invalid
Propagandize morality	0	1	Invalid	Invalid
Propagandize life	0	1	Invalid	Invalid
Propagandize public	0	1	Invalid	Invalid

Notably, in Table 7 there are two completely different sets of collocates for the node: one set in the CCEM (*propagandize theory/idea/extremism/democracy/partnership/clash*) and the other in the COCA (*propagandize communism/history/idealism/issue/morality/life/public*).

4.3 Semantic Prosody

Originating from Sinclair, the term "semantic prosody" has different interpretations in corpus linguistics. However, there is a general consensus that it relates to the attitudinal meaning. In Sinclair's words, semantic prosody is "on the pragmatic side of the semantic/pragmatics continuum" (Sinclair 1996: 87). In Louw's words, it expresses "the attitude of its speaker or writer" (Louw 2000: 60). A given word tends to collocate regularly with a group of other words characterized by favorable, neutral, or unfavorable meanings. As a result, the given word takes on some of the meaning of the group of other words, and becomes associated with a pleasant or unpleasant meaning. In this paper, this is defined as a positive, negative, or neutral semantic prosody.

Since collocation is defined in this paper as a co-occurrence of any two lexical items "with mutual expectancy greater than chance" (Wei 2002: 100), the collocations with frequencies greater than two are included in order to analyze their semantic prosodies. The semantic prosodies of the collocations of the verbs of *publicity* are shown in Table 8.

Table 8. Semantic prosodies of the collocations

Verbs	Original/Norm. freq. in CCEM			Original/Norm. freq. in COCA		
	Positive	Negative	Neutral	Positive	Negative	Neutral
Publicize	20/0.69	32/1.1	462/15.93	14/0.13	84/0.79	318/3
Propagate	0/0	5/0.17	14/0.48	9/0.08	15/0.14	25/0.24
Disseminate	20/0.69	18/0.62	66/2.28	12/0.11	7/0.07	154/1.45
Preach	23/0.79	5/0.17	18/0.62	103/0.97	38/0.35	330/3.11
Circulate	5/0.17	6/0.21	60/2.07	1/0.01	5/0.05	165/1.56
Propagandize	2/0.07	2/0.07	3/0.10	2/0.02	0/0	5/0.05

From Table 8, we can see that the collocations for the verb *publicize* has primarily a neutral semantic prosody in both corpora. However, it is more likely to co-occur with a pleasant collocate in the CCEM than in the COCA. The log-likelihood data indicate that the collocations with positive and neutral semantic prosodies in the CCEM are significantly overused with a p-value of 0.000. Similarly, the verb propagate is found to chiefly accompany a neutral collocate in both corpora, and the collocations with a neutral semantic prosody are significantly overused in the CCEM with a p-value of 0.041.

An overwhelming majority (89%) of the collocations for the verb *disseminate* have a neutral semantic prosody in the COCA. The rate for the collocations in the CCEM is about 63%. Moreover, the collocations with positive ($p = 0.000$) and neutral ($p = 0.004$) semantic prosodies are significantly ($p = 0.000$) overused in the CCEM. The verb *preach* has a mainly neutral semantic prosody in the COCA, and accounts for more than 70% of its total co-occurrence. However, it has a greater chance (50%) of having a positive semantic prosody in the CCEM. The collocational patterns with neutral semantic prosodies are significantly underused ($p = 0.000$) in the CCEM in companion to the COCA.

The verb *circulate* has a predominantly neutral semantic prosody in both corpora, as shown in Table 8. Specifically, the rates are 85% in the CCEM and 96% in the COCA. Moreover, the collocations with a positive semantic prosody are significantly overused in the CCEM with a p-value of 0.001.

The verb *propagandize* is a special case. Except for the collocate *theory*, no other collocate occurs more than once in either the CCEM or the COCA. In the CCEM, the verb goes with two favorable words (democracy and partnership), two neutral words (theory and idea) and two unfavorable words (extremism and clash). In the COCA, it is associated with seven other words, two of which can be said to have a pleasant meaning (morality and idealism), and the remaining five of which can be said to have a neutral meaning (communism, history, issue, life, and public).

5 Discussion

The answer to the first research question is a resounding affirmative. Of the six verbs under investigation, five are overused or underused in the CCEM, and only the verb propagate is used with almost the same frequency in both corpora. These differences in frequency may reflect the differences in the attitudes of China's news media and American news outlets.

First of all, China's English news reporters show a potential preference for reporting the positive side of daily events, even though they strive to be objective. That might explain why Chinese journalists significantly overuse the verbs *publicize, disseminate,* and *circulate*, which have an approving or at least a neutral connotation. In addition, the verb *propagandize*, which has a disapproving connotation, is not overused significantly. This style of reporting may be a reflection of Chinese culture, which values harmony very highly. Moreover, these Chinese English news media sources are targeted primarily at an international readership. It is, therefore, all the more important to present the more positive aspects of China to the world. *China Today*, a monthly magazine investigated in this study, sets out to "promote a knowledge of China's culture, geography, economy and social affairs as well as positive view of the People's Republic of China and its government to people outside of China" (https://en.wikipedia.org/wiki/China_Today, retrieved 30 May 2017).

Secondly, China's English news media have a different focus of news coverage than their American counterparts. Although the Chinese media try to cover both domestic and international news, they are in fact more concerned with what is happening in China. For example, the *Beijing Review*, a weekly magazine, stated its editorial policy in its first, 1958 volume (the publication was called the *Peking Review* at that time), to "provide timely, accurate, first-hand information on economic, political and cultural developments in China, and her relations with the rest of the world" (Peking Review 958: 3). The *China Daily*, a daily newspaper, states that it "provides 24-h authoritative information on China through multiple channels" (http://www.chinadaily.com.cn/static_e/digitalmedia.html, retrieved 30 May 2017). These editorial policies result in unbalanced news coverage. The underused verb *preach* in the CCEM supports this argument because the word closely associates with Christianity, which is not common in China.

The answer to the study's second research question is strongly positive. The companions to the verbs denoting publicity (either in the object slots or in the subject slots when the verbs are used in a passive form) vary in frequency, form, and meaning in the texts of Chinese English news media and American media. The differences in using particular collocational patterns in particular contexts may be caused by the contrasting cultures, language attitudes, and language habits of the reporters.

These collocational patterns can be categorized into three distinct groups. The unique collocational patterns occurring only in the CCEM fall into the first group. Examples include *publicize policy/asset/regulation/spirit/law/expenditure*, as shown in the concordance lines below:

1. Officials would visit these families, **publicize** the **policy** allowing a second ch…
2. Early in January, Wenzhou **publicized** a trial **policy** that would allow individu…

3. Bureau called on 20 travel agencies to **publicize** the new **law** and required it…
4. The Gazette account **publicizes** administrative **laws** and orders of the…
5. …political bureau approved and agreed to **publicize** the **Regulation** on Strictly…
6. Central authorities **publicized** late last month 65-item **regulations…**

These collocations, specific to Chinese English texts, may come from their corresponding expressions in the Chinese language. In Chinese, there are such phrases as xuān chuán (*publicize*) zhèng cè (*policy*) (see examples 1 and 2 above), xuān chuán (*publicize*) fǎ lǜ (*law*) (see examples 3 and 4 above), and xuān chuán (*publicize*) fǎ guī (*regulation*) (see examples 5 and 6 above). They are directly translated into English. This conclusion is supported by Kachru's (1990) argument that nativized collocations for world Englishes originate from loan translations of the collocational structure in the source language.

Another source of nativized collocational patterns are the specific concepts and referents only found in China.

7. …conscientiously study, **publicize** and implement the **spirit** of the 18th…
8. A campaign will be launched to **publicize** the Party national congress **spirit**.
9. …from the CPC to foreign countries to **publicize** the **spirit** of the CPC session.
10. …with six eminent monks **preaching Buddhism**. The thoughtful composition…
11. …brought to China by monks who were **preaching Buddhism**.

For examples 7 through 9, the collocation publicize spirit represents a specific political concept in China, and the word *spirit* is frequently used in conjunction with the Communist Party of China (CPC) sessions. Buddhism is a major religion in China, but is not as common in the United States. Therefore, the collocation *preaching Buddhism* (see examples 10–11) does not occur in the texts of American newspapers.

Some unique collocational patterns in the CCEM have been found to have undergone a process of semantic shift. The collocations *publicize asset* and *publicize expenditure* are cases representative of this shift.

12. …Government began to **publicize** its **expenditure** related to the three public…
13. ….requesting them to **publicize** their **expenditure** on public receptions,…
14. …and city governments have **publicized expenditure** on overseas trips, …
15. …, also expressed their willingness to **publicize** their families' **assets**.
16. "Their relatives' **assets** should also be **publicized** as some officials will trans…
17. "I would like to **publicize** my family's **assets** if the authorities…
18. …soon, the concept of **publicizing** officials' **assets** is beginning to tak…

The so called "three public expenditures" is a specific Chinese expression, meaning the consumption of public funds in official receptions, the acquisition of vehicles and visits abroad. The Chinese government at various levels is required to disclose these three public expenditures. So *publicize* here means to make known something heretofore kept secret (see examples 12 through 14). It is the same with the collocational pattern for publicize asset. As a measure for anti-corruption policy in China, government officials are required to make public their assets. Therefore, *publicize* in this context can mean to make something public (see examples 15 through 18).

The overused collocational patterns found in the CCEM are classified as the second group. Some of the overused collocations may be derived from fixed Chinese expressions. For example, in Chinese, it is common to say "gōng bù (*publicize*) míng dān (*list*)", "gōng bù (*publicize*) hēi míng dān (*blacklist*)", and "gōng bù (*publicize*) jié guǒ (*result*)", which are more frequently expressed in English as "provide list", "keep blacklist", and "announce result" in the COCA. Some overused collocational patterns might be the results of the difference in the use of certain words in the two corpora. For instance, *publicize information/knowledge* is overused in the CCEM, while in the COCA, American journalists prefer to use the terms *disseminate information/ knowledge*.

The third group is comprised of the underused collocational patterns found in the CCEM. These patterns mainly reflect the different political, economic, and social situations in China and America. In the COCA, one can find more collocations, as shown in the following concordance lines:

19. …a study that examines whether well **publicized** mass **shootings** increase…
20. …in recent years, despite highly **publicized** school **shootings**. Many states…
21. ….including invitations to **preach sermons** in churches. He isn't sure about…
22. …her to stand behind a pulpit and **preach** a **sermon**, or teach from the Bible…
23. ….even from the pulpit while **preaching** the **Gospel**. All of the above is true…
24. ….a bold and unmistakable voice, **preaching** the **Gospel** of Grace in a way…
25. ….records of the attacks; they have **circulated petitions** and rallied crowds…
26. …called Prosperity Patriots **circulated** a **petition** to get the alcohol issue…

Shootings take place more often on campuses in the United States. As a result, it is the focus of reporting in American newspapers, whereas such incidents rarely take place in China. Similarly, the public circulation of a political petition is unlikely to occur in China. The phrases *preach sermon* and *preach Gospel* depict situations in which speeches are delivered on religious subjects during occasions of public speaking, which is also uncommon in China.

The answer to the last research question is also affirmative, but to varying degrees. Generally speaking, the verbs *publicize, propagate, disseminate, preach,* and *circulate* have a primarily neutral semantic prosody in the COCA. However, they are found to be used differently in the CCEM. First, although the verbs *publicize, disseminate,* and *circulate* mainly have a neutral semantic prosody in the CCEM, they are more likely to co-occur with a favorable word in the CCEM than in the COCA. Moreover, Chinese news reports tend to overuse the verbs *publicize, propagate, disseminate,* and *circulate* with a positive or neutral semantic prosody. This may indicate that they prefer to report more good news or tend to take a more positive attitude toward what they report. Secondly, the collocations of preach with a negative meaning are found to be underused in the CCEM, which additionally indicates that English news media in China exhibit a preference for reporting on positive events. Thirdly, while the verb *propagandize* itself has a disapproving connotation, it co-occurs with favorable or neutral words in the COCA, such as *propagandize morality/idealism/history*, etc. It thus has a neutral or positive semantic prosody in the COCA. In the CCEM, *propagandize* has an equal chance to pair with a favorable, unfavorable, or neutral word.

6 Conclusion

Compared to their Chinese counterparts, the reporting style of English news reporters in China closely resembles the Western style. However, the tone of the English language news reports is still in line with all news media in China. The primary concern of this study was the attitudinal meanings embedded in the collocations exhibited in the CCEM and the COCA. It was assumed that if there were any differences between them, this would be reflected in their collocational patterns and semantic prosody. Below is a brief summary of the study's major findings:

(1) The six verbs examined have varying frequencies in the CCEM and the COCA. Four verbs are overused, and one verb is underused in the CCEM. (2) The collocational patterns of the verbs indicating publicity in the CCEM are different from those found in the COCA in frequency, form, and meaning. Chinese journalists, on the one hand, use some unique collocations that occur only in the CCEM, and on the other hand underuse or overuse some collocations to illustrate their attitudes toward specific conditions and situations. (3) The semantic prosodies of the collocations in the CCEM are quite different from those found in the COCA. Chinese journalists are more inclined to use collocational patterns with a positive or neutral semantic prosody, and tend to underuse those with a negative semantic prosody.

The findings of this study suggest that English news media in China have a different focus for news coverage compared to their American counterparts, and that they tend to take a more positive attitude towards what they report.

Acknowledgement. This work was supported by a grant from the Humanities and Social Science Planning and Funding office of Ministry of Education of the People's Republic of China under Grant No. 15YJA740048 (China English or Chinese English: A Study Based on China English Corpus), a grant from Guangdong Planning Office of Philosophy and Social Science (Grant No. GD14XWW20) and a grant from the Laboratory of Language Engineering and Computing of Guangdong University of Foreign Studies under the project entitled "A Study on Grammatical Nativization of China English" (Grant No. LEC2017ZBKT002). It was also part of the research achievements of the special innovation project entitled "A Study on the Definitions of Chinese-English Dictionaries for CFL Learners from the User's Perspective" (Grant No. 2017WTSCX34) granted by the Department of Education of Guangdong Province.

References

Cowie, A.P.: Phraseology: Theory, Analysis and Applications. Oxford University Press, Oxford (1998)

Gao, C.: Jiyu yuliaoku de zhongguo xinwen yingyu zhutici yanjiu (A corpus-based study on the key words in China's news English). Beijing Dier Waiguoyu Xueyuan Xuebao (Journal of No. 2 Beijing Foreign Languages Institute) **136**(6), 36–43 (2006a)

Gao, C.: Jiyu yuliaoku de xinwen yingyu biaoti yanjiu (A corpus-based study on the titles in China's news English). Jiangsu Waiyu Jiaoxue Yanjiu (Jiangsu Foreign Languages Teaching and Research) **136**(6), 36–43 (2006b)

Guo, Z., Huang, Y.: Hybridized discourse: Social openness and functions of English media in post-Mao China. World Englishes **21**(2), 217–230 (2002)

Hu, J., Zhang, P.: Jiyu yuliaoku de zhongguo yingyu baozhang gaopin biange fongci bentuhua tezheng yanjiu (A corpus-based study on nativization of high-frequency verbs of transformation in China's English Newspapers). Yuliaoku Yuyanxue (Corpus Linguistics) **2**(1), 59–70 (2015)

Kachru, B.B.: The Alchemy of English: The Spread, Functions, and Models of Non-native Englishes. University of Illinois Press, Urbana, Chicago (1990)

Li, W.: Zhongguo yingyu xinwen baoz-hang zhong de cicu (Word clusters in China English News Articles Corpus). Zhongguo Waiyu (Foreign Languages in China) **4**(3), 28–38 (2007)

Louw, B.: Contextual prosody theory: bringing semantic prosody to life. In: Heffer, C., Sauston, H. (eds.) Words in Context: In Honour of John Sinclair, pp. 48–94. ELR, Birmingham (2000)

Peking Review: Introducing peking review. Peking Rev. **1**(1), 3 (1958)

Sinclair, J.: The search for units of meaning. Textus **9**, 75–106 (1996)

Stubbs, M.: Collocations and semantic profiles: on the cause of the trouble with quantitative methods. Funct. Lang. **2**(1), 1–33 (1995)

Wei, N.: Ciyu Dapei de Jiedian yu Yanjiu Tixi (The Definition and Research Methods of Collocation). Shanghai Jiaotong University Press, Shanghai (2002)

Wen, Q., Yu, X.: Yingyu de guojihua yu bentuhua (Internationalization and nativization of the English language). Guowai Waiyu Jiaoxue (Foreign Languages Teaching Abroad) **3**, 6–11 (2003)

Xiao, R., McEnery, T.: Collocation, semantic prosody, and near synonymy: a cross-linguistic perspective. Appl. Linguist. **27**(1), 103–129 (2006)

Yu, X., Wen, Q.: Zhongguo yingyu baozhang zhong pingjia xing xingrongci dapei de bentuhua tezheng (Collocation patterns of evalua-tive adjectives of English in Chinese newspapers). Waiyu Jiaoxue yu Yanjiu (Foreign Language Teaching and Research) **254**(5), 23–28 (2010)

Zhang, X.: Zhongguo yingyu baokan fazhan xianzhuang ji celue fenxi (On the current situation and development strategies of China's English newspapers). Xinwen Fenxi (News Analysis) **9**, 103–104 (2012)

Zhang, Y.: Zhongguo teseci zai zhongwai chuanmei de shiyong tedian ji yingxiang yinsu (On the characteristics of and factors influencing the use of the words and expressions with Chinese characteristics in Chinese and International media). Xueshu Yanjiu (Academic Research) **7**, 151–156 (2016)

Location Dependent Information System's Queries for Mobile Environment

Ajay K. Gupta(✉) and Udai Shanker

Department of Computer Science and Engineering,
M.M.M. University of Technology, Gorakhpur, India
ajay25g@gmail.com, udaigkp@gmail.com

Abstract. Location dependent information services can be characterized as the applications that coordinate a cell phone's area or position with other data to give enhanced value of services to the client in the right place and at the right time from anywhere. Limited battery power and frequent disconnection due to moving environment prompts mobile distributed database to be a fertile land for many mobile databases researchers and specialists. New policies/protocols must be designed to efficiently handle the issued nearest neighbor queries. Our works involves design of new cache replacement policies, indexing, pre-fetching protocols with comparison of their performances from existing policies/ protocols and reporting for future research directions.

Keywords: Mobile computing · Location dependent data · Cache replacement
Predicted region · Root-mean squared distances · Valid scope
Cache invalidation

1 Introduction

Location-Based Services (LBS) are one of the emerging applications among various mobile and wireless based technology. LBS provide context aware information to the client at the right time in the right place. Many of the technological constrains are added to these applications to maintain integrity and consistency of result acquired from LBS-Server. The basic reference architecture of mobile databases contains three entities: fixed hosts, mobile units, and base stations. Mobile units are low power moving object having lesser computational functionality that move around a geographical region. These geographical regions are basically divided into wireless cells i.e. mobile client contains data centric applications and roams between wireless cells. LBSs [1, 2] are gaining popularity in current trends where most of the applications use some context-aware information included with the mobile host. Context-aware information includes time, location, and device identity nearby to a given entity. The potential sources of this information are web browser, camera, microphone, GPS Receiver, a server associated with the given entity. Here, entity can be person, device, or application. Location services can be characterized as services that incorporate a cell phone's area or position with other data to give enhanced value to user. It answers the location-related queries which are initiated by moving user, where the location is the parameter of the query which are provided to the client either explicitly with query or

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 218–226, 2018.
https://doi.org/10.1007/978-3-319-91455-8_19

implicitly using a global positioning system (GPS). Some of the applications [3–7] which are gaining popularity in our daily life can be local information access (Traveler information system, navigation maps, news etc.). Apart from this, user makes some nearest-neighbor queries (show me all nearest hotel, ATM, Saloon).

The rest of this paper is organized as follows. Section 2 explores the issues and research challenges important for the performance of LBS. Sections 3 and 4 describes the contributions, major findings of our experimentations. Section 5 has given conclusion and scope for future works in LBS.

2 Performance Issues and Research Challenges

Numerous research works have been done separately on Predictability & Consistency of LBS. To maintain consistency in nearest-neighbor based applications, system needs to identify the data-item's valid scope and store it with them in the cache [8–10]. Due to the associated valid scope with given data item, the user might reject the received result if user has moved to different places and may ask to the server for reprocessing of the query to get valid result. To improve the performance of the system, there should be some mechanism to obtain fast and accurate answers to issued queries. Very little works has been done to optimize the location dependent query processing [11–14]. Data replication is employed in any distributed system to improve reliability and reduce communication costs and response time thus to reduce the incurred network traffic. Data replication technique creates various copies of same data and place it to various nodes based on some performance criteria. The problem of replication in LBS is to find the number of replica that system should create, the place where it should be put and maintain the consistency among various replica for better performance of given system [15–17]. The bandwidth of downlink communication is much greater than that of the uplink communication. The policy [18] should be designed in such a way that messages sent to the server would be less than message received from server. The order in which the query executed is defined as query scheduling [19–21]. Mobility of users makes it more challenging to handle the query scheduling task. One of the limitations in mobile host is its limited battery power. So this is also a considerable issue while making any application for mobile environment as if processing power is restrained; it compromises the capability of every mobile node offering applications and services [22, 23]. Based on previous theoretical studies and concepts, many focused implementations and simulation [24–26] has been done. However, none of the previous simulations has complete prototype consisting of all of the functionality for LBS. So, one of the important issues is implementation of complete prototype [27] for consistent location aware system.

2.1 Predictability and Consistency

For Nearest-Neighbor query applications, the cached spatial data result may become invalid because of the client movement. To maintain consistency, system needs to identify the data-item's valid scope and store it with them in the cache [8–10]. The concept of valid scope information was first proposed by Zheng et al. [8], in which it was used to assemble a semantic cache that allows you to reuse the cached data. GPS serves to identification of the location of any mobile device in geometric location model. The ancient policy particularly Polygonal Endpoints and Approximate Circle schemes show poor performance for invalidation information in terms of overhead and imprecision. In [9] Kumar et al. gives a comparison of various methods to find best suitable candidate for valid scope (i.e. best suitable sub polygon of a given polygon). They proposed a generalized algorithm known as CEB_G which improves the caching efficiency in comparison to that of basic CEB algorithm. The CEB_G algorithm adjusts the overhead and accuracy of valid scope. Moreover, to further improve caching efficiency they proposed a new algorithm CEFAB which considers user's movement pattern and speculation for its future access.

2.2 Cache Replacement Policy

Whenever a query issued, if cache doesn't contains searched data items then system execute replacement process. The problem is to be addressed and also improve cache-hit ratio [28–34]. The conventional cache replacement policy such as LFU, LRU, LRU-K [28] have been widely used in various applications in past. The working principle of these policies are that the access pattern shows temporal locality i.e. the future access pattern dependent on only past access pattern rather than spatial information. Furthest Away Replacement policies [29] which considers both clients current location as well as movement direction in replacement policy. The eviction is made sequentially in the order of distance from the client. This policy dismisses the client's temporal access properties. In the event when mobile clients' direction updated frequently, then it will make unpredicted effect on membership of objects as it will show frequent switching between the in-direction set (towards valid scope) and the out-direction sets (moving far from valid scope). PAID policy evicts the data having the least cost when cache replacement is performed. PAID (Probability Area Inverse Distance) [30] has the limitation that the priorities for the client's location nearby data objects in cache and the effect of the size of data object have not been considered. To overcome this limitation, Mobility Aware Replacement Scheme (MARS) [31] came into existence. The data objects updates are considered in this policy. This policy evaluates the various score such as spatial score, temporal score and an object retrieval cost. The update rate in location dependent data is very small as compared to that of temporal data. In deciding cache replacement data item, the anticipated region has major impact. None of the above described cache replacement policies is fit best if there is frequent updation in client's movement direction because previous schemes consider data distance only and to work with frequent changing direction based location dependent, data the scheme should incorporate functionalities that can predict possible client's near future region/area. In [32] Kumar et al. proposed Prioritized Predicted Region based Cache Replacement Policy (PPRRP) and compared it against previous cache replacement schemes Euclidean, FAR, Manhattan, LRU, PA and PAID. To improve the system performance predicted region is widely employed in location based services. Kumar et al. have given a scheme PPRRP which finds the predicted region of valid scope for client's current position and assigns precedence to the closed data items. The client's movement direction is not considered while assigning priorities to data item. Moreover, using moving interval length (MIc) as radius in this policy has a drawback that system needs to compute the predicted region on the changes in moving client's direction or velocity.

2.3 Prefetching and Indexing

To answer mobile object database queries, searching each location in database is overwhelming task. It degrades the performance of the overall location dependent information system (LDIS). For better performance we would like to do the spatial indexing [35, 36] of location attributes but this indexing cannot be directly applied in LDIS to answer the queries. The reason behind this, the spatial index needs to be update continuously with changes in locations and which would further increase the work load on the system and communication overhead in excessive amount. To achieve better performance, indexing phenomenon is used in which server pre-computes index information and stores it with data for future queries. Here the question of indexing problem is how can we efficiently index the valid scopes of all data instances of a given query type? It is more difficult to index the geometric location mobile queries in comparison to symbolic location models query. In the spatial indexing, structures such as minimal bounding rectangle are used to map the spatial objects. If the MBR will overlaps to each other, then the search performance would be degraded as the overlapping area searched more than once. In air indexing server pre-computes the index information and broadcast it on the outgoing channel. A mobile client seeking for a query can search for its index and can predict the arrival time of the desired data; it is advantageous for mobile client as it allows going on power saving till the queried records arrive on the requested channel. The disadvantage of this phenomenon is searching for additional indexing data makes broadcast cycle longer.

3 New Direction for Research

The future scope of LBS is very vast and diverse. Meeting the issues and challenges discussed previously from all the characteristics would require more extensive and coordinated research efforts in the specific areas such as new metrics for location aware database, performance and predictability, methods and trade-off between various pairs of parameters etc. From the implementation point of view, the work undertaken will address the problems of Location Aware Information System. The main areas of research are listed below.

1. Development of efficient geometric model for valid scope and cache invalidation policy [8–10].
2. Optimization of location dependent query processing and reduce power consumption [11–14].
3. Development of new algorithms for improvement in cache-hit ratio of replacement policy [28–34].
4. Defining efficient process for data prefetching and indexing [35, 36].

5. Performance evaluation of above algorithms with reference to database size, cache size, moving/prediction interval, query interval etc.
6. Analyzing the effects of main memory, secondary storage and buffer size on system performance.
7. Development of fault-Tolerant approaches, Recovery Scheme & Schedulable Conditions for LBS.
8. Implementation of complete prototype for consistent location aware system.

4 Our Contributions

In [44], we have presented various research issues of location aware moving object databases. It includes various existing policies in location-aware mobile data management namely cache invalidation and replacement, mobility recommendation, location data map matching, replication and location privacy. Various sub-areas in the directions of location-aware information system are being explored, where there is any possible research scope. The work done by us gives better solutions for some of the listed research directions stated in the previous section.

1. The accuracy of mobility prediction degrades in LDIS when it involves a lot of random movements [37–39]. Thus, these random movements must be reduced to get accurate mobility prediction. In [45] a sequential pattern mining method in moving client's movement histories [40–43] for the coverage region is employed to find frequent mobility patterns. The paper investigates clustering technique to extract similar mobility behaviors in users moving histories. The SPMC-PRRP model for next location prediction in predicted region was proposed to be used in estimating the distance between data item's valid scope reference point to the anticipated next location of the client. The cache replacement cost function for eviction of data item uses the next location prediction for effective cost computation of valued data items.
2. In one of paper, a cache replacement policy MPRRP [46] is proposed that consists of weighted cache replacement cost function for eviction of data item when cache becomes full. A normalized negative cosine function which considers present moving direction of client is used to assign weight for replacement cost. The predicted region radius estimation method that was defined in PPRRP has been modified in the proposed MPRRP. The radius of predicted region circle is estimated by root mean square distance in the place of moving interval radius. This leads to reducing the unnecessary computation overhead. The proposed MPRRP policy added the temporal locality factor i.e. frequency of use in addition to spatial score for computation of the replacement cost. MPRRP achieves up to 5% performance improvement in terms of cache hit ratio compared to previous replacement policies [28–34].
3. In one of the paper [47] mobility rules based on similarities between user's movements data has be framed to be used in next location prediction. The conditional data distance equation has been revised and being used depending upon whether the data item's valid scope falls within predicted region or outside the

predicted region. The proposed policy achieves significant performance improvement in cache hit ratio on varying outlier ratio, minimum confidence and support threshold.

5 Conclusions

Handling of cache replacement policy is a key issue being studied by us. Despite this, there are many research accomplishments and techniques which have emerged from the area of location bases services. They lead to the increasingly growing interest in the performance mobile distributed database system. Many research accomplishments and techniques, which have emerged from the area of location aware moving database systems. In this research proposal, light has been shed on promising challenges in location aware moving database systems for applying our efforts and resources in a better way to cope up with them. Efforts will also be made to investigate these points for improving the performance for given research area. As for the future work in cache replacement policy, data dissemination schemes, pre-fetching and Hidden Markov Model with Bi-clustering for LDIS can be selected as research area which will overcome the challenges posed by it.

References

1. Weiser, M.: The computer for the 21st century. Sci. Am. Int. Edn. **265**(3), 66–75 (1991)
2. Barbara, D.: Mobile computing and databases: a survey. IEEE Trans. Knowl. Data Eng. **11**(1), 108–117 (1999)
3. Acharya, S., Franklin, M., Zdonik, S.: Balancing push and pull for data broadcast. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, Phoenix, Ariz., pp. 183–194 (1997)
4. Xiao, X., Zheng, Y., Luo, Q., Xie, X.: Finding similar users using category-based location history. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS 2010, pp. 442–445 (2010)
5. Zheng, Y., Zhang, L., Ma, Z., Xie, X., Ma, W.-Y.: Recommending friends and locations based on individual location history. ACM Trans. Web **5**(1), 1–44 (2011)
6. Calabrese, F., Di Lorenzo, G., Ratti, C.: Human mobility prediction based on individual and collective geographical preferences. In: Proceedings of the IEEE Conference on Intelligent Transportation Systems, ITSC, pp. 312–317 (2010)
7. Jeong, J., Lee, K., Abdikamalov, B., Lee, K., Chong, S.: TravelMiner: on the benefit of path-based mobility prediction. In: 2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), London, UK, pp. 1–9 (2016)
8. Zheng, B., Xu, J., Member, S., Lee, D.L.: Cache invalidation and replacement strategies for location-dependent data in mobile environments. IEEE Trans. Comput. **51**, 1141–1153 (2002)
9. Kumar, A., Misra, M., Sarje, A.K.: Strategies for cache invalidation of location dependent data in mobile environment. In: Proceedings of the 2005 International Conference on Parallel and Distributed Processing Techniques and Applications, PDPTA 2005, Las Vegas, Nevada, USA, pp. 38–44 (2005)

10. Ren, Q., Dunham, M.H.: Using semantic caching to manage location dependent data in mobile computing. In: 6th ACM/IEEE Mobile Computing and Networking (MobiCom), Boston, MA, USA, vol. 3, no. 2, pp. 210–221 (2000)
11. Seydim, A.Y., Dunham, M.H., Kumar, V.: Location dependent query processing. In: Proceedings of the 2nd ACM International Workshop on Data Engineering for Wireless and Mobile Access, pp. 47–53 (2001)
12. Madria, S.K., Bhargava, B., Pitoura, E., Kumar, V.: Data organization issues for location-dependent queries in mobile computing. In: Proceedings of the East-European Conference on Advances in Databases and Information Systems Held Jointly with International Conference on Database Systems for Advanced Applications: Current Issues in Databases and Information Systems, 05–08 September, pp. 142–156 (2000)
13. Ilarri, S., Mena, E., Illarramendi, A.: Location-dependent query processing: where we are and where we are heading. ACM Comput. Surv. **42**(3), 12–73 (2010)
14. Wolfson, Y.Y.O., Sistla, A., Chamberlain, S.: Updating and querying databases that track mobile units. Distrib. Parallel Databases **7**(3), 257–387 (1999)
15. Michael, K.: Location-based services : a vehicle for IT & T convergence. In: Advances in e-Engineering and Digital Enterprise Technology, pp. 467–477. Professional Engineering Publishing, University of Wollongong Research Online UK (2004)
16. Zhang, G., Liu, L., Seshadri, S., Bamba, B., Wang, Y.: Scalable and reliable location services through decentralized replication. In: 2009 IEEE International Conference on Web Services, ICWS 2009, pp. 632–638 (2009)
17. Wu, S.: Dynamic data management for location based services in mobile environments. In: Proceedings of the Seventh International Database Engineering and Application Symposium, pp. 180–189 (2003)
18. Patil, A.S.P., Nimbhorkar, S.U.: A survey on location based authentication protocols for mobile devices. IJCSN Int. J. Comput. Sci. Netw. **2**(1), 44–47 (2013)
19. Babcock, B., Babu, S., Motwani, R., Datar, M.: Chain: operator scheduling for memory minimization in data. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data - SIGMOD 2003, pp. 253–265 (2003)
20. Schroeder, B., Harchol-Balter, M., Iyengar, A., Nahum, E.: Achieving class-based QoS for transactional workloads. In: Proceedings of the International Conference on Data Engineering, vol. 2006, pp. 153–155 (2006)
21. Kjaergaard, M.B., Langdal, J., Godsk, T., Toftkjær, T.: EnTracked: energy-efficient robust position tracking for mobile devices. In: Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services, pp. 221–234. ACM (2009)
22. Ravi, N., Scott, J., Han, L., Iftode, L.: Context-aware battery management for mobile phones. In: Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications, pp. 224–233. IEEE Computer Society (2008)
23. Thilliez, M., Delot, T., Lecomte, S.: An original positioning solution to evaluate location-dependent queries in wireless environments. J. Digit. Inf. Manag. Spec. Issue Distrib. Data Manag. **3**(2), 108–113 (2005)
24. Zhu, X., Zhu, G., Guan, P.: Exploring location-aware process management. Geo-Informat. Resour. Manag. Sustain. Ecosyst. **399**, 249–256 (2013)
25. Liang, T.Y., Li, Y.J.: A location-aware service deployment algorithm based on k-means for cloudlets. Mob. Inf. Syst. **2017**, 1–10 (2017)
26. Michael, K.: Location-based services: a vehicle for IT & T convergence. In: Advances in e-Engineering and Digital Enterprise Technology, pp. 467–477. Professional Engineering Publishing, University of Wollongong Research Online UK (2004)
27. Joy, P.T., Jacob, K.P.: Cache replacement strategies for mobile data caching. Int. J. Ad Hoc Sens. **3**(4), 1–9 (2012)

28. O'Neil, E.J., O'Neil, P.E., Weikum, G.: The LRU-K page replacement algorithm for database disk buffering. In: Proceedings of the ACM SIGMOD Conference, vol. 1, pp. 297–306 (1993)
29. Dar, S., Franklin, M.J., Jonsson, B.T., Srivastava, D., Tan, M.: Semantic data caching and replacement. In: VLDB, pp. 330–341 (1996)
30. Lai, K.Y., Tari, Z., Bertok, P.: Location-aware cache replacement for mobile environments. In: Global Telecommunications Conference, GLOBECOM 2004, vol. 6(11), pp. 3441–3447. IEEE, Dallas (2004)
31. Kumar, A., Misra, M., Sarje, A.K.: A predicted region based cache replacement policy for location dependent data in mobile environment. In: 10th Inter-Research-Institute Student Seminar in Computer Science, vol. 7, no. 2, pp. 1–8. IIIT, Hyderabad (2008)
32. Kumar, A., Misra, M., Sarje, A.K.: A weighted cache replacement policy for location dependent data in mobile environments. In: Proceedings of the 2007 ACM Symposium on Applied Computing, SAC 2007, Seoul, Republic, Korea, vol. 7, no. 3, pp. 920–924 (2007)
33. Kumar, A., Misra, M., Sarje, A.K.: A new cost function based cache replacement policy for location dependent data in mobile environment. In: 5th Annual Inter Research Institute Student Seminar in Computer Science, vol. 5, no. 1, pp. 1–8. Indian Institute Technology, Kanpur (2006)
34. Xu, J., Zheng, B., Lee, W.C., Lee, D.L.: The D-tree: An index structure for planar point queries in location- based wireless services. IEEE Trans. Knowl. Data Eng. **16**(12), 1526–1542 (2004)
35. Huang, B., Wu, Q.: A spatial indexing approach for high performance location based services. J. Navig. **60**(1), 83–93 (2007)
36. Jeong, J., Lee, K., Abdikamalov, B., Lee, K., Chong, S.: TravelMiner: on the benefit of path-based mobility prediction. In: 2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), London, UK, pp. 1–9 (2016)
37. Luo, X., Camp, T., Navidi, W.: Predictive methods for location services in mobile ad-hoc networks. IEEE Ubiquit. Comput. **3**(4), 99–107 (2012)
38. Ying, J.J., Lee, W., Tseng, V.S.: Mining geographic-temporal-semantic patterns in trajectories for location prediction. ACM Trans. Intell. Syst. Technol. **5**(1), 1–34 (2013)
39. Ying, J.J.-C., Lee, W.-C., Weng, T.-C., Tseng, V.S.: Semantic trajectory mining for location prediction. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS 2011, pp. 34–40 (2011)
40. Körner, C., May, M., Wrobel, S.: Spatiotemporal modeling and analysis—introduction and overview. KI-Künstliche Intelligenz **26**(3), 215–221 (2012)
41. Morzy, M.: Mining frequent trajectories of moving objects for location prediction. In: Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition, MLDM 2007, pp. 667–680 (2007)
42. Lu, E.H.C., Tseng, V.S., Yu, P.S.: Mining cluster-based temporal mobile sequential patterns in location-based service environments. IEEE Trans. Knowl. Data Eng. **23**(6), 914–927 (2011)
43. Gupta, A.K., Prakash, S.: Secure communication in cluster-based ad hoc networks: a review. In: Lobiyal, D.K., Mansotra, V., Singh, U. (eds.) Next-Generation Networks. AISC, vol. 638, pp. 537–545. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-6005-2_54
44. Gupta, A.K., Shanker, U.: Recent research trends on location dependent information systems. In: Springer International Conference on Pattern Recognition Techniques (ICPR 2017). Ambedkar Institute of Advanced Communication Technologies and Research, New Delhi, 22–23 December 2017

45. Gupta, A.K., Shanker, U.: SPMC-CRP: a cache replacement policy for location dependent data in mobile environment. In: Proceedings of the 6th International Conference on Smart Computing and Communications (ICSCC 2017), 7–8 December 2017, pp. 632–639. NIT, Kurukshetra (2017). https://doi.org/10.1016/j.procs.2017.12.081
46. Gupta, A.K., Shanker, U.: Modified predicted region based cache replacement policy for location dependent data in mobile environment. In: Proceedings of the 6th International Conference on Smart Computing and Communications (ICSCC 2017), 7–8 December 2017, pp. 917–924. NIT, Kurukshetra (2017). https://doi.org/10.1016/j.procs.2017.12.117
47. Gupta, A.K., Shanker, U.: SPMC-PRRP: a predicted region based cache replacement policy. In: Proceedings of the International Conference on Data and Information System (ICDIS-2017). Indira Gandhi National Tribal University (IGNTU), Amarkantak, 17–18 November 2017

Shapelets-Based Intrusion Detection for Protection Traffic Flooding Attacks

Yunbin Kim[1], Jaewon Sa[1], Sunwook Kim[2], and Sungju Lee[1](✉)

[1] Department of Computer Information Science,
Korea University, Sejong 30019, Korea
{kyb2629, sjwon92, peacfeel}@korea.ac.kr
[2] Department of SW Contents Research Laboratory,
ETRI, Daejeon, Korea
swkim@etri.re.kr

Abstract. The intrusion detection for the network traffic is a technique to detect abnormal traffic flow patterns in periodic network packets. The traffic flooding attacks can be detected by the abnormal intrusion detection techniques that detects well known attack patterns. In this paper, we propose an intrusion detection way to classify normal and abnormal traffic packet pattern by converting traffic into time series data and analyzing them, and apply the information gain technique to reduce the learning execution times. That is, the normal and abnormal packet patterns are classified by applying the shapelets technique to the time-series pattern between the normal traffic and the abnormal traffic packet patterns. The experimental results show that the proposed method classifies normal patterns and traffic flooding attacks into 95% accuracy.

Keywords: Bigdata analysis · Intrusion detection · Traffic flooding attack Time-series analysis · Shapletes analysis

1 Introduction

The big data analysis and processing are an important issue with increased IoT devices, and many researches have been studied by using machine learning, parallel processing, cloud computing, and sensors [1–5]. Recently, as IT technology has become more popular, personalized service is provided to users through various technologies including prediction model design in various fields, and reliability of personal information including network security has been highlighted accordingly [2]. In particular, big data, including sensitive data such as personal information, personally identifiable information, and intellectual property are likely to be exposed to cyber attacks and hacking. Therefore, it is very important and difficult to establish effective security for Big Data systems and services. In the network intrusion detection technology, the intrusion model can be divided into two types of normal and abnormal traffics. The anomaly based detection technique can generate a profile of a user's general pattern and analyzes patterns [6]. Therefore, the intrusion detection system can classify a pattern obtained from past intrusions [6].

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 227–238, 2018.
https://doi.org/10.1007/978-3-319-91455-8_20

DDoS (Distributed Denial of Service) attack, which is one kind of abnormal infiltration analyzed by abnormal-based intrusion detection technology, creates a large number of zombie PC remotely, and uses it to increase the traffic exponentially. In addition, the cases of traffic flooding attacks are continuously increasing, and thus an efficient detection technique for such abnormal intrusion attacks is required.

In this paper, we propose an intrusion detection way to detect and classify abnormal intrusion attacks by transforming normal and abnormal traffic packet patterns into time-series pattern, and apply shapelets analysis technique to transformed time-series data. The intrusion detection refers to analyzing and classifying network traffic data and structure abnormally. Also, we apply the information gain technique to features selection to reduce the learning execution times. Based on the experimental result, we confirmed that, proposed approach can provide 95% accuracy by using information gain and shapelets analysis.

The rest of this paper is structured as follows. Section 2 describes the related works on researches for the intrusion detection systems and the time-series data analysis methods. Section 3 describes packet characteristics of traffic flooding attacks and measurement methods, and how to apply the shapelets analysis and reduce the features by using information gain technique. Sections 4 and 5 describe the experimental results and conclusions.

2 Background

2.1 Intrusion Detection System

The intrusion detection systems protect the computer and mobile devices. Static analysis methods run the file and examine the contents of the file. Moser et al. [7] uses a number of virus/malware approaches such as bus conversion, noxiousness and variant techniques. Siddiqui et al. [8] protected the file system using the file function, which N-grams are sequences of bytes of a certain length, and contain bytes adjacent to each other [9]. Wavelet transform [10] is another source of file functionality. Bilar [11] proposed the mnemonic of the instruction using the predictor of the malicious program. Statistical machine learning and data science methods [12] have been increasingly used for malware detection, including approaches based on support vector machines, logistic regression, Naïve Bayes, neural networks, deep learning, wavelet trans-forms, decision trees and k-nearest neighbors [8, 10, 13–19].

The entropy analysis [10, 16, 20–22] is an effective technique for abnormal data detection by pointing to the possible 6080 presence of deception techniques. Despite polymorphism or obfuscation [23], files with high entropy are more likely to have encrypted sections in them. When an abnormal data switches between content regimes (*i.e.*, native code, encrypted section, compressed section, text, padding), there are corresponding shifts in its entropy time series [10]. In general, entropy analysis of data for intrusion detection, either the mean entropy of the entire data, or the entropy of chunks of code in sections of the file are computed. This simplistic entropy statistics approach may not be sufficient to detect expertly hidden malware, which for instance, may have additional padding (zero entropy chunks) to pass through high entropy filters.

2.2 Time-Series Clustering of Network Traffic

According to Keogh [24], clustering of time series can be categorized into two categories: full clustering and sub-clustering. Full clustering refers to grouping many individual time series into similar clusters or classes. Subsequence clustering refers to the use of sliding windows to extract subsequences from a single time series, and clustering is applied to the extracted subsequences. One of the most widely used approaches is hierarchical clustering. A similarity measure (*i.e.*, Euclidean distance) is applied to generate a pairwise distance matrix of primitive data. This approach generally applies to time series with the same length, but dynamic time warping (DTW) can be applied as a similarity measure to handle variable length time [25, 26]. The generation of the distance matrix is typically a computationally expensive operation for long time series [27]. Other widely used clustering algorithms such as K-means can also be applied to raw data [28]. By applying transformations to reduce the dimensionality of the data, as opposed to performing clustering on raw information, you can reduce the complexity of time series clustering. The purpose of the transformation is to first extract a specific function from the data, then apply a similarity measure and use the result as input to the clustering algorithm. In [27], the authors propose global feature extraction from individual time series (*i.e.*, trend, periodicity, and kurtosis, etc.). Time-series with similar global characteristics are clustered together. In [29], clustering is performed on the histogram representation of the data. Other transforms such as DFT (Discrete Fourier Transform) [30, 31], SVD (Singular Value Decomposition) [32] and APCA (Adaptive Piecewise Constant Approximation) [33] have also been proposed.

These transformations can extract the global properties of the time series. The main disadvantage of these approaches is the fact that when the local shape similarity is fundamental (*i.e.*, a sequence of signal strength measurements), the overall characteristics are not sufficient to adequately distinguish the time series. The use of wavelets has been discussed in the literature as a dimensional reduction technique that enables the extraction of localized shape features in the time domain [31, 34–37]. Transformations such as DFT can determine all spectral components in a time series, but cannot determine when these spectral components are present in the data (*i.e.*, time localization of features is not possible). The wavelet decomposition is provided to solve this problem. Recently, shape transformations have been proposed as approaches to cluster time series according to the shape [38–40, 44]. With this approach, a series of shapes (*i.e.*, subsequences with high discrimination power) are extracted from the time series collection.

2.3 Shapelets-Based on Time-Series Analysis

The shapelet is defined as a subsequence of one-time series in [38]. The subsequence S of length L is defined as a subset of one continuous value from the time series. The shapelets are selected by capturing unique shape features that are com-mon in time series classes. The shapelets can be found in L through a search of all possible subsequences of each time series as candidates for the shapelets [44]. However, this process is time consuming and a more efficient technique for shapelets generation has been proposed [38, 39]. The process of discovering shapelets for time-series clustering involves three main steps: creating candidates, measuring similarities between candidates and time series, and finally evaluating the quality of candidates. Regarding the generation of a shapelets candidates, it is first necessary to define the length of the candidate subsequence. Generally, a subsequence with a length between the predefined values l_{min} and l_{max} is considered. Using a generic search to generate the shapelets candidates, all possible subsequences with lengths between l_{min} and l_{max} are extracted from the time series of L. This process is slow and inefficient for large time series sets with long lengths. Rather than applying exhaustive search to create shapelets, we apply the algorithm proposed by Zakaria et al. In [39], we have made some modifications to accommodate the fact that we deal with time series of different lengths. [39], the authors proposed the use of unchecked it to collect time series.

3 Proposed Methods

In this paper, we use the NSL-KDD dataset, a quantified version of KDD CUP'99. The NSL-KDD dataset needs to extract useful features because it contains irrelevant data, redundant data, and noise data.

For this reason, data dimension reduction is performed through the information gain technique of feature selection. As a result of feature selection, the top 10 features with high weight are extracted and classified into normal and abnormal data by applying it to Shapelets, a machine learning technique.

Figure 1 shows the overall configuration of the proposed method.

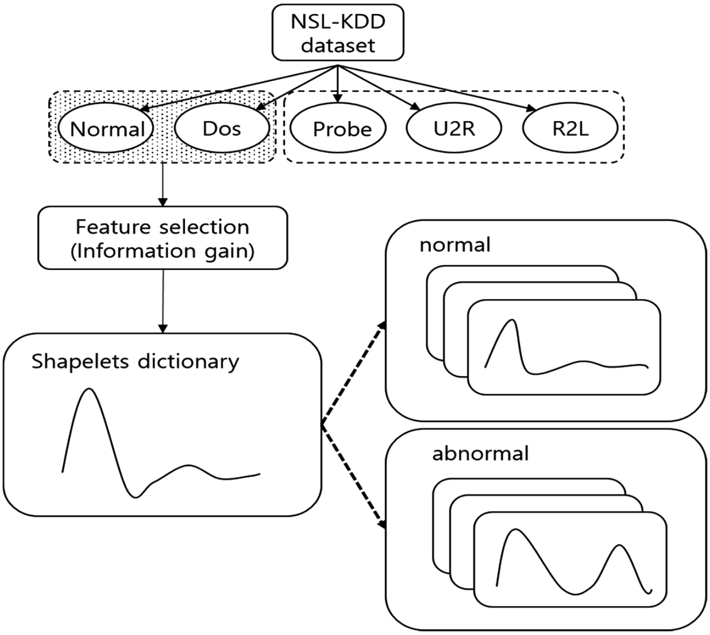


Fig. 1. The overall structure of the proposed method.

3.1 DoS Attack Detection Using Time Series Pattern

NSL-KDD [43] is a refined version of the data set KDD CUP'99, which is part of the DARPA scheme. It has four types of attacks: normal traffic packets and abnormal packets on the real network. Each attack consists of 41 features. In addition, the four attack types consist of DoS, Probe, U2R, and R2L. It also supports learning and test datasets, and the dataset features include protocol type, service, src_byte, and dst_byte, which are the contents of the network header, and access details such as host and guest logins. In this paper, we focus on traffic flooding attack using shapelets based time series analysis and use DoS attack traffic dataset among four attack types.

Table 1 shows the distribution of packets by four attack types (*i.e.*, DoS, Probe, U2R, R2L) provided by the NSL-KDD dataset.

Table 1. Distribution of packets by attack type.

Types of attack	# of packets	Ratio distribution of packet (%)
DoS	50,943	71
Probe	18,216	25
U2R	72	1
R2L	2,231	3
Sum of total	71,462	100

The four types of attacks provided by NSL-KDD are as follow.

- DoS: Denial of Service Attack, it is an attack that maliciously attacks a system and causes the system to run out of resources, thereby preventing its intended use.
- Probe: An attack that collects system vulnerabilities before attempting an actual attack.
- U2R: User to Root, it is an attack that attempts to gain administrator privileges.
- R2L: Remote to Local, an attack where an unauthorized user gains access from outside.

3.2 Reducing the Feature by Using Information Gain

The training data set of NSL-KDD consists of 4,756,832 packets. In the case of intrusion detection, it is necessary to learn about the types of attacks added periodically in order to detect new attack types. Therefore, a method for reducing the execution time of learning is needed. That is, when all the data sets are used, the accuracy of the overfitting may be reduced and the learning time may be increased.

The entropy is used for numerical operations to find the best conditions for separating data. This means the traffic flooding packets generated from the data set. If a given data set contains a lot of different types of values, the entropy is high, and if it is distributed over the same types of values, the entropy is set low. The entropy is used for numerical operations to find the best conditions for separating data. If there are many different kinds of results in a given data set, entropy is high, and entropy is low if the same kind of results exist. The entropy has a value between 0 and 1, that is, when entropy is 0, only the same types of data exists, and when entropy is 1, it is not separated at all.

$$E(S) = -\sum\nolimits_{x \in X} p(x) \log_2 p(x) \tag{1}$$

$$p(x) = \frac{freq(S_x)}{|S|} \tag{2}$$

The entropy calculated using Eq. (1) can be used to calculate a value of information gain that can distinguish data with high discrimination power.

$$Information\,Gain(S) = E_{high_{level}}(S) - \sum\nolimits_{t \in T} p(t) E_{low_{level}}(t) \tag{3}$$

$E_{high\_level}(S)$ is the entropy of a parent node, so that the information gain is the entropy of the parent node minus the entropy of the child node, taking into account the weights proportional to the number of records in the lower node.

In this paper, to solve the problem of decreasing the accuracy and increasing the learning time according to over-sum, we applied the information gain method [46] to select the top 10 features among the 41 features and apply it to the time-series analysis method. Table 2 lists the top 10 information gains used in this paper.

Table 2. Information gain-based feature selection.

No.	Features	Information gain score (Priority score)
1	Src_bytes	1.345491
2	Service	1.097208
3	Flag	0.962485
4	Diff_srv_rate	0.947248
5	Dst_host_diff_srv_rate	0.886182
6	Same_srv_rate	0.878879
7	Count	0.820505
8	Dst_host_same_srv_rate	0.800392
9	Dst_host_srv_count	0.777514
10	Dst_bytes	0.722053

4 Experimental Results

The shapelets technique is used to classify time series patterns and classifies them into several classes using subsequences extracted between time series patterns [41, 42, 44]. That is, the Euclidean distance is calculated for each traffic time series pattern using the extracted representative subsequence by learning several traffic time series patterns. Then, normal and abnormal binary classification is performed according to the criterion of the threshold value with respect to the calculated distance.

To effectively reduce the learning and test execution time of the shapelets technique, Fast-shapelets technique [41, 42, 44, 45] was applied and the shapelets were created to maximize the distance between normal and abnormal pattern classes.

After generating the shape through the Euclidean distance method in the normal traffic class and the abnormal traffic class, binary classification was performed based on the threshold values of the abnormal traffic class and the generated shapelets. We used labeled training and testing data sets to distinguish between normal and abnormal data. The training set consisted of 67,343 normal and 43,281 abnormal data, and the test set consisted of 9,710 normal and 5,076 abnormal data. In Fig. 1, each feature is set on the x-axis, and the packet value is shown on the y-axis (Fig. 2).

To classify the two labeled classes, a subsequence (i.e., Shapelet) was extracted through a learning process. It is confirmed that they are classified by mapping to normal and abnormal data using the extracted Shapelet. In this paper, we classify traffic flooding attack (*i.e.*, DoS) packet and normal data packet using shapelets algorithm, and confirm the classification process through the following decision tree.

In order to show the accuracy performance of applying the proposed approach to NSL-KDD packet data, we define the following three accuracy metrics (Fig. 3).

$$precision = \frac{TP}{TP + FP} \tag{4}$$

$$recall = \frac{TP}{TP + FN} \tag{5}$$

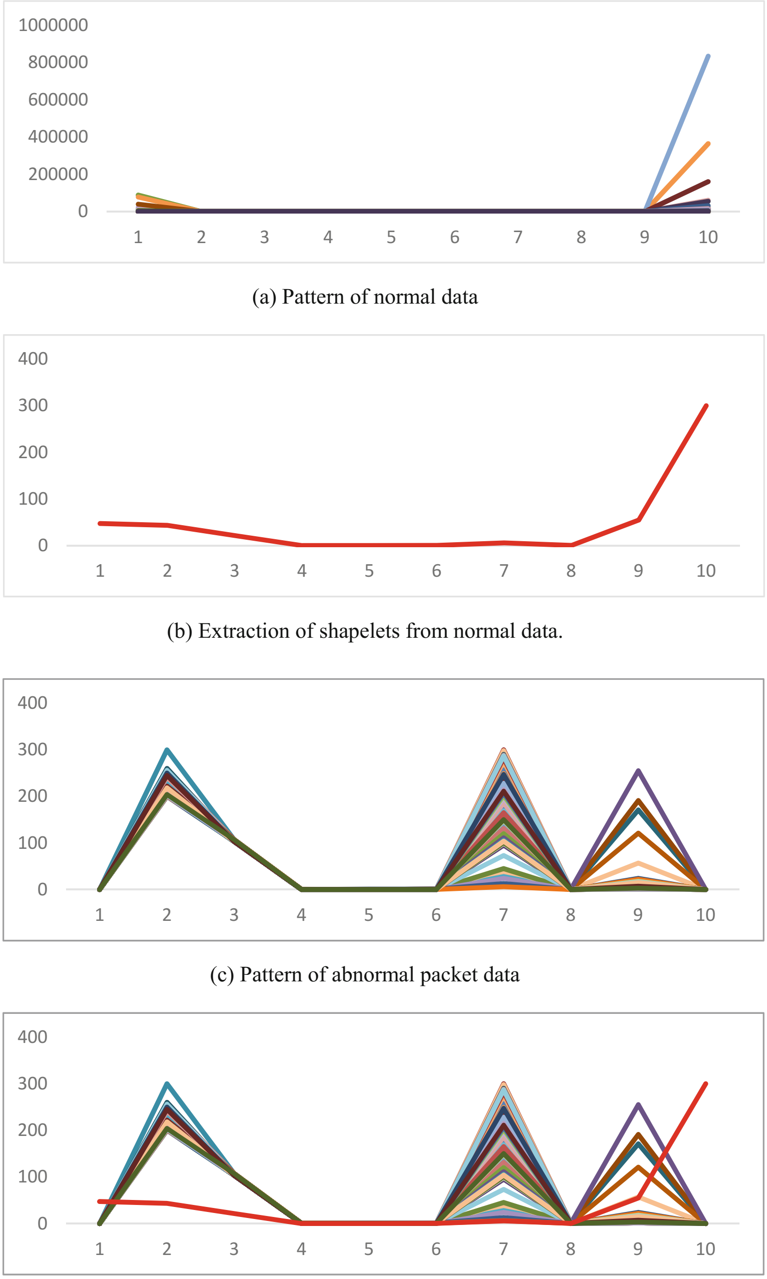
$$acurracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

Experiments in this paper were performed on Intel Core i5-4690 3.5 GHz, 8 GB RAM environment. The data used in the experiment are the NSL-KDD data set, and the classifier for packet classification is shapelets.

In order to verify the validity of the proposed method, we conducted a comparative experiment with SVM, which is a typical technique in machine learning algorithms. The ratio of training and test data was constructed in the same way as the proposed method, and the kernel function used RBF (*i.e.*, Radial basis function).

Table 3 shows the metrics for classification of normal and abnormal packets for the SVM and the proposed method. SVM and the proposed method were verified through the *precision*, *recall*, and *accuracy*, which are measures to judge classification accuracy. The *precision* is the ratio of the number of the normal packets detected to the actual number of packets, and the *recall* is the ratio of the number of normal packets detected by the algorithm among the actual packets. And *accuracy* means total accuracy.

Table 4 shows the execution time of the learning by reducing the number of features through Information Gain. As a result of reducing the number of features, it is confirmed that the performance improvement is about 25 times higher than the execution time using all 41 features.



(a) Pattern of normal data

(b) Extraction of shapelets from normal data.

(c) Pattern of abnormal packet data

(d) Abnormal packet data and the shapelets.

Fig. 2. Pattern of the normal data and abnormal attack packet. (a) shows the normal data packet that keeps a low value and records a high value in the tenth feature (Dst_byte), and (b) shows an extraction of shapelets from normal data, (c) shows abnormal packet data that records a high value in for a second and seventh features, and (d) shows that the shapelets extracted from the normal data is applied to the abnormal packet data.

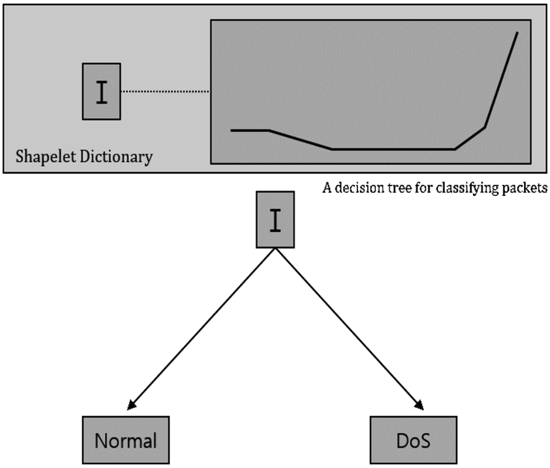


Fig. 3. The shapelets dictionary and decision tree for classifying packets: the representative subsequence shapelets is extracted from the learned normal data to classify the two classes (Normal, and traffic flooding packets, *i.e.*, DoS).

Table 3. Classification accuracy of SVM and proposed methods.

Rating scale	Result (%)	
	SVM	Proposed methods
Precision	98.4	96
Recall	95.6	97
Accuracy	96.3	95

Table 4. Computational time according to number of features.

	# of features	Learning time (sec.)
Non-feature selection	41	428
Feature selection	10	17

5 Conclusions

The intrusion detection system is an important technique that sets the criteria for reliability and validity for hosts that support the network services. In this paper, we proposed a shapelets technique for detecting abnormal traffic based on traffic flooding attack and confirmed that the classification accuracy was about 95%. Also, we confirmed that there is a 25 times improvement in the performance time by reducing the number of features with information gain technique. In the future works, we will conduct research on real-time attack detection by classifying each attack technique and reducing the execution time.

Acknowledgment. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2015R1C1A1A02037688).

References

1. Chung, Y., Lee, S., Jeon, T., Park, D.: Fast video encryption using the H.264 error propagation property for smart mobile devices. Sensors **15**(4), 7953–7968 (2015)
2. Lee, S., Jeong, T.: Forecasting purpose data analysis and methodology comparison of neural model perspective. Symmetry **9**(7), 108 (2017)
3. Lee, S., Kim, H., Chung, Y., Park, D.: Energy efficient image/video data transmission on commercial multi-core processors. Sensors **12**(11), 14647–14670 (2012)
4. Lee, S., Kim, H., Sa, J., Park, B., Chung, Y.: Real-time processing for intelligent-surveillance applications. IEICE Electr. Express **14**(8), 20170227 (2017)
5. Lee, S., Jeong, T.: Cloud-based parameter-driven statistical services and resource allocation in a heterogeneous platform on enterprise environment. Symmetry **8**(10), 103 (2016)
6. Depren, O., Topallar, M., Anarim, E., Ciliz, M.K.: An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert Syst. Appl. **29**(4), 713–722 (2005)
7. Moser, A., Kruegel, C., Kirda, E.: Limits of static analysis for malware detection. In: 23rd Computer Security Applications Conference, ACSAC 2007, pp. 421–430. IEEE, Miami Beach (2007)
8. Siddiqui, M., Wang, M.C., Lee, J.: A survey of data mining techniques for malware detection using file features. In: 46th Conference Proceedings on xx, pp. 509–510. ACM, Alabama (2008)
9. Tahan, G., Rokach, L., Shahar, Y.: Mal-ID: Automatic malware detection using common segment analysis and meta-features. J. Mach. Learn. Res. **13**, 949–979 (2012)
10. Wojnowicz, M., Chisholm, G., Wolff, M., Zhao, X.: Wavelet decomposition of software entropy reveals symptoms of malicious code. J. Innovation Digit. Ecosyst. **3**(2), 130–140 (2016)
11. Bilar, D.: Opcodes as predictor for malware. Int. J. Electr. Secur. Digit. Forensics **1**(2), 156–168 (2007)
12. Friedman, J., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning, vol. 1, pp. 337–387. Springer, New York (2001). https://doi.org/10.1007/978-0-387-21606-5
13. Alazab, M., Venkatraman, S., Watters, P., Alazab, M.: Zero-day malware detection based on supervised learning algorithms of API call signatures. In: 9th International Conference Proceedings on Australasian Data Mining, vol. 121, pp. 171–182. Australian Computer Society, Ballarat (2011)
14. Davis, A., Wolff, M.: Deep Learning on Disassembly Data. In: Black Hat, USA (2015)
15. Kolter, J.Z., Maloof, M.A.: Learning to detect malicious executables in the wild. In: 10th ACM SIGKDD International Conference Proceedings on Knowledge Discovery and Data Mining, pp. 470–478. ACM (2004)
16. Lyda, R., Hamrock, J.: Using entropy analysis to find encrypted and packed malware. IEEE Secur. Priv. **5**(2), 40–45 (2007)
17. Schultz, M.G., Eskin, E., Zadok, F., Stolfo, S.J.: Data mining methods for detection of new malicious executables. In: Conference Proceedings on Security and Privacy, 2001 IEEE Symposium, pp. 38–49. IEEE, Oakland (2001)

18. Shabtai, A., Moskovitch, R., Elovici, Y., Glezer, C.: Detection of malicious code by applying machine learning classifiers on static features: a state-of-the-art survey. Elsevier **14**(1), 16–29 (2009)
19. Shafiq, M.Z., Tabish, S.M., Mirza, F., Farooq, M.: PE-Miner: mining structural information to detect malicious executables in realtime. In: Kirda, E., Jha, S., Balzarotti, D. (eds.) RAID 2009. LNCS, vol. 5758, pp. 121–141. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04342-0_7
20. Baysa, D., Low, R.M., Stamp, M.: Structural entropy and metamorphic malware. J. Comput. Virol. Hacking Tech. **9**(4), 179–192 (2013)
21. Sorokin, I.: Comparing files using structural entropy. J. Comput. Virol. **7**(4), 259 (2011)
22. Wojnowicz, M., Chisholm, G., Wolff, M.: Suspiciously structured entropy: wavelet decomposition of software entropy reveals symptoms of malware in the energy spectrum. In: International Conference Proceedings on FLAIRS, pp. 294–298 (2016)
23. O'Kane, P., Sezer, S., McLaughlin, K.: Obfuscation: the hidden malware. IEEE Secur. Priv. **9**(5), 41–47 (2011)
24. Keogh, E., Lin, J.: Clustering of time-series subsequences is meaningless: implications for previous and future research. Knowl. Inf. Syst. **8**(2), 154–177 (2005)
25. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: International Conference Proceedings on Discovery Data Mining, vol. 10, pp. 359–370 (1994)
26. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. Knowl. Inf. Syst. **7**(3), 358–386 (2005)
27. Wang, X., Smith, K., Hyndman, R.: Characteristic-based clustering for time series data. Data. Min. Knowl. Discov. **13**(3), 335–364 (2006)
28. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: 5th Proceedings on Berkeley symposium, vol. 1(14), pp. 281–297 (1967)
29. Lin, J., Khade, R., Li, Y.: Rotation-invariant similarity in time series using bag-of-patterns representation. J. Intell. Inf. Syst. **39**(2), 287–315 (2012)
30. Agrawal, R., Faloutsos, C., Swami, A.: Efficient similarity search in sequence databases. In: Lomet, D.B. (ed.) FODO 1993. LNCS, vol. 730, pp. 69–84. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57301-1_5
31. Lin, J., Vlachos, M., Keogh, E., Gunopulos, D.: Iterative incremental clustering of time series. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K., Ferrari, E. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 106–122. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24741-8_8
32. Korn, F., Jagadish, H.V., Faloutsos, C.: Efficiently supporting ad hoc queries in large datasets of time sequences. In: International Conference Proceeding on Management of data, vol. 26(2), pp. 289–300. ACM, Tucson (1997)
33. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Locally adaptive dimensionality reduction for indexing large time series databases. In: International Conference Proceeding on Management of data, vol. 30(2), pp. 151–162. ACM, Santa Barbara (2001)
34. Chan, K.P., Fu, A.W.C.: Efficient time series matching by wavelets. In: 15th International Conference Proceedings on Data Engineering, pp. 126–133. IEEE, Sydney (1999)
35. Popivanov, I., Miller, R.J.: Similarity search over time-series data using wavelets. In: 18th International Conference Proceeding on Data Engineering, pp. 212–221. IEEE, San Jose (2002)
36. Vlachos, M., Lin, J., Keogh, E., Gunopulos, D.: A wavelet-based anytime algorithm for k-means clustering of time series. In: Proceedings Workshop on Clustering High Dimensionality Data and its Applications, pp. 23–30 (2003)

37. Antoniadis, A., Brossat, X., Cugliari, J., Poggi, J.M.: Clustering functional data using wavelets. Int. J. Wavelets **11**(1), 1350003 (2013)
38. Hills, J., Lines, J., Baranauskas, E., Mapp, J., Bagnall, A.: Classification of time series by shapelet transformation. Data. Min. Knowl. Discov. **28**(4), 851–881 (2014)
39. Zakaria, J., Mueen, A., Keogh, E.: Clustering time series using unsupervised-shapelets. In: 12th International Conference Proceedings on Data Mining (ICDM), pp. 785–794. IEEE, Brussels (2012)
40. Zakaria, J., Mueen, A., Keogh, E., Young, N.: Accelerating the discovery of unsupervised-shapelets. Data. Min. Knowl. Discov. **30**(1), 243–281 (2016)
41. Patri, O., Wojnowicz, M., and Wolff, M.: Discovering malware with time series shapelets. In: 50th International Conference Proceedings on System Science, Hawaii (2017)
42. Castro-Hernandez, D., Paranjape, R.: Classification of user trajectories in LTE HetNets using unsupervised shapelets and multiresolution wavelet decomposition. IEEE Trans. Veh. Technol. **66**(9), 7934–7946 (2017)
43. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 1999 data set. In: Computational Intelligence for Security and Defense Applications, CISDA 2009, pp. 1–6. IEEE, Ottawa (2009)
44. Ye, L., Keogh, E.: Time series shapelets: a new primitive for data mining. In: 15th ACM SIGKDD International Conference Proceedings on Knowledge discovery and data mining, pp. 947–956. ACM, Paris (2009)
45. Rakthanmanon, T., Keogh, E.: Fast shapelets: a scalable algorithm for discovering time series shapelets. In: International Conference Proceedings on Data Mining, pp. 668–676. Society for Industrial and Applied Mathematics (2013)
46. Gao, Y., Feng, Y., Tan, J.: Exploratory study on cognitive information gain modeling and optimization of personalized recommendations for knowledge reuse. J. Manuf. Syst. **43**, 400–408 (2017)

Tuple Reconstruction

Ngurah Agus Sanjaya Er[1,4(✉)], Mouhamadou Lamine Ba[2],
Talel Abdessalem[1,3,4], and Stéphane Bressan[3]

[1] Télécom Paristech, Paris, France
sanjaya.agus@telecom-paristech.fr
[2] Université Alioune Diop de Bambey, Bambey, Senegal
[3] National University of Singapore, Singapore, Singapore
[4] UMI IPAL, CNRS, Paris, France

Abstract. Set of tuples expansion system (STEP) extracts information from the Web in the form of tuples. It builds a graph of entities consisting of Web pages, wrappers, seeds, domains, and candidates as its nodes while the relationships between them as edges. The final weight given for each node after running random walks on the graph is used to order the extracted candidates. Due to the nature of the regular expressions used as wrappers, some of the extracted candidates may contain "noise" and therefore can be considered as "false". These false candidates may rank higher than the "true" ones on the list because they are extracted from many Web pages or produced by many different wrappers. Minimizing these false candidates is necessary to ensure the validity of the result presented.

In this research, we propose a method to tackle the aforementioned problem of STEP by reconstructing tuples. We begin with extracting binary tuples from the Web. These binary tuples consist of a key attribute and a property of the attribute. To validate the truthfulness of the binary tuples, we apply truth-finding algorithms. This helps us in building a credible list of binary tuples. We propose two methods to reconstruct tuples from binary ones. We use the reconstructed tuples to enrich the graph of entities of STEP such that the "true" candidates receive more confidence and rank higher in the graph. We show that our approach is efficient and significantly improve the confidence level of the tuples extracted by STEP. We also conduct an experiment on a real-world case of populating a database relation from the Web with our proposed approach.

Keywords: Set expansion · Tuples · Reconstruction · Truth-finding

1 Introduction

Set of tuples expansion system, such as STEP [12], extracts information from the World Wide Web in the form of tuples. Particularly, given a set of tuple examples <*Fat Tony's*, *Kuta*, *+62 857 9264 1911*>, <*Warung Laota*, *Tuban*,

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 239–254, 2018.
https://doi.org/10.1007/978-3-319-91455-8_21

+62 361 8947490>, STEP returns a list consisting of the name of restaurants, its location, as well as the telephone number. The list is ordered based on the confidence level of the tuple candidates. Wang et al. [35] proposed the graph of entities and applied PageRank algorithm on the graph. Er et al. [12] have applied the approach on tuples. The final weight assigned to each node by PageRank is then used to rank the tuple candidates. Truth-finding algorithm can also be applied to a set of tuples candidates where only one tuple candidate is selected as the truth for each object of the dataset. In [13], the authors experimented with eleven truth-finding algorithms, including state-of-the-art such as Majority voting, TruthFinder [38], Cosine, 2-Estimates, and 3-Estimates [19], LCA [29], Depen and its variants [9], and compared their performance with PageRank. It is shown that the truth-finding algorithms outperformed PageRank in terms of average precision, recall, and F-measure.

Motivations. PageRank [28] is primarily used by Google to measure the importance of Website pages with refers to a query given by the user. Each link from other Web pages to a target Web page is treated as a vote for the target page. The Web pages are then ranked according to its corresponding vote count. In the particular case of STEP [12], a graph is built by defining a set of entities (Web pages, wrappers, seeds, domains, and tuple candidates) as nodes and the relationships between entities as links. A tuple candidate node is only linked with wrapper nodes. This means that the rank of a tuple candidate in the final list only depends on the number of wrappers used to extract that particular candidate. The wrappers in STEP are regular expression-based wrappers. They are generated by comparing the contexts of a pair of seeds in a page. This method may produce more tuple candidates, but on the other hand, it is more prone to extracting "false" ones. These false candidates may receive as many vote count as the true candidates due to the nature of the graph of entities. They may have more votes if they are extracted by many wrappers on the same page or from other wrappers from other pages. The reverse situation where true candidates have more votes than the false ones can also happen. This is the motivation of this research where we propose a solution to ensure that true candidates will always have more votes than the false ones and rank higher on the list. We explain briefly our proposed method next.

Tuples such as those given as examples earlier consist of n elements. Each element is a value for an attribute of a real-world entity. We use the term element and attribute interchangeably throughout this paper. Among these n-elements, one of them is the key attribute while all the remaining elements are the properties of the key attribute. Consider the tuple <*Fat Tony's*, *Kuta*, *+62 857 9264 1911*> where $n = 3$, the key attribute in this tuple is "Fat Tony's" while "Kuta" and "+62 857 9264 1911" are values for attributes "location" and "telephone number" of Fat Tony's respectively. These two attribute values and their key attribute are actually tuples of length $n - 1$, <*Fat Tony's*, *Kuta*> and <*Fat Tony's*, *+62 857 9264 1911*>. If we have the knowledge of these two tuples a priori, add them to the graph of entities as new nodes, and create links to the candidate tuple <*Fat Tony's*, *Kuta*, *+62 857 9264 1911*>, then the tuple of length

$n = 3$ would gain more weight after we perform the PageRank algorithm. Subsequently, the candidate tuple should also rank higher in the final list presented to the user. Based on this intuition, we propose to extract binary tuples from Web pages. To yield a trustworthy list of binary tuples, we apply several truth-finding algorithms on the extracted binary tuples. Next, we evaluate the performance of each algorithm in terms of the precision, recall, and F-measure. We select the best algorithm as a tool to verify the truthfulness of the binary tuples. From these verified binary tuples, we reconstruct tuples of n-element ($n > 2$). The reconstructed tuples are then added to the graph of entities with the purpose of enriching the graph. We show that the addition of these reconstructed tuples can improve the confidence in the extracted candidates.

Contribution. The contribution of this research is the following.

- The tuple reconstruction problem and one of its possible solution. We first define the problem and propose a method to solve the problem. We also evaluate the performance of the proposed method by means of several truth-finding algorithms.
- We enrich the graph of entities in [12] with the reconstructed tuples. We extensively evaluate the performance of PageRank in the graph before and after the addition of the reconstructed tuples. We show that this approach can significantly improve the confidence level in the list of extracted tuples.
- We experiment with an application of the proposed approach on populating database relations directly from the Web. We show that the approach is effective.

We organize the remaining of the paper as the following. We review previous research relating to our work in Sect. 2. Our proposed approach is detailed in Sect. 3. We detail the experiments conducted and analyze the results in Sect. 4. We conclude the paper in Sect. 5.

2 Related Work

This section summarizes the related work on set expansion and entity profiling.

2.1 Set Expansion

The set expansion problem is related to finding and extracting other members of a semantic class from a data source when given some examples or seeds. To be more concrete, when given HTML pages (the data source), a set of names (the seeds) of restaurants in Kuta (the semantic class), the set expansion goal is to extract other names of restaurants in Kuta from the HTML pages. A wide range of research areas have gained the benefit of embedding set expansion systems including knowledge extraction [25], question answering [37], and vocabulary or dictionary construction [7,33]. Set expansion systems use the following framework to achieve their goal:

- **fetch relevant documents**. Retrieve relevant documents containing the seeds. The sources from which set expansion systems retrieve the relevant documents can range from encyclopedia [4], Web pages [6,12,35,36], search logs [27,40], etc.
- **infer patterns and extract candidates**. In order to be able to generate patterns, set expansion systems must first locate occurrences of each seed in each of the retrieved documents. The surrounding contexts in which the seeds occur are then compared to generate patterns (*wrappers*). Candidates are then extracted from the documents using the inferred wrappers.
- **ranking**. The last step in the framework is to apply a ranking mechanism, such as nearest neighbor [27], Bayesian sets [36], iterative thresholding [21], random walk [35], and PageRank [12,36] on the extracted candidates. Due to a large number of the extracted candidates, the ranking step seems necessary although set expansion system such as DIPRE [6] does not employ any ranking strategies.

Brin [6] proposed DIPRE where the duality between patterns and target relations is exploited. In the first step, from the occurrences of each seed in the collected documents, DIPRE infers wrappers which are used to extract new candidates. Next, the newly found candidates are used to retrieve more documents and the previous step is carried out. These two steps are repeated until no more candidates can be extracted. SEAL [35] was proposed by Wang et al. In the paper, the authors explained how to extract atomic entities from semi-structured corpus. A weighted graph model is introduced to rank the extracted candidates. The weight for each candidate is the result of applying a random walk on the graph. In [36], Wang et al. extended SEAL to extract binary relations. ER et al. [12] proposed to generalize the set expansion problem to set of tuples expansion. A set of tuples is used as input to the system where each tuple consists of more than two elements. Each element in the tuple belongs to a different semantic class while being all semantically connected in the real world. For instance, when the user gives a set of tuples <*Fat Tony's*, *Kuta*, *+62 857 9264 1911*>, <*Warung Laota*, *Tuban*, *+62 361 8947490*> a system implementing the approach returns relational instances composed of the name of restaurants in Bali with their corresponding location and telephone number. In [13], the authors apply truth-finding algorithms to the set of extracted candidates and compare the performance with PageRank. It has been proven that the level of confidence of the extracted candidates is significantly improved.

2.2 Entity Profiling

Constructing a structured profile for a real-world entity combines techniques used for information extraction and data integration. *Information extraction* is the task consisting of extracting entity-related structured facts or records from unstructured data (e.g., free texts) or semi-structured data (e.g., web tables). Finding and extracting such data have gained interest in the research community. Abdessalem et al. [1] proposed ObjectRunner where a user can specify the target data freely. To extract the data from Web pages, ObjectRunner leverages not only the description of the input but also the page structure. More details of the system, as well as the experiments conducted, can be found in [8]. DIADEM [18] is a large-scale extractor of structured data. Phenomenological and ontological knowledge is integrated into the system to explore potential Websites, identify structured data, and automatically infer wrappers. In WADar [26], the inferred wrapper is continuously improved by observing the data extracted from previous runs. Other proposed solutions are specifically designed to extract product specifications from Web pages [31] or to extract structured data from content management systems [15], emails [39], etc.

The different pieces of extracted facts from different sources can be grouped and then integrated to build a complete profile related to each entity. This is the task of *data integration*. Since the facts extracted from various sources can come from different schemas, attributed with different names, and have different values, data integration usually consists of three steps to resolve each of the previously mentioned task: *schema mapping*, *record linkage*, and *truth-finding*. Schema matching related to the task of finding specifications to describe the relationships between data in two heterogeneous schemas [14]. In [32], the authors review existing approaches of schema matching and differentiate them based on the matchers. Record linkage also referred to as entity resolution and reference reconciliation deals with the problem of identifying data records that refer to the same real-world entity. A survey of recent approaches can be found in [20], while [3] studies the theoretical properties, and [22] evaluates the existing approaches to real-world problems. Truth-finding also referred to as fact-checking, truth discovery [10,38,41,42], and data fusion [5,11,23,30], tries to tackle the problem of deciding the true information as well as trustworthy sources automatically. It achieves this goal by analyzing the data values provided, source overlapping, and conflicts. Various application domains such as data integration [9,41], information retrieval [2], Open Linked Data [24], and set expansion [13] have gained the benefit of applying truth-finding approach. Majority voting is the earliest approach proposed for truth-finding and has since been widely adopted. Recent approaches, however, have implemented an iterative process to advance the traditional models. These advance approaches fall into one of these categories: *agreement-based methods*, including Cosine, 2-Estimates and 3-Estimates [19], TruthFinder [38], uses the notion of majority to iteratively computes source trustworthiness and value confidence score until convergence, *MAP Estimation-based methods*, including MLE [34], LTM [41], LCA models [29], is built on Maximum A Posteriori paradigm, and *Bayesian Inference-based methods*, including four variants of Depen models [9], SmartMTD [16,17] is based on a bayesian analysis.

3 Tuple Reconstruction

In this section, we show our approach of reconstructing tuples. We consider fixed finite sets of *attribute labels* $\mathscr{A}$ and *values* $\mathscr{V}$. We formally defined the following.

Definition 1. *A* tuple *t consists of a set of v where v is a mapping $v : \mathscr{A} \rightarrow \mathscr{V}$.*

To form a tuple, we must first select a set of attributes for a real-world entity. For instance, we select the entity to be a restaurant, and for this object, we choose $n = 3$ attributes which include the name of the restaurant, the location, and the telephone number. Each of these attributes can take exactly one value from $\mathscr{V}$ domain. If we take "Fat Tony's", "Kuta", and "+62 847 9264 1911" as the values for the attributes then we form the tuple <*Fat Tony's*, *Kuta*, *+62 857 9264 1911*>.
Let n and i denote the number of elements and the index of each element in a tuple respectively.

Definition 2. *The* key attribute *is the identifier of a tuple and comprised of the first $n - 1$ elements for n-element tuple where $n \geq 2$.*

The key attribute acts as the identifier of the tuple. For example, in the tuple <*Fat Tony's*, *Kuta*>, the key attribute is *Fat Tony's* where it represents the name of the restaurant.

3.1 Extracting Binary Tuples

Before we can reconstruct n-element tuples ($n > 2$), we must first extract binary tuples from Web pages. We define the attribute domain as {the name of the restaurant, the postal code, the location/city, the email address, the Website URL, Facebook page, Twitter, Instagram, GooglePlus, and Youtube channel}. Each of these binary tuples consists of the key attribute and another attribute. We apply the following procedure to extract binary tuples from the Web with the provided key attribute values.

- Run a query to the search engine (Google) using each key attribute value as the search query and collect the top-m Web pages returned by the search engine.
- To get the candidate values for attribute location from each of the Web pages collected, conduct the following.
  - Remove HTML tags from the Web page.
  - Apply named-entity recognition tagger and collect "location" tagged words.
  - Verify each "location".
- To get all candidate values for the other attributes, use the predefined regular expression for each of the attributes and search for matching texts on the Web page.
- Generate the binary tuples by combining the key attribute and an arbitrary candidate attribute value (*greedy* approach).
- Apply a truth-finding algorithm on the set of binary tuples generated from the previous step.

3.2 Generating Tuples

Let n and i denote the number of elements and the index of each element in a tuple respectively. As we described briefly in Sect. 1, an n-element tuple can be constructed from a pair of $(n-1)$-element tuples for $n > 2$. These two $(n-1)$-element tuples should have the same elements ordered from index $i = 1$ to $n-2$. In general, a pair of n-element tuples can be *merged* to construct an $(n+1)$-tuple if they both share the same $n-1$ elements. Consider the following two tuples <*Fat Tony's*, *Kuta*> and <*Fat Tony's*, *80361*> wherein both tuples $n = 2$. The two tuples can be combined to form <*Fat Tony's*, *Kuta*, *80361*>. If, for instance, we generate another $n = 3$ tuple <*Fat Tony's*, *Kuta*, *+62 857 9264 1911*> using the same procedure, then the two tuples can be combined to generate an $n = 4$ tuple <*Fat Tony's*, *Kuta*, *80361*, *+62 857 9264 1911*>. The process is repeated until there is no more $n+1$ tuple can be generated from n tuples. This tuple reconstruction process is similar to the process of generating *lattice* in association rules mining. We refer to the tuples generated from this process as *reconstructed tuples.*

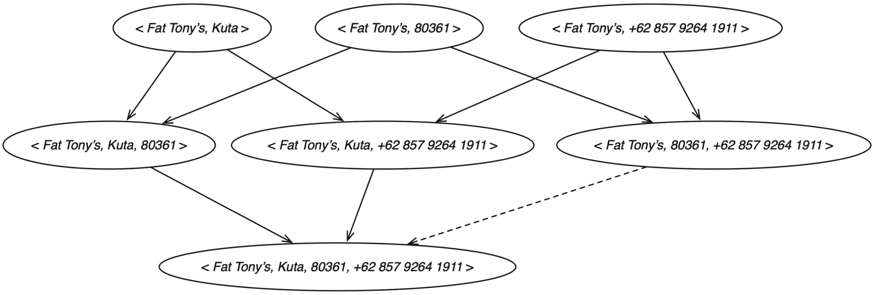

< Fat Tony's, Kuta >
< Fat Tony's, 80361 >
< Fat Tony's, +62 857 9264 1911 >
< Fat Tony's, Kuta, 80361 >
< Fat Tony's, Kuta, +62 857 9264 1911 >
< Fat Tony's, 80361, +62 857 9264 1911 >
< Fat Tony's, Kuta, 80361, +62 857 9264 1911 >

Fig. 1. Tuple reconstruction illustration

Generating tuples with the previously explained method is quite *strict* because the two n-element tuples to be combined must have the same $n-1$ elements as its prefix. Consider Fig. 1 where on the first level of the graph we have three binary tuples <*Fat Tony's*, *Kuta*>, <*Fat Tony's*, *80361*>, and <*Fat Tony's*, *+62 857 9264 1911*>. The binary tuples can be combined to generate the nodes in the second level. Note that two of the three tuples in the second level have the same key attribute. Thus, the combination of the tuples <*Fat Tony's*, *Kuta*, *80361*> with <*Fat Tony's*, *Kuta*, *+62 857 9264 1911*> yields the tuple <*Fat Tony's*, *Kuta*, *80361*, *+62 857 9264 1911*> in the third level. If we compare the previous tuple with the rightmost node of the second level, we can see that it only missing the element "80361". We can combine the middle and the rightmost node in the second level to form the tuple in the third level if we disregard the key attribute of both tuples. This method is looser and can help the candidate tuples (*nodes*) get more vote (*edges*) which in turn give more confidence on the tuples itself. We present the comparison of the *strict* and *loose* methods of generating reconstructed tuples in Table 2.

4 Performance Evaluation

We report here the result of our evaluation the performance of PageRank on the graph of entities with and without the addition of the reconstructed tuples.

4.1 Experimentation Setting

Input Datasets. We built a dataset of restaurants to intensively performed tests on our proposed approach. The restaurants dataset is constituted of the name of restaurants (the key attribute) and other essential information with refers to the restaurant such as the address, the telephone number, the postal code, the location, the Web page URL, as well as the social media (the Facebook page URL, the Twitter and Instagram account, Youtube page). We also manually constructed the baseline for the dataset from Tripadvisor (http://www.tripadvisor.com). We excluded the Web pages used as the baseline from the search results.

Evaluated Metrics. To conduct the performance evaluation we measure three metrics: precision (p), recall (r), and F-measure. To compute the metrics we use the following equations:

$$p = \frac{\sum_{i=1}^{|R|} Entity(i)}{\mid R \mid}; r = \frac{\sum_{i=1}^{|R|} Entity(i)}{|G|};\ \text{F-measure} = 2 * \frac{p * r}{p + r} \tag{1}$$

Entity(i) returns a true value if the ground truth contains the i-th candidate or false otherwise. $|R|$ and $|G|$ denotes the number of distinct candidates and the size of the ground truth respectively.

4.2 Experiments

Extracting Binary Tuples. We start our experiment by applying our approach of extracting binary tuples from Web pages. A list of 837 restaurants is used as input. We use $m = 10$ to collect the top ten Web pages returned by Google for each restaurant. For the sake of fairness, we exclude any Web pages under the domain of www.tripadvisor.com from the search results. The extracted binary tuples are then fed into eleven truth-finding algorithms including Cosine (CO), 2-Estimates (2E), 3-Estimates (3E), Depen (DP), Accu (AC), AccuSim (AS), AccuNoDep (AN), TruthFinder (TF), SimpleLCA (SL), GuessLCA (GL), and MLE [34](ML) where each of these algorithms will select the true value for each of the mentioned attributes. We then compare the selected true values for each algorithm with the baseline and calculate the precision, recall, and F-measure. We select the best algorithm based on these metrics. The result of the measurement is shown in Table 1.

Table 1. Precision, recall, and F-measure of truth-finding algorithms

	CO	2E	3E	DP	AC	AS	AN	TF	SL	GL	ML
Precision	0.4746	0.5424	0.5000	0.5212	0.5551	0.5466	0.5551	0.5551	0.5424	0.5381	0.2226
Recall	0.7887	0.9014	0.8310	0.8662	0.9225	0.9085	0.9225	0.9225	0.9014	0.8944	1.0000
F-measure	0.5926	0.6772	0.6243	0.6508	0.6931	0.6825	0.6931	0.6931	0.6772	0.6720	0.3641

Reconstructing Tuples. From the binary tuples extracted earlier, we can construct new tuples with $n = 3$. From the new tuples with $n = 3$, we can then form tuples with $n = 4$, and so on. Thus, if we have binary tuples, we can recursively construct new tuples with $n > 2$. We run the *strict* and *loose* method of reconstructing the tuples and compare the statistics of both methods. Table 2 shows the comparison of the two methods. The leftmost column indicates the number of element n, where the next column shows that we experiment with 4 and 5 elements in a tuple. The first row denotes the number of nodes (V) and edges (E) in the graph, while the second row compares the two graphs, the first is the graph without the addition of reconstructed tuples (OG) whilst the latter is the graph with the addition of the reconstructed tuples (NG). The “S” and “L” label in both of the NG columns denote the strict and loose methods of reconstructing tuples respectively.

Table 2. Statistics for methods of tuples reconstruction

		V			E		
		OG	NG		OG	NG	
n	4	13224	**S**	30348	49776	**S**	101921
			L	31137		**L**	115212
	5	16334	**S**	57048	62060	**S**	230540
			L	60914		**L**	412754

PageRank Performance on Graph of Entities. In this experiment, we are interested in evaluating the performance of PageRank on the graph of entities with the addition of the reconstructed tuples. We run the PageRank algorithm on the graph of entities twice. First, the graph of entities only consists of n-elements candidate tuples, and other entities introduced in [12]. Next, we add the reconstructed tuples with the number of elements equals to 2 until $n - 1$ recursively. We then calculate the number of “false” candidates contained in the top-k candidates of the two graphs. We present the result of the comparison in Table 3. k in the top row denotes the top-k in the list of candidate tuples, while the capital K in other rows indicates thousands. We use k in the range of 5000 until 60000 with a 5000 increment. The first two columns of Table 3 denote the same labels as in Table 2. “S” and “L” label in the third columns denote the strict and loose methods of reconstructing tuples. The fourth column in Table 3 represent the same labels as in the second row of Table 2. The value in thousand inside the brackets, for example (13K) for $n = 4$, type of tuple reconstruction is strict (S), and in the graph without the reconstructed tuples (OG), denote the number of the extracted tuples ($|R| = 13000$).

Table 3. False candidates contained in the top-k list

				Top-k										
				5K	10K	15K	20K	25K	30K	35K	40K	45K	50K	60K
n	4	S	OG	0	0	630 (13K)	*	*	*	*	*	*	*	*
			NG	0	0	0	0	0	630 (30K)	*	*	*	*	*
		L	OG	0	0	630	630	630	630	630 (31K)	*	*	*	*
			NG	0	0	0	0	0	0	630 (31K)	*	*	*	*
	5	S	OG	0	0	0	865	865	865	865	865	865	865	865 (57K)
			NG	0	0	0	0	0	0	0	0	0	0	865 (57K)
		L	OG	0	0	0	865	865	865	865	865	865	865	865 (61K)
			NG	0	0	0	0	0	0	0	0	0	0	206 (61K)

4.3 Result Analysis

Binary Tuples Extraction Analysis. From Table 1, we can see that three algorithms (Accu, AccuNoDep, and TruthFinder) are the best in terms of precision and F-measure. AccuNoDep leaves out the assumption of source dependence and because it receives exactly the same scores as Accu, we can conclude that there is no dependency between sources from which we extract the tuples. This is also supported by the fact that TruthFinder, which uses a dampening factor to address the possibility of source dependence, also has the same scores. However, it has been proven that the use of the dampening factor is not necessarily effective [9]. Thus, we select Accu as the algorithm to use in validating the binary tuples extracted from the Web.

Reconstructing Tuples Analysis. Table 2 shows the statistic of the proposed methods of reconstructing tuples. Both of the proposed methods add a significant amount of new nodes and edges into the graph. Compared to the original graph (OG), the tuple reconstruction process almost tripled the nodes in the new graph (NG) for both $n = 4$ and 5. For the number of edges, the new graph has twice as many as the original graph for $n = 4$, while for $n = 5$ it is five to six times larger. This large number of new nodes and edges contributes to the fact that the false candidates rank lower in the new graph as opposed to the original one. We detail the evaluation next.

PageRank Performance Analysis. From Table 3, we can see that the effect of introducing the reconstructed tuples and adding them into the graph of entities improve the confidence level of the extracted candidates. For each of the OG row in the table, the "false" candidates can be found quite early in the list ($k = 15000$ and 20000 for $n = 4$ and 5 respectively). For n = 4, with the strict or loose method for reconstructing tuples, the false candidates are introduced in $k = 15000$, where the total number of the extracted candidates are between 30K–31K. This means that the false candidates receive the same number of votes as half of the total candidates. In other words, we can not trust half of the list because it contains false candidates. For $n = 5$, contrary to the earlier case, the false candidates can be found nearly at the end of the list (in range 50K–57K and greater than 60K for strict and loose method respectively). However, using the loose method we can see that adding reconstructed tuples to the graph helps minimize the number of false candidates by more than 75% (865 to 206 false candidates). From this observation, we can conclude that the addition of reconstructed tuples into the graph of entities can improve the level of confidence in the list of the extracted candidates.

Table 4. False positives deduced by Accu

Object	Property	Value	Source	Label
Ma-Joly	PHONE	+62 361 753 780	http://www.ma-joly.com/	False
Ma-Joly	PHONE	+62 361 753 781	http://weddingsatmajoly.com/	True
Ma-Joly	PHONE	+62 878 6081 4531	http://www.bali-indonesia.com/magazine/ma-joly-restaurant.htm	False

Micro-Analysis. The tuple reconstruction process depends heavily on the binary tuples extracted from the Web. As explained earlier, we apply eleven state-of-the-art truth-finding algorithms on the extracted binary tuples. We also concluded that Accu is the best truth-finding algorithm for the task. Nevertheless, we encountered some cases where Accu deduce "false positive". Let us take a look at Table 4. From the table, we can see that there are three sources where each of the sources contributes a single fact for attribute "PHONE" of an object "Ma-Joly". The value "+62 361 753781" is considered as the "true" value for the phone attribute. This is actually incorrect because from the official Website of Ma-Joly we know that the phone number is "+62 361 753780". If we use the result of the truth-finding algorithm then we would extract <*Ma-Joly, +62 361 753781*> instead of <*Ma-Joly, +62 361 753780*>. This would also affect the tuples reconstructed from the binary tuple and in the end influence the confidence in the extracted tuples of STEP. One easy way to fix this is by always prioritizing the information extracted from official Web pages. We use this intuition in our experiment on populating database relation which we will detail next.

4.4 Populating Database Relation

We continue our running example here and conduct a simple experiment on extracting tuples for a database relation "Restaurant". The relation consists of three columns namely "Name", "Telephone", and "Address". We restrict ourselves for the purpose of the performance evaluation to extract only restaurants in the area of Kuta, Bali, Indonesia, otherwise, the domain is too large to cover. We choose TripAdvisor[1] as our baseline. There are 871 restaurants in the Kuta area according to TripAdvisor. We manually identify the telephone number, address, as well as the official Web page of each restaurant.

We give the following two tuples as examples, <*Warung Laota*, *+62 361 8947490*, *Jl. Raya Kuta 530*> and <*Made's Warung*, *+62 361 755297*, *Jl. Raya Seminyak*>. Our goal here is to populate the relation with tuples of length 3 ($n = 3$). The restaurant name is the key attribute. We then decompose these two examples tuples into two groups of binary tuples. The first group consists of <*Warung Laota*, *+62 361 8947490*> and <*Made's Warung*, *+62 361 755297*>, while the second <*Warung Laota*, *Jl. Raya Kuta 530*> and <*Made's Warung*, *Jl. Raya Seminyak*>. We run the STEP algorithm on the two groups separately. The statistics of running the algorithm on these groups is shown in Table 5.

Table 5. Binary tuples statistic

#Group	#Distinct_restaurant	#Tuples_extracted	#Distinct_source
1	783	10,910	1,739
2	765	9,595	1,523

The columns in Table 5 show the group number, the total number of distinct restaurant, the total number of tuples extracted, and the total number of distinct source respectively. From the table we can see that STEP extract more tuples in the form of the name of restaurant and telephone (#Group 1) number compared to the name of restaurant and address (#Group 2). This is quite predictable because the most common information on a Web page about a restaurant is its telephone number. This intuition is supported by the fact that in the first group STEP manages to retrieve significantly more sources (216) than the second group, thus extracts more tuples. From this statistic, we can expect to generate a maximum of 765 tuples of length 3.

The extracted candidates for each group are then used as input to the truth-finding algorithms. In [13], we experimented with eleven truth-finding algorithms using several datasets and concluded that the Accu algorithm achieved the best performance. Thus, in this experiment, we choose Accu as the truth-finding algorithm. The result of running truth-finding algorithm on the extracted candidates is a list of facts considered as the true value. However, it may contain false positives, i.e. false facts which are labeled as true by the truth-finding algorithm.

[1] https://www.tripadvisor.com/Restaurants-g297697-Kuta_Kuta_District_Bali.html.

We compare the list given by the truth-finding algorithm with our manually built baseline to measure the accuracy. In this comparison, we disregard the information on sources of the facts that are considered as true by the truth-finding algorithm. Column 3 in Table 6 shows the accuracy of the truth-finding algorithm. Next, we detect the false positives by first checking whether there are facts extracted from official Web pages that are considered as false and then compare the facts with the baseline. We count the number of facts extracted from official Web pages which match our baseline. We recalculate the accuracy and column 4 in Table 6 shows the new accuracy. There is an improvement (4.2% and 0.9% for #Group 1 and 2 respectively) in terms of accuracy if we prioritize the facts extracted from official Web pages rather than just entirely depend on the truth-finding algorithm. As we can see that the improvement is not much for #Group 2. This is understandable because there are too many variations on writing an address. Although we have made the comparison to be as fair as possible (by transforming the text to lowercase, removing spaces and special characters, etc.), but still it only improves the accuracy by less than one percent. Nevertheless, we conclude that taking into account facts that are extracted from official Web pages can indeed improve the accuracy. This will also give us more confidence on the extracted binary tuples.

Once we have the binary tuples then the tuple reconstruction process is straightforward. To get an n-length tuple we need to combine pairs of $(n-1)$ length tuples. In our experiment $n = 3$, thus we combine all pairs of the binary tuples extracted from the previous step which have the same key attribute. For example, from #Group 1 we have <*Bale Udang Mang Engking*, *+62 361 8947119*>, while from #Group 2 <*Bale Udang Mang Engking*, *Jl. Nakula no. 88*>. The two tuples can be combined and yield <*Bale Udang Mang Engking*, *+62 361 8947119*, *Jl. Nakula no. 88*>. From our experiment, we manage to construct a total of 64 tuples of length 3. Although from #Group 1 and 2 we extract 196 and 119 binary tuples respectively, not all of the key attribute (in this case the name of the restaurant) are contained in both groups. For example, in #Group 1 we have <*69 Tequila Bar*, *+62 361 752208*>, while the tuple with the same key attribute is missing from #Group 2. This is the reason why we can only construct 64 tuples of length 3.

Table 6. Prioritizing official Web pages to minimize false positives

#Group	#Distinct_restaurant	#w/o_Official_webpage	#w_Official_webpage
1	783	20% (157)	24.2% (190)
2	765	14.1% (108)	15% (119)

5 Conclusion

We present two methods for reconstructing n-element tuples from binary tuples. The binary tuples are extracted from the Web using a key attribute as the query to the search engine. To help ensure the truthfulness of the extracted binary tuples, we apply several truth-finding algorithms and measure its individual performance in terms of precision, recall, and F-measure. The best algorithm is selected as a verification tool in generating the list of truthful binary tuples. The binary tuples are then used to enrich the graph of entities with the intuition that this will help us in minimizing the number of false candidates. The empirical evaluation shows that the approach is efficient and practical. We have proved that our approach can significantly improve the confidence level of the extracted candidates by lowering the rank of the false candidates while giving higher rank to the true candidates. The entity profiling process described in this research is simple yet efficient. We are investigating ways of improving this process by adding ontological knowledge, applying natural language processing to help us in defining the relationship between the key attribute and its properties. Ensuring the truthfulness of the binary tuples also provides challenges such as the case of multiple truth for an attribute, as well as the multiple sources for attribute values. We hope to be able to tackle these problems and to further improve the level of confidence for the extracted candidates. Nevertheless, we show that our proposed approach can directly be applied to a real-world case of populating database relation automatically from the Web.

Acknowledgment. This work has been partially funded by the Big Data and Market Insights Chair of Télécom ParisTech and supported by the National University of Singapore under a grant from Singapore Ministry of Education for research project number T1 251RES1607.

References

1. Abdessalem, T., Cautis, B., Derouiche, N.: Objectrunner: lightweight, targeted extraction and querying of structured web data. PVLDB **3**(2), 1585–1588 (2010). http://www.comp.nus.edu.sg/ vldb2010/proceedings/files/papers/D18.pdf
2. Ba, M.L., Berti-Equille, L., Shah, K., Hammady, H.M.: VERA: a platform for veracity estimation over web data. In: WWW (2016)
3. Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S.E., Widom, J.: Swoosh: a generic approach to entity resolution. VLDB J. **18**(1), 255–276 (2009). http://dx.doi.org/10.1007/s00778-008-0098-x
4. Bing, L., Lam, W., Wong, T.L.: Wikipedia entity expansion and attribute extraction from the web using semi-supervised learning. In: WSDM, New York, NY, USA (2013)
5. Bleiholder, J., Draba, K., Naumann, F.: FuSem: exploring different semantics of data fusion. In: VLDB, Vienna, Austria (2007)
6. Brin, S.: Extracting patterns and relations from the World Wide Web. In: Atzeni, P., Mendelzon, A., Mecca, G. (eds.) WebDB 1998. LNCS, vol. 1590, pp. 172–183. Springer, Heidelberg (1999). https://doi.org/10.1007/10704656_11
7. Chen, Z., Cafarella, M., Jagadish, H.V.: Long-tail vocabulary dictionary extraction from the web. In: WSDM, New York, NY, USA (2016)
8. Derouiche, N., Cautis, B., Abdessalem, T.: Automatic extraction of structured web data with domain knowledge. In: 2012 IEEE 28th International Conference on Data Engineering, pp. 726–737, April 2012

9. Dong, X.L., Berti-Equille, L., Srivastava, D.: Integrating conflicting data: the role of source dependence. PVLDB **2**(1), 550–561 (2009)
10. Dong, X.L., Berti-Equille, L., Srivastava, D.: Truth discovery and copying detection in a dynamic world. PVLDB **2**(1), 562–573 (2009)
11. Dong, X.L., Naumann, F.: Data fusion: resolving data conflicts for integration. PVLDB **2**(1), 1654–1655 (2009)
12. Er, N.A.S., Abdessalem, T., Bressan, S.: Set of t-uples expansion by example. In: iiWAS, New York, NY, USA (2016)
13. Er, N.A.S., Ba, M.L., Abdessalem, T., Bressan, S.: Truthfulness of candidates in set of t-uples expansion. In: Benslimane, D., Damiani, E., Grosky, W.I., Hameurlain, A., Sheth, A., Wagner, R.R. (eds.) DEXA 2017, Part I. LNCS, vol. 10438, pp. 314–323. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64468-4_24
14. Fagin, R., Haas, L.M., Hernández, M., Miller, R.J., Popa, L., Velegrakis, Y.: Clio: schema mapping creation and data exchange. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) Conceptual Modeling: Foundations and Applications. LNCS, vol. 5600, pp. 198–236. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02463-4_12
15. Faheem, M., Senellart, P.: Adaptive web crawling through structure-based link classification. In: Allen, R.B., Hunter, J., Zeng, M.L. (eds.) ICADL 2015. LNCS, vol. 9469, pp. 39–51. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-27974-9_5
16. Fang, X.S.: Truth discovery from conflicting multi-valued objects. In: WWW, pp. 711–715 (2017)
17. Fang, X.S., Sheng, Q.Z., Wang, X., Ngu, A.H.: Value veracity estimation for multi-truth objects via a graph-based approach. In: WWW, pp. 777–778 (2017)
18. Furche, T., Gottlob, G., Grasso, G., Guo, X., Orsi, G., Schallhart, C., Wang, C.: Diadem: thousands of websites to a single database. Proc. VLDB Endow. (PVLDB) **7**, 1845–1856 (2014)
19. Galland, A., Abiteboul, S., Marian, A., Senellart, P.: Corroborating information from disagreeing views. In: WSDM, New York, USA, February 2010
20. Getoor, L., Machanavajjhala, A.: Entity resolution: theory, practice & open challenges. Proc. VLDB Endow. **5**(12), 2018–2019 (2012). https://doi.org/10.14778/2367502.2367564
21. He, Y., Xin, D.: Seisa: set expansion by iterative similarity aggregation. In: WWW, New York, NY, USA (2011)
22. Köpcke, H., Thor, A., Rahm, E.: Evaluation of entity resolution approaches on real-world match problems. Proc. VLDB Endow. **3**(1–2), 484–493 (2010). https://doi.org/10.14778/1920841.1920904
23. Li, Q., Li, Y., Gao, J., Zhao, B., Fan, W., Han, J.: Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In: SIGMOD, Snowbird, Utah, USA, May 2014
24. Liu, W., Liu, J., Duan, H., Zhang, J., Hu, W., Wei, B.: TruthDiscover: resolving object conflicts on massive linked data. In: WWW, pp. 243–246 (2017)
25. Moens, M., Li, J., Chua, T. (eds.): Mining User Generated Content. Chapman and Hall/CRC, Boca Raton (2014)
26. Ortona, S., Orsi, G., Buoncristiano, M., Furche, T.: WADaR: joint wrapper and data repair. Proc. VLDB Endow. **8**(12), 1996–1999 (2015). https://doi.org/10.14778/2824032.2824120
27. Paşca, M.: Weakly-supervised discovery of named entities using web search queries. In: CIKM, New York, NY, USA (2007)

28. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report (1999)
29. Pasternack, J., Roth, D.: Latent credibility analysis. In: WWW, Rio de Janeiro, Brazil, May 2013
30. Pochampally, R., Das Sarma, A., Dong, X.L., Meliou, A., Srivastava, D.: Fusing data with correlations. In: SIGMOD, Snowbird, Utah, USA, May 2014
31. Qiu, D., Barbosa, L., Dong, X.L., Shen, Y., Srivastava, D.: Dexter: large-scale discovery and extraction of product specifications on the web. Proc. VLDB Endow. **8**(13), 2194–2205 (2015). https://doi.org/10.14778/2831360.2831372
32. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB J. **10**(4), 334–350 (2001). https://doi.org/10.1007/s007780100057
33. Sarker, A., Gonzalez, G.: Portable automatic text classification for adverse drug reaction detection via multi-corpus training. J. Biomed. Inform. **53**, 196–207 (2015)
34. Wang, D., Kaplan, L., Le, H., Abdelzaher, T.: On truth discovery in social sensing: a maximum likelihood estimation approach. In: IPSN, Beijing, China, April 2012
35. Wang, R.C., Cohen, W.W.: Language-independent set expansion of named entities using the web. In: ICDM (2007)
36. Wang, R.C., Cohen, W.W.: Character-level analysis of semi-structured documents for set expansion. In: EMNPL, Stroudsburg, PA, USA (2009)
37. Wang, R.C., Schlaefer, N., Cohen, W.W., Nyberg, E.: Automatic set expansion for list question answering. In: EMNLP, Stroudsburg, PA, USA (2008)
38. Yin, X., Han, J., Yu, P.S.: Truth discovery with multiple conflicting information providers on the web. IEEE TKDE **20**, 796–808 (2008)
39. Zhang, W., Ahmed, A., Yang, J., Josifovski, V., Smola, A.J.: Annotating needles in the haystack without looking: product information extraction from emails. In: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2015, pp. 2257–2266. ACM, New York (2015). https://doi.org/10.1145/2783258.2788580
40. Zhang, Z., Sun, L., Han, X.: A joint model for entity set expansion and attribute extraction from web search queries. In: AAAI (2016)
41. Zhao, B., Rubinstein, B.I.P., Gemmell, J., Han, J.: A Bayesian approach to discovering truth from conflicting sources for data integration. PVLDB **5**(6), 550–561 (2012)
42. Zhao, Z., Cheng, J., Ng, W.: Truth discovery in data streams: A single-pass probabilistic approach. In: CIKM, Shangai, China, November 2014

A Cost-Sensitive Loss Function for Machine Learning

Shihong Chen[1,2], Xiaoqing Liu[3](✉), and Baiqi Li[2]

[1] Collaborative Innovation Center for 21st-Century Maritime Silk Road Studies,
Guangdong University of Foreign Studies, Guangzhou 510006, China
200911836@oamail.gdufs.edu.cn
[2] School of Information Science and Technology,
Guangdong University of Foreign Studies, Guangzhou, China
821673112@qq.com
[3] School of Internet Finance and Information Engineering,
Guangdong University of Finance, Guangzhou 510521, China
ibm255@126.com

Abstract. In training machine learning models, loss functions are commonly applied to judge the quality and capability of the models. Traditional loss functions usually neglect the cost-sensitive loss in different intervals, although sensitivity plays an important role for the models. This paper proposes a cost-sensitive loss function based on an interval error evaluation method (IEEM). Using the key points of grade-structured intervals, two methods are proposed to construct the loss function: a piecewise function linking by key points, and a curve function fitting by key points. The proposed function was evaluated against three different loss functions based on a BP neural network. The comparison results show that the proposed loss function based on IEEM made the best prediction of the $PM_{2.5}$ air quality grade in Guangzhou, China.

Keywords: IEEM · Loss function · Loss sensitivity · Machine learning

1 Introduction

The core of machine learning is a model trained by the training data, and a method to adjust the parameters of the model according to its loss function. The purpose of the training is to minimize the model's average misprediction loss [1, 2].

Different machine learning models also have specific choices for the loss function. Frequently used loss functions include the 0–1 loss function [3] (for classifiers), the square loss function [4] (for least square methods), the hinge loss function [5] (for support vector machines), the logarithmic loss function [6] (for logistic regression), the exponential loss function [7] (for AdaBoost models), and the relative loss function [8].

The classification and prediction of machine learning methods are often hampered by outliers and asymmetric costs that are associated with different types of errors. For example, for cancer judgments, incorrectly classifying a cancer patient as healthy has more serious consequences than raising a false alarm on a healthy person [9]. For disaster loss prediction, underestimating a disaster loss value incurs heavier costs than

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 255–268, 2018.
https://doi.org/10.1007/978-3-319-91455-8_22

overestimating it by the same amount. Solving such problems entails the use of cost-sensitive learning as surrogate loss functions, which attempt to minimize the expected misprediction cost, rather than use a simple measure such as the mean squared error. Also, several loss functions, such as Huber [10], LinLin [11, 12], and LinEx [13], have been proposed as alternatives.

However, all the traditional loss functions and their alternatives ignore the loss sensitivity caused by the different intervals or grades. Therefore, the cost sensitivity of different data intervals should also be considered when calculating the loss.

We have previously reported on the interval error evaluation method (IEEM) [14], which proposed that the sensitivity of the error should be processed partly in accordance with the different intervals. In this paper, the IEEM loss function is proposed: a new cost-sensitive loss function based on the IEEM. The IEEM loss function is defined as $L(y,\hat{y}) = (g(y) - g(\hat{y}))^2$, where y is the predictive value and $\hat{y}$ is the actual value. We also propose two methods to construct the function g, which can adjust loss sensitivity reasonably. The first method is a piecewise function that connects the key points, and uses interval range as a horizontal axis and grade numbers as a vertical axis. The other method is fitting a derivable curve function by the key points; this could be an exponential or a logarithmic function. We have empirically evaluated our proposed method in the domain of $PM_{2.5}$ air quality grade prediction. The results show a marked performance improvement of the proposed function over the Huber and least squares loss functions.

The organization of this paper is as follows: Sect. 2 describes related work and Sect. 3 details the IEEM loss function. Section 4 describes our experiments and results predicting the $PM_{2.5}$ air quality grade. Finally, Sect. 5 summarizes the paper.

2 Related Work

Several studies have been made on cost-sensitive learning in the data mining and statistics fields. Most of them focus on classification problems. In this section, we review the traditional loss functions and several representative cost-sensitive loss functions proposed in the past decades, especially for regression problems.

For a regression problem, it assume that the loss function $L(e)$ depends only on the prediction error $e = y - \hat{y}$.

2.1 Least Squares Loss Function

Most models for classification and regression with cost-blind loss functions assume that all errors have the same cost. The classic loss function for regression problems is the least squares loss, which refers to the square of the error, that is,

$$L(e) = (e)^2. \tag{1}$$

Although this loss is mathematically rather easy to handle, and the corresponding learning algorithms are often computationally feasible, it is well known that minimizing a model's loss based on the least squares loss is sensitive to outliers [2].

Furthermore, many loss functions are for special purposes, such as the hinge loss function, the logarithmic loss function, the exponential loss function, and the relative loss function. These are all cost-blind loss functions like the squares loss function, so this paper does not discuss them. Several cost-sensitive loss functions are introduced in the following sub-sections.

2.2 Huber Loss Function

To reduce the effect of outliers, the Huber loss function considers a convex differentiable cost function, which is quadratic for small errors and linear for others, that is [10, 15],

$$L_\delta(e) = \begin{cases} (e)^2, e \leq \delta \\ \delta \cdot (2|e| - \delta),\ \mathit{otherwise} \end{cases} \tag{2}$$

where δ is a parameter that must be set in advance. Figure 1 illustrates a squares loss function and two Huber loss functions with different δ s. Compared with the squares loss function, a Huber function can reduce the loss statistically when errors are far from the origin. So the Huber function is one kind of cost-sensitive loss function that adjusts the sensitivity by setting the value of δ. Huber loss functions for regressive learners have been proven robust, and widely applied to robust regression [16–18]. Some improved Huber loss functions have also been proposed [19–21].

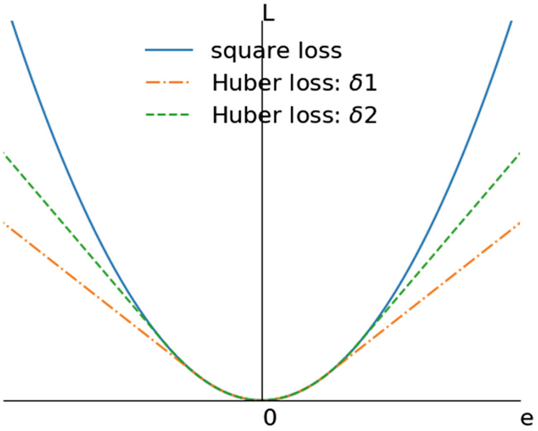


Fig. 1. A squares loss function and two Huber loss functions $(\delta_1 < \delta_2)$

2.3 Asymmetric Loss Function

Sometimes, overestimating and underestimating are associated with different types of prediction errors where $L(e)$ and $L(-e)$ are generally different: these are called asymmetric losses. Several asymmetric loss functions have been proposed in the last decades. Lin-lin, the earliest asymmetric loss function, shown in Fig. 2(a), was proposed by Granger in 1969. It has the following form [22]:

$$\mathrm{L}(e) = \begin{cases} a * e, & y - \hat{y} \geq 0 \\ b * e, & y - \hat{y} < 0 \end{cases}. \tag{3}$$

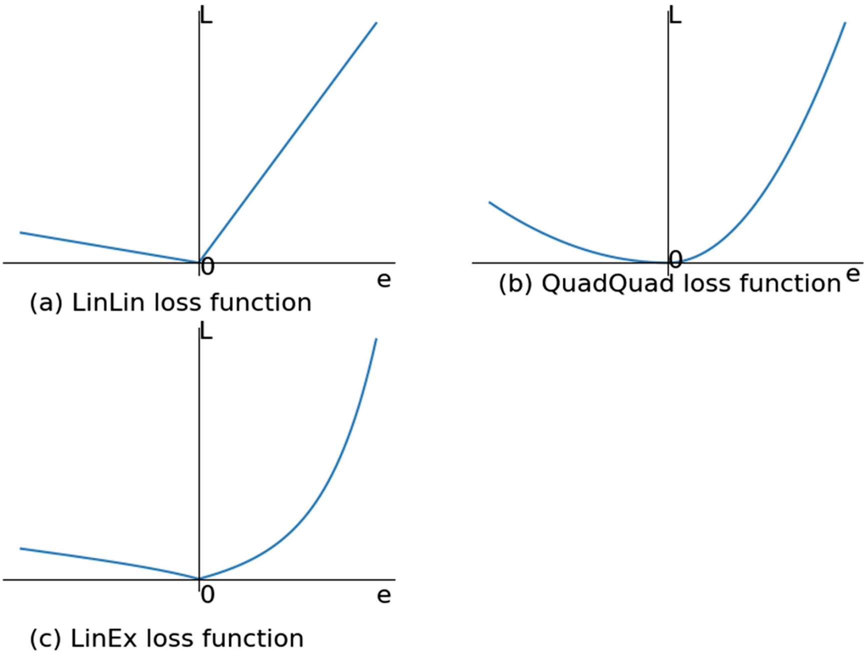
The quad-quad loss function, shown in Fig. 2(b), assumes that the asymmetric loss increasing with error is not at a linear, but at a quadratic rate [4]; that is,

$$\mathrm{L}(e) = \begin{cases} a * e^m, & y - \hat{y} \geq 0 \\ b * e^m, & y - \hat{y} < 0 \end{cases}. \tag{4}$$

In particular, Varian proposed the LinEx loss function, which is better known—even the shape of the cost function is different for the two types of errors; that is,

$$\mathrm{L}(y, \hat{y}) = b(\exp(\alpha(y - \hat{y})) - \alpha(y - \hat{y}) - 1). \tag{5}$$

Here, b > 0, $\alpha \neq 0$. Figure 2(c) shows that LinEx is approximately linear (increasing slowly) on one side and exponential (increasing quickly) on the other side. The left side could be approximately linear and the right side exponential (when $\alpha > 0$), or the opposite when $\alpha < 0$. The LinEx loss function is very useful, especially in



(a) LinLin loss function

(b) QuadQuad loss function

(c) LinEx loss function

Fig. 2. Three possible cost functions: LinLin, QuadQuad and LinEx

the field of statistics. In addition, several improved loss functions based on LinEx have been proposed [23, 24].

This series of loss function is widely used in the field of statistics. Constructing an appropriate function is often an art and requires deep domain expertise [4].

2.4 Tuning Method for Cost-Sensitive Loss

Individually making each regression learner cost-sensitive is nontrivial and often laborious [25], because parameter(s) setting is involved, professional knowledge and experience are necessary, and few optimizing or improving methods exist. To avoid these cumbersome parameter settings, the post hoc tuning method tries to find a regulatory function of the prediction of a regular regression model to make the final cost-sensitive prediction [4, 26]. The method is generally divided into two steps. First, learn a cost-blind regression model *f*, and $\mathrm{e} = y - \hat{y} = f(x) - \hat{y}$. Then, construct an adjusted regression model $f'(x) = g(f(x))$ and the loss function

$$L = g(\mathrm{y}) - \hat{y}. \tag{6}$$

The special case of a tuning function in [26] is in the form of $g(f) = f + \delta$. So the loss function is

$$L_\delta = \delta + y - \hat{y}. \tag{7}$$

A brute-force or hill-climbing algorithm can be used to find the appropriate δ. Zhaou et al. [4] improved f' as a polynomial function, which has wider applicability. This tuning method for cost-sensitive loss was actually an optimizing measure for a learner model to use a post hoc tuning algorithm. Although the cost of regression was reduced, the method could not explain how it works.

3 IEEM Loss Function

For a regression problem with a dependent variable *y* and a vector of independent variables *x*, a regression model is a mapping $f : x \to y$ learned from a training sample $\mathrm{S} = \{x_i, \hat{y}_i | i = 1, 2, 3 \ldots N\}$ by some learning method: $y_i = f(x_i)$. If the *i*th prediction error e_i incurs a loss $L(y_i, \hat{y}_i)$ and *L* is the loss function, the average misprediction loss of the regression model, as estimated on S, is defined as

$$\theta = \frac{1}{N}\sum\nolimits_{i=1}^{N} L(y_i, \hat{y}_i) \tag{8}$$

It has been pointed out that a loss function has the following three properties [12, 26]: (1) $L(0) = 0$, (2) $\min L(e) = 0$, because $L(e) \geq 0$, and (3) Assuming $e_1 = y_1 - \hat{y}$, $e_2 = y_2 - \hat{y}$, if $|e_1| \langle |e_2|$ and $e_1 \cdot e_2 \rangle 0$, then $L(e_1) < L(e_2)$.

Among these three properties of a loss function, the first one means to accurately predict no loss; the second means that there is loss in prediction error; and the third means that to the same true value, a larger prediction error will bring greater loss. Keeping to these properties, we propose the IEEM loss function as follows.

3.1 IEEM Loss Function

Sometimes we divide data into several grades according to certain rules, such as rainfall grades, disaster grades, risk grades, and wealth grades. Data at the same grade have identical rules or the same method is applied on the data. We have proposed IEEM [14], which evaluates error based on data interval divisions.

In the defined domain R of x, assuming m intervals are divided, the ith $(1 \leq i \leq m)$ interval is defined as

$$X_i = [\underline{x}_i, \bar{x}_i) = \{x \in R | \underline{x}_i \leq x < \bar{x}_i, i = 1, 2, \ldots, m\}. \tag{9}$$

Therefore, $\bar{x}_i = \underline{x}_{i+1}$ if R is continuous. We further assume that the maximum evaluation error in the ith interval is $\mathrm{k_i}$, that is, $k_i = \left| g(\bar{y}_i) - g\left(\underline{y}_i\right) \right|, i = 1, 2, \ldots, m$. $\mathrm{K_i}$ is defined as the accumulation from $\mathrm{k_1}$ to $\mathrm{k_i}$, that is,

$$K_i = \sum\nolimits_1^i k_i \tag{10}$$

Usually, $\mathrm{k_i}$ is suggested to be 1, or it can also be elastically set after professional consideration. In this paper, we assume $k_i = 1,\ i = 1, 2, \ldots, m$. Thus we define key points based on $(\bar{x}_i, \mathrm{K_i})$, i = 1, 2… m, as shown in Fig. 3. Based on the key points of intervals, a monotone function $g(y)$ is constructed as an error evaluation function. IEEM is then defined as

$$e_{ieem} = g(y) - g(\hat{y}). \tag{11}$$

The error measure converts the traditional equation $\mathrm{e} = y - \hat{y}$ to Eq. (9). Based on Eq. (11) and the squares loss function, the IEEM loss function is defined as

$$L(y, \hat{y}) = (g(y) - g(\hat{y}))^2 \tag{12}$$

How to construct the function g is the key to the IEEM loss function. We propose construction steps as follows:

Step1. Set the grades' range as an abscissa axis and K_i as a vertical axis,
Step2. Draw $(\underline{x}_1$, 0) and the key points $(\bar{x}_i, K_i)$, as shown in Fig. 3,
Step3. Construct the IEEM loss function by linking the key points. Two methods, piecewise-IEEM $\left(g_p\right)$ and curve-IEEM (g_c), are proposed to link the key points, as described in Subsects. 3.2 and 3.3.

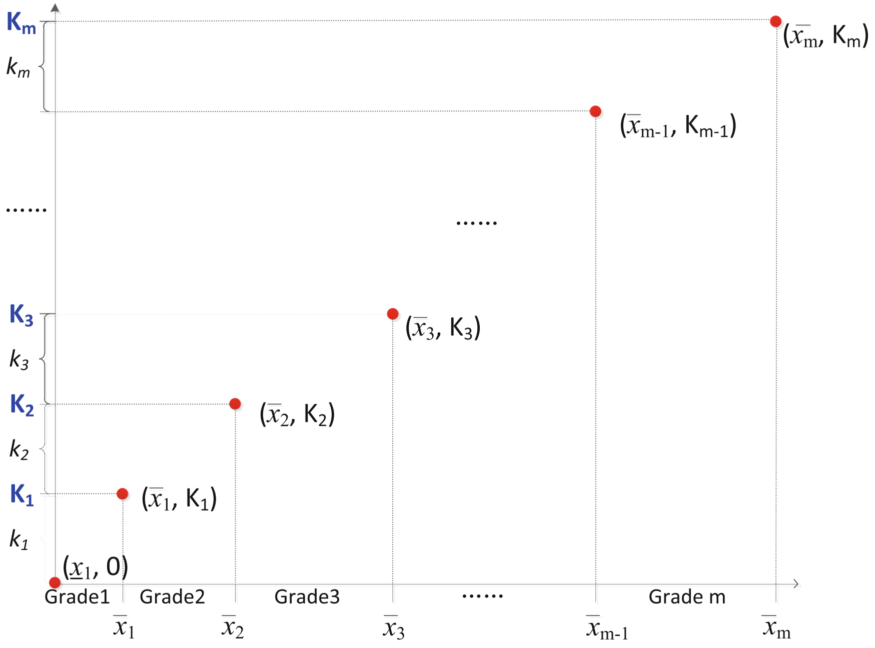


Fig. 3. Schematic diagram of IEEM key points

3.2 Piecewise-IEEM Loss Function

This method treats function g as a piecewise function divided by intervals; that is,

$$g_p(x) = \begin{cases} g_1(x), & x \in X_1 \\ g_2(x), & x \in X_2 \\ \cdots \\ g_i(x), & x \in X_i \\ \cdots \\ g_m(x), & x \in X_m \end{cases}, \tag{13}$$

where m is the number of intervals within the defined domain and $X_i = [\underline{x}_i, \bar{x}_i)$ is the ith interval of the value range. For simplicity, we set $g_p(x)$ as a piecewise linear function, and the maximum error as 1 in the same grade, that is, $k_i = 1$. Then we obtain $g_p(x)$ as

$$g_p(x) = \begin{cases} \frac{x-\underline{x}_1}{\bar{x}_1-\underline{x}_1}, & x \in X_1 \\ \frac{x-\underline{x}_2}{\bar{x}_2-\underline{x}_2} + 1, x \in X_2 \\ \cdots \\ \frac{x-\underline{x}_i}{\bar{x}_i-\underline{x}_i} + 2, x \in X_i \\ \cdots \\ \frac{x-\underline{x}_m}{\bar{x}_m-\underline{x}_m} + m - 1, x \in X_m \end{cases}. \tag{14}$$

Taking air quality grade prediction as an example, we will show how to construct the IEEM loss function. Air quality grade is based on the average daily concentration of $PM_{2.5}$, also known as particulate matter, which refers to particles in the atmosphere that are less than or equal to 2.5 μm which seriously pollute the air. The standards for the air quality grade are different in many countries. The air quality grade standard in the United States is chosen in this paper, as shown in Table 1.

Table 1. Air quality grade corresponding to the average daily concentration of $PM_{2.5}$ in the United States

Air quality		Average daily concentration of $PM_{2.5}$ $(\mu g/m^3)$
Description	Grade	
Good	1	0–12
Medium	2	12–35
Unhealthy for sensitive persons	3	35–55
Unhealthy	4	55–150
Very unhealthy	5	150–250
Toxic	6	250–500

Set the grade number as a vertical axis with the maximum error as 1 in the same grade $(k_i = 1)$ and the $PM_{2.5}$ concentration as an abscissa axis, and then draw the key points of the grade divisions. The key points are $(\bar{x}_i, i)$, i = 1, 2… 6. Link these key points with line segments, as shown in Fig. 4. g_p was acquired as

$$g_p(x) = \begin{cases} \frac{x}{12}, & x \in [0, 12) \\ \frac{x-12}{23} + 1, x \in [12, 35) \\ \frac{x-35}{20} + 2, x \in [35, 55) \\ \frac{x-55}{95} + 3, x \in [55, 150) \\ \frac{x-150}{100} + 4, x \in [150, 250) \\ \frac{x-250}{250} + 5, x \in [250, 500) \\ 6, & x \geq 500 \end{cases} . \quad (15)$$

Obviously, g_p is a monotone function in the domain. In Fig. 4, a simple example is shown to explain how to calculate the e_{ieem}. Assuming $\hat{y} = 320$, $y = 205$, then

$$\begin{aligned} e_{ieem} &= g_p(y) - g_p(\hat{y}) \\ &= 5.3 - 4.5 \\ &= 0.8 \end{aligned}$$

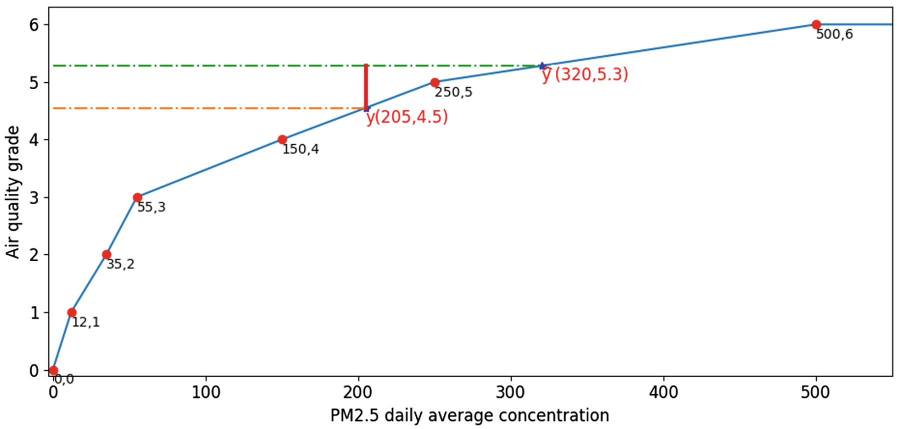


Fig. 4. IEEM error calculation diagram

Finally, the piecewise-IEEM loss function is defined as

$$L_p(y,\hat{y}) = \left(g_p(y) - g_p(\hat{y})\right)^2 \tag{16}$$

3.3 Curve-IEEM Loss Function

The piecewise-IEEM loss function is a nonsegmented function, which is difficult to deal with in mathematics. It is necessary to find a simpler function but which still has the three properties of a loss function. A smooth curvilinear function (g_c), such as a logarithmic function or an exponential function, is one candidate. This derivable curve function may be fitted by or near the key points. In addition, the curve-IEEM loss function based on g_c is defined as

$$L_c(y,\hat{y}) = (g_c(y) - g_c(\hat{y}))^2. \tag{17}$$

By observing the distribution of the key points in Fig. 4, we selected a logarithm function and performed the function fitting. Therefore,

$$g_c(x) = a * \ln(b + x) + c. \tag{18}$$

In accordance with the key points, we used the curve fitting function of the Python SciPy library to simulate Eq. (18). The parameters thus can be obtained as a = 1.627, b = 12.44, and c = − 4.131. The root mean squared error (RMSE) and the coefficient of determination (R^2) are commonly used indices for testing the fitting effect. The closer to 0 the RMSE value is, the better the fitting [27]. The range of R^2 is [0, 1], and the closer to 1, the better the fitting [28]. In this case, RMSE = 0.1827 and

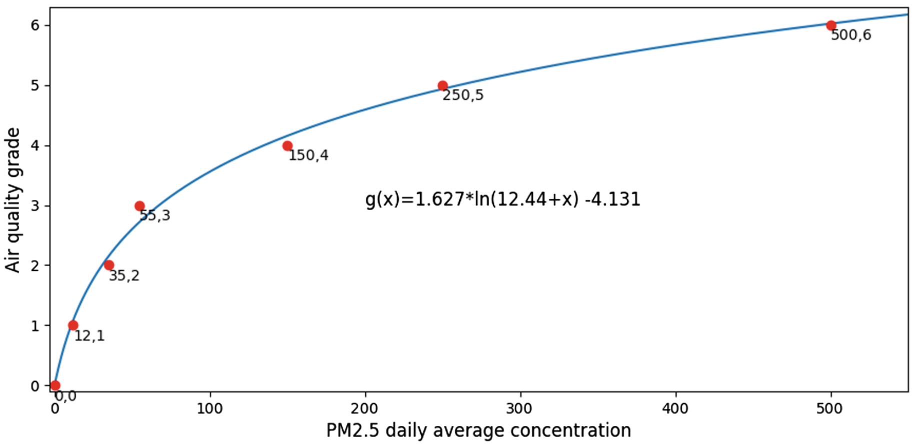


Fig. 5. Logarithmic loss function fitting effect diagram

$R^2 = 0.9952$. Figure 5 shows that the fitting effect is acceptable. Therefore, the function g_c is acquired:

$$g_c(x) = 1.627 * \ln(12.44 + x) - 4.131, x \geq 0. \quad (19)$$

By incorporating Eqs. (18) and (16), Eq. (20) is acquired (shown in Fig. 5):

$$L_c(y, \hat{y}) = 2.6471 * [\ln(12.44 + \mathrm{y}) - \ln(12.44 + \hat{y})]^2 \quad (20)$$

Through the analysis, the IEEM loss function meets the three properties of loss function listed in the introduction of this section. Compared with the tuning method; that is, Eq. (6), the IEEM loss function not only revises predictive value y but also revises the actual value $\hat{y}$ with the same rule. Thus, the position of y and $\hat{y}$ can be recognized, and the sensitivity of loss can be adjusted according to not only the distance between y and $\hat{y}$ but also the position of their intervals.

Human perception of error and the calculation of sensitive loss is a fuzzy process, so fuzzy mathematics can be used to analyze them. The function g both in piecewise-IEEM and curve-IEEM can be regarded as a fuzzy function, which can help to make the machine learning model understand loss in a condition of ambiguity.

4 Evaluation and Results

In this section, we report on the implementation and empirical evaluation of the IEEM loss function. We apply the function to a BP neural network model of $PM_{2.5}$ concentration air quality grade prediction, which is a cost-sensitive regression problem described in Sect. 3.2. For performance comparison, we applied several loss functions to the same neural network model. Because this prediction was not about symmetric loss, symmetric loss functions such as lin-lin, quad-quad, and LinEx were excluded.

Two other types of loss functions were applied to compare: a squares loss function and a Huber loss function.

4.1 The Data

We used the concentration of $PM_{2.5}$ in Guangzhou from November 2011 to June 2017 from the monitoring data of the US embassy[1]. This $PM_{2.5}$ concentration data set is recorded hourly and summed up as the average daily $PM_{2.5}$ concentration, as shown in Fig. 6.

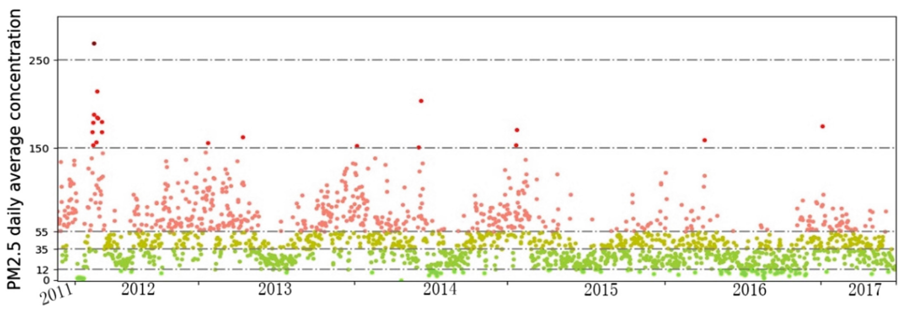


Fig. 6. Daily distribution diagram of $PM_{2.5}$ concentration in Guangzhou in recent years

Many factors affect the concentration of $PM_{2.5}$, but its short-term fluctuations are mainly related to seasonal and meteorological factors [29].

As shown in Fig. 6, the $PM_{2.5}$ concentration fluctuated seasonally. We thus sum up the average monthly $PM_{2.5}$ concentration as a seasonal fluctuation coefficient.

$PM_{2.5}$ concentration is closely related to weather conditions such as wind speed, wind direction, and humidity. After evaluating the literature [30–33] and making several tests, we chose 18 meteorological factors: surface temperature, maximum surface temperature, minimum surface temperature, average wind speed, maximum wind speed, direction of maximum wind speed, extreme wind speed, precipitation, average temperature, maximum temperature, minimum temperature, atmospheric pressure, maximum pressure, minimum pressure, sunshine hours, relative humidity, minimum relative humidity, and vaporization. All these Guangzhou meteorological data were from the meteorological data center of the China Meteorological Administration[2].

Air quality on a certain day is not only related to the air condition and meteorological factors of the previous day but also has a strong correlation with weather conditions on the forecast day [30]. Also, the accuracy rate of the meteorologic forecast was high enough to be an effective reference for predicting the air quality. So the factors used as input of the prediction model included the seasonal coefficient, meteorological factors of

[1] Web site: http://www.stateair.net/web/post/1/3.html.

[2] Web site: http://www.cma.gov.cn/2011qxfw/2011qsjgx.

the prediction day and the previous day, and the $PM_{2.5}$ concentration of the previous day, which composed a 38-dimension input structure. All these inputs were normalized in range [0, 1]. After removing incomplete records, we obtained a dataset containing 1,900 records. Ten percent of the data were randomly selected as test data and the rest as training data.

The output values need to be converted to average daily $PM_{2.5}$ concentration values, this is to say, air quality grades for the precision test.

4.2 The Results

Because the purpose of this study was to test the performance of each loss function, we used the same machine learning method to forecast the comparison effectiveness. We constructed a BP network based on a TensorFlow framework, which has four layers: the number of nodes in each layer was 38, 50, 20, and 1. The optimizer was based on the AdamOptimizer algorithm, and the learning rate was set at 0.01. The method of cross validation was used for training, and the number of steps was 5,000.

Three kinds of loss functions were constructed: the squares error loss function based on Eq. (1), the Huber loss function based on Eq. (2), and two IEEM loss functions based on Eqs. (16) and (20). In Eq. (2), for the Huber loss function, an appropriate value of parameter δ needed to be given previously; nevertheless, there was no reasonable measure to acquire it. So we used a brute-force algorithm to search it from 10 to 150 with a 0.1 skip. Finally, the best value (55.3) of parameter δ was acquired while the BP network reached the best performance.

These three loss functions have the same level of time and space overhead. The results of the three loss functions which were applied in the same data set and the same structure of BP neural network are shown in Table 2. In the comparison, the Huber loss function did better than the squares loss function, and improved the accuracy from 64.74% to 67.37%, because the former function effectively reduced anomalous variance caused by outliers and improved the robustness of the model. The IEEM loss functions, both piecewise-IEEM and curve-IEEM, obtained the two highest accuracies, which were 70.53% and 71.05%, respectively. IEEM loss functions not only controlled the cost sensitivities of losses caused by outliers but also reasonably adjusted the cost sensitivities of the losses caused by normal data. Thus, it taught the machine learning model how humans sense losses caused by error.

Table 2. Performance of a BP neural network in different loss functions

Loss function	Accuracy
Squares loss	64.74%
Huber ($\delta = 55.3$)	67.37%
Piecewise-IEEM	70.53%
Curve-IEEM	71.05%

5 Conclusions

This paper proposes a new cost-sensitive loss function for machine learning models. The proposed function is based on interval division, where grade division is as an existing and reliable method. Two methods are proposed to construct the IEEM loss function: a piecewise-IEEM loss function and a curve-IEEM loss function. Because it incorporates the three properties of loss functions and can be explained with fuzzy mathematics, the IEEM loss function is reasonable and authoritative. Furthermore, it is easy, rapid, and can be constructed with little skills. The results of comparing the proposed function with the squares loss and Huber loss functions show that the IEEM loss function is more accurate in $PM_{2.5}$ air quality grade prediction.

Acknowledgements. This work was supported by National Natural Science Foundation of China (grant No. 61772146), the Colleges Innovation Project of Guangdong (grant No. 2016KTSCX036) and Guangzhou program of Philosophy and Science Development for 13rd 5-Year Planning (grant No. 2018GZGJ40).

References

1. Zhu, M.: Comparison Research of SVR Algorithms Based on Several Loss Functions. East China Normal University (2012)
2. Steinwart, I.: How to compare different loss functions and their risks. Constr. Approx. **26**, 225–287 (2007)
3. Shalev-Shwartz, S., Shamir, O., Sridharan, K.: Learning linear and kernel predictors with the 0–1 loss function. In: International Joint Conference on Artificial Intelligence, pp. 2740–2745 (2011)
4. Zhao, H., Sinha, A.P., Bansal, G.: An extended tuning method for cost-sensitive regression and forecasting. Decis. Support Syst. **51**, 372–383 (2011)
5. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995). https://doi.org/10.1007/978-1-4757-3264-1
6. Cesa-Bianchi, N., Lugosi, G.: Worst-case bounds for the logarithmic loss of predictors. Mach. Learn. **43**, 247–264 (2001)
7. Hu, J., Luo, G., Li, Y., Cheng, W., Wei, X.: An AdaBoost algorithm for multi-class classification based on exponential loss function and its application. Acta Aeronaut. Astronaut. Sin. (2008)
8. Reich, Y., Barai, S.V.: Evaluating machine learning models for engineering problems. Artif. Intell. Eng. **13**, 257–272 (1999)
9. Cheng, T., Lan, C., Wei, C.P., Chang, H.: Cost-sensitive learning for recurrence prediction of breast cancer (2010)
10. Huber, P.J.: Robust estimation of a location parameter. Ann. Math. Stat. **35**, 73–101 (1964)
11. Ping, W., Daming, J.: Parameter design based on exponential loss function. J. Changzhou Inst. Technol. **28**, 16–20 (2015)
12. Granger, C.W.J.: Outline of forecast theory using generalized cost functions. Span. Econ. Rev. **1**, 161–173 (1999)
13. Zellner, A.: Bayesian estimation and prediction using asymmetric loss functions. Publ. Am. Stat. Assoc. **81**, 446–451 (1986)

14. Xiaoqing, L., Shihong, C., Danling, T., Yonghui, Y.: Interval Error Evaluation Method (IEEM) and it's application. Stat. Decis., 84–86 (2016)
15. Huber, P.J., Ronchetti, E.M.: Robust Statistics, 2nd edn. Wiley, New York (2011)
16. Karasuyama, M., Takeuchi, I.: Nonlinear regularization path for the modified Huber loss Support Vector Machines. In: International Joint Conference on Neural Networks, pp. 1–8 (2010)
17. Yamamoto, T., Yamagishi, M., Yamada, I.: Adaptive proximal forward-backward splitting applied to Huber loss function for sparse system identification under impulsive noise. IEICE Tech. Rep. Sig. Process. **111**, 19–23 (2012)
18. Chen, C., Yan, C., Zhao, N., Guo, B., Liu, G.: A robust algorithm of support vector regression with a trimmed Huber loss function in the primal. Soft. Comput. **21**, 1–9 (2016)
19. Chen, C., Li, Y., Yan, C., Dai, H., Liu, G.: A robust algorithm of multiquadric method based on an improved Huber loss function for interpolating remote-sensing-derived elevation data sets. Remote Sens. **7**, 3347–3371 (2015)
20. Cavazza, J., Murino, V.: Active-labelling by adaptive Huber loss regression (2016)
21. Peker, E., Wiesel, A.: Fitting generalized multivariate Huber loss functions. IEEE Signal Process. Lett. **23**, 1647–1651 (2016)
22. Granger, C.W.J.: Prediction with a generalized cost of error function. J. Oper. Res. Soc. **20**, 199–207 (1969)
23. Coetsee, J., Bekker, A., Millard, S.: Preliminary test and Bayes estimation of a location parameter under BLINEX loss. Commun. Stat. **43**, 3641–3660 (2014)
24. Arashi, M., Tabatabaey, S.M.M.: Estimation in multiple regression model with elliptically contoured errors under MLINEX loss. J. Appl. Probab. Stat. **3**, 23–35 (2008)
25. Domingos, P.: MetaCost: a general method for making classifiers cost-sensitive. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 155–164 (1999)
26. Bansal, G., Sinha, A.P., Zhao, H.: Tuning data mining methods for cost-sensitive regression: a study in loan charge-off forecasting. J. Manag. Inf. Syst. **25**, 315–336 (2008)
27. Chai, T., Draxler, R.R.: Root mean square error (RMSE) or mean absolute error (MAE)? Geoscientific Model Dev. **7**, 1247–1250 (2014)
28. Chesher, A.: A note on a general definition of the coefficient of determination. Biometrika **78**, 691–692 (1991)
29. Long, L., Lei, M., Jianfeng, H., Dangguo, S., Sanli, Y., Yan, X., Lifang, L.: PM2.5 concentration prediction model of least squares support vector machine based on feature vector. J. Comput. Appl. **34**, 2212–2216 (2014)
30. Ting, D., Jianhui, Z., Yong, H.: AQI levels prediction based on deep neural network with spatial and temporal optimizations. Comput. Eng. Appl. **53**, 17–23 (2017)
31. Song, L.I., Wang, J., Zhang, D.C., Xia, W.: Simulation analysis of prediction optimization model for atmospheric PM2.5 pollution index. Comput. Simul. (2015)
32. Chen, Y., Wang, L., Zhang, L.: Research on application of BP artificial neural network in prediction of the concentration of PM2.5 in Beijing. J. Comput. Appl. **30**, 153–155 (2016)
33. Zhou, S., Li, W., Qiao, J.: Prediction of PM2.5 concentration based on recurrent fuzzy neural network. In: Control Conference, pp. 3920–3924 (2017)

Top-N Trustee Recommendation with Binary User Trust Feedback

Ke Xu, Yi Cai(✉), Huaqing Min, and Jieyu Chen

South China University of Technology, Guangzhou 510006, China
{kexu,ycai,hqmin}@scut.edu.cn, ouxuaner@icloud.com

Abstract. Trust is one of the most important types of social information since we are more likely to accept viewpoints from whom we trust. Trustee recommendation aims to provide a target individual with a list of candidate users she might be trust. However, most existing work on this topic focuses on the use of trusters' interest but ignores the influence of trustees for recommendation. In this article, we propose a simple but effective method with the incorporation of both interest and influence of users for trustee recommendation based on binary user-user trust feedback. Specifically, we first introduce LDA twice on truster-documents corpus and trustee-documents corpus respectively to discover interest communities of users and influence communities of users. We then perform matrix factorization method on each community and finally design a merge method to rank the top-N trustees for a target user. Experimental results on Epinions dataset demonstrate that our proposed method outperforms other counterparts by large margins.

Keywords: Trustee recommendation · Topic modeling
Communities · Matrix factorization

1 Introduction

Social recommender systems have attracted much more attention during the past few years due to the prevalence of online social networking services. In a social recommender with trust implementation, like Epinions[1], where users can specify whom to trust and build her social trust-network. This process of trust generation is uni-directed, i.e., if user u add user v to her trust list while user v is not necessarily to confirm the action. User u thus becomes one of user v's trusters and user v becomes one of user u's trustees.

Confronting a vast volume of data resources, users require a method for fast finding their desired data [11,12]. Trustee recommendation, also known as a type of Top-N user recommendation became an important research topic, since people are more willing to receive suggestions from users they trust. Most work on this topic are designed for truster's interest extraction. However, they neglect

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

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, pp. 269–279, 2018.
https://doi.org/10.1007/978-3-319-91455-8_23

the fact that people often influence each other by recommending items/users. That is, the influence of trustees should also be considered in order to achieve better recommendation performance.

Armed with this concept, we utilize the binary user trust feedback (which in this case indicate truster-trustee relationships) and propose a two-step approach to recommend trustees to a target user in this work. We first employ LDA method twice separately on truster-documents corpus and on trustee-documents corpus to discover interest communities and influence communities of users. Then we apply matrix factorization on every discovered communities. Based on the results obtained after matrix factorization, we organize two candidate lists according to interest communities and influence communities respectively. Finally we devise a method to merge these two candidates lists for final trustee recommendation. Extensive experiments on a real-word dataset Epinions demonstrate that the proposed method outperforms counterparts by large margins.

The remainder of the paper is organized as follows. Related studies are reviewed in Sect. 2. Section 3 introduces the proposed method, and in Sect. 4, we validate the effectiveness of the proposed method by experimental evaluation on a real-word dataset. Finally, we conclude this paper in Sect. 5.

2 Related Work

CF approach utilizes the wisdom of crowds and has achieved great success in recommending area [9,15]. Matrix Factorization (MF) is one of the most successful CF method and has also shown to be very valuable in scenarios with implicit feedback [3,4,7,8]. *IF-MF* [4] is the state-of-the-art MF extensions for implicit feedback, which predicts if an item is selected or not coupled with a confidence level. In another direction, various LDA [1] models have been proposed. Reference [2] designs a LDA-based model to group users to handle popular users. Work in [5] presents a topic model to discover user-oriented and community-oriented topics simultaneously for recommending users. LDA is used in [10] to mine interests of users based on ratings and tags. Reference [6] uses topic model to analysis users' repost behaviors. Work [13,14] propose a UIS-LDA model, which is able to incorporate users' interest and social connection to predict user preferences for better user recommendation. However, all of these work focuses on the use of truster's interest but ignores the influence of trustees for recommendation.

CB-MF is the most similar work to our proposed method *DuLDA-MF*. It utilizes LDA for clustering users into communities to enhance the existing MF-based user recommendation. To the best of our knowledge, it is the first work to consider both interest and influence of users for user recommendation. However, it roughly maps both followers interest and followees influence into the same latent space. That is, it failures to distinguish the two factors and also difficult to be explained.

3 The Proposed Method

The scope of our recommendation method is to rank candidate trustees for a target user where only user-user social trust information is provided. Technically, we first introduce a dual LDA process to discover interest communities of users and influence communities of users (See Sect. 3.1), and then apply MF on each community for Top-N trustee recommendation (See Sect. 3.2).

To facilitate the following discussion, we introduce a number of notations. Let U represents the set of users, E represents the set of user pairs, each $e(f, g) \in E$ indicates a *truster-trustee* relation from truster f to trustee g. Trusters set F, trustees set G are formulated as follows:

$$F = \{f | f \in U \wedge \exists (g \in U \wedge e(f, g) \in E)\} \tag{1}$$

$$G = \{g | g \in U \wedge \exists (f \in U \wedge e(f, g) \in E)\} \tag{2}$$

Hence, the task of our Top-N trustee recommendation can be formalized as follows: giving a set of social trust relation $e(f, g)$, for each user u, recommend her a small list (N) of ordered trustees from that she has not yet added to her trust list.

3.1 Discover Communities of Users

LDA is one of the most advanced algorithms for topics modeling. In this work, we introduce LDA twice on truster-trustee relationships for topics extraction, namely *DuLDA* for convenient. Specifically, *DuLDA* includes a truster-documents LDA process and a trustee-documents LDA process. The former is for extracting interest topics of users, and the latter is for extracting influence topics of users.

Discover Interest Communities. Just as one has a topic in mind when choosing a word for a document, likewise a user has an interest in mind when select another user as trustee. Therefore, we regard each trustee $g \in G$ as a word, every truster $f \in F$ as a truster-document d_f containing all her trustees. The truster-document d_f and truster-documents corpus D_f are formulated as follows:

$$d_f = \{g | g \in G \wedge \exists e(f, g) \in E\} \tag{3}$$

$$D_f = \bigcup_{f \in F} d_f \tag{4}$$

The plate notation for this truster-documents LDA is showed in Fig. 1 where z^{in}, θ^{in} and ϕ^{in} are random variables, and g is the observed variable. α^{in}, β^{in} are given hyper parameters. We denote that $|D_f|$ is the number of truster-documents corpus, $|F|$ is the number of trusters and each d_f has N_{d_f} trustees. θ^{in} with *Dirichlet* prior α^{in} depicts the distribution of per-truster-document on K^{in} interest topics, ϕ^{in} with *Dirichlet* prior β^{in} captures the proportion of per-trustee is assigned from interest topics Z^{in}.

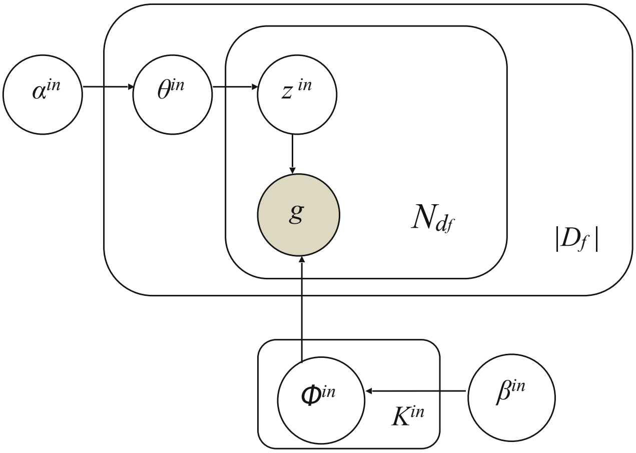


Fig. 1. The plate notation for *truster-documents LDA*.

We denote the variables z_i corresponding to the i-*th* trustee in a truster-document d_f, z_i is the interest topic allocation for this trustee. $\mathbf{z}^{\neg i}$ represents all interest topics allocation except for z_i. During the sampling process, we sample variable z_i for each iteration as:

$$Pr(z_i = z^{in}|\mathbf{z}^{\neg i}, D_f, \alpha^{in}, \beta^{in}, g) \propto \frac{n^{\neg i}_{d_f,z^{in}} + \alpha^{in}}{\sum\limits_{z' \in Z^{in}} n^{\neg i}_{d_f,z'} + K^{in} \times \alpha^{in}} \times \frac{n^{\neg i}_{z^{in},g} + \beta^{in}}{\sum\limits_{g' \in G} n^{\neg i}_{z^{in},g'} + |G| \times \beta^{in}} \quad (5)$$

In the above formula, $n^{\neg i}_{z^{in},g}$ denotes the number of times that an observed trustee g under topic z^{in} excluding z_i; $n^{\neg i}_{d_f,z^{in}}$ refers to the count of a document d_f was assigned to topic z^{in} except for z_i. After sampling is complete, we infer the latent variable $\theta^{in}_{d_f}$ via the following equation:

$$\theta^{in}_{d_f} = \frac{n_{z^{in},g} + \alpha^{in}}{\sum\limits_{z' \in Z^{in}} n_{d_f,z'} + K^{in} \times \alpha^{in}} \quad (6)$$

For each interest topic z^{in}, we then form a corresponding interest community c^{in}. It includes trusters in $c^{in}.F$ and trustees in $c^{in}.G$, which are given by follows:

$$c^{in}.F = \{f | f \in F \wedge \exists (Pr(z^{in}|d_f) \geq \gamma)\} \quad (7)$$

$$c^{in}.G = \{g | g \in G \wedge \exists (Pr(z^{in}|d_g) \geq \zeta)\} \quad (8)$$

where both γ, ζ are thresholds.

Since a higher $Pr(z^{in}|d_f)$ or $Pr(z^{in}|d_g)$ indicates the user is more strongly associated with the topic, for the corresponding community, we regard $Pr(z^{in}|d_f)$ as the trusters membership and $Pr(z^{in}|d_g)$ as the trustees membership. Among them, $Pr(z^{in}|d_f)$ is defined as:

$$Pr(z^{in}|d_f) = \frac{Pr(z^{in}|d_f)}{\sum\limits_{z' \in Z^{in}} Pr(z'|d_f)} \tag{9}$$

The numerator $Pr(z^{in}|d_f)$ can be obtained from $\theta^{in}_{d_f}$, and $Pr(z^{in}|d_g)$ can be achieved with the following equation:

$$Pr(z^{in}|d_g) = \frac{\sum\limits_{f \in d_g} Pr(z^{in}|d_f)}{\sum\limits_{z' \in Z^{in}} \sum\limits_{f \in d_g} Pr(z'|d_f)} \tag{10}$$

The edge in an interest community c^{in} denoted as $c^{in}.E$ is given by:

$$c^{in}.E = \{e(f,g)|e(f,g) \in E \wedge f \in c^{in}.F \wedge g \in c^{in}.G\} \tag{11}$$

Discover Influence Communities. A user often has various influence to attract another user to follow her. Therefore, we regard each truster $f \in F$ as a word, every trustee $g \in G$ as a trustee-document d_g containing all her trusters. The trustee-document d_g and trustee-documents corpus D_g are formulated as follows:

$$d_g = \{f|f \in F \wedge \exists e(f,g) \in E\} \tag{12}$$

$$D_g = \bigcup_{g \in G} d_g \tag{13}$$

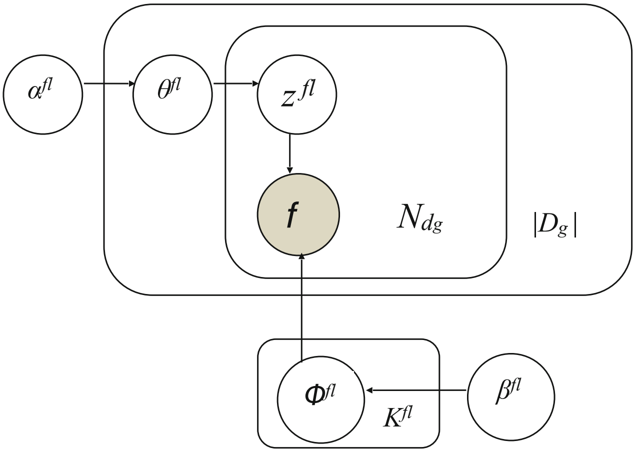


Fig. 2. The plate notation for *trustee-documents LDA*.

This trustee-documents LDA plate notation is showed in Fig. 2 where z^{fl}, θ^{fl} and ϕ^{fl} are random variables, and f is the observed variable. α^{fl}, β^{fl} are given hyper parameters. We denote that $|D_g|$ is the number of trustee-documents corpus, $|G|$ is the number of trustees, and each d_g has N_{d_g} trusters. θ^{fl} with *Dirichlet* prior α^{fl} depicts the distribution of per-trustee-document on K^{fl} influence topics; ϕ^{fl} with *Dirichlet* prior β^{fl} captures the proportion of per-truster is assigned from influence topics Z^{fl}.

We denote the variables z_i corresponding to the i-*th* truster in a trustee-document d_g, z_i is the influence topic allocation for this truster. During our sampling process, we sample variable z_i for each iteration as:

$$Pr(z_i = z^{fl}|\mathbf{z}^{\neg i}, D_g, \alpha^{fl}, \beta^{fl}, f) \propto \frac{n^{\neg i}_{d_g, z^{fl}} + \alpha^{fl}}{\sum\limits_{z' \in Z^{fl}} n^{\neg i}_{d_g, z'} + K^{fl} \times \alpha^{fl}} \times \frac{n^{\neg i}_{z^{fl}, f} + \beta^{fl}}{\sum\limits_{f' \in F} n^{\neg i}_{z^{fl}, f'} + |F| \times \beta^{fl}} \tag{14}$$

In the above formula, $n^{\neg i}_{z^{fl}, f}$ denotes the number of times that an observed truster f under topic z^{fl} excluding z_i; $n^{\neg i}_{d_g, z^{fl}}$ refers to the count of a document d_g was assigned to topic z^{fl} except for z_i. After sampling is complete, we infer the latent variable $\theta^{fl}_{d_g}$ via the following equation:

$$\theta^{fl}_{d_g} = \frac{n_{z^{fl}, f} + \alpha^{fl}}{\sum\limits_{z' \in Z^{fl}} n_{d_g, z'} + K^{fl} \times \alpha^{fl}} \tag{15}$$

For each influence topic z^{fl}, we then form a corresponding influence community c^{fl}. It includes trusters in $c^{fl}.F$ and trustees in $c^{fl}.G$, which are given by follows:

$$c^{fl}.F = \{f | f \in F \wedge \exists(Pr(z^{fl}|d_f) \geq \gamma)\} \tag{16}$$

$$c^{fl}.G = \{g | g \in G \wedge \exists(Pr(z^{fl}|d_g) \geq \zeta)\} \tag{17}$$

where both γ, ζ are thresholds.

Similar with interest communities formation, we regard $Pr(z^{fl}|d_f)$ as the trusters membership and $Pr(z^{fl}|d_g)$ as the trustees membership. Among them, $Pr(z^{fl}|d_g)$ is defined as:

$$Pr(z^{fl}|d_g) = \frac{Pr(z^{fl}|d_g)}{\sum\limits_{z' \in Z^{fl}} Pr(z'|d_g)} \tag{18}$$

The numerator $Pr(z^{fl}|d_g))$ can be obtained from $\theta^{fl}_{d_g}$, and $Pr(z^{fl}|d_f)$ can be achieved with the following equation:

$$Pr(z^{fl}|d_f) = \frac{\sum\limits_{g \in d_f} Pr(z^{fl}|d_g)}{\sum\limits_{z' \in Z^{fl}} \sum\limits_{g \in d_f} Pr(z'|d_g)} \tag{19}$$

The edge in an influence community c^{fl} denoted as $c^{fl}.E$ is given by:

$$c^{fl}.E = \{e(f, g) | e(f, g) \in E \wedge f \in c^{fl}.F \wedge g \in c^{fl}.G\} \tag{20}$$

3.2 User Recommendation

After independently training truster-documents LDA model and trustee-documents LDA model, we can obtain two sets of communities: K^{in} numbers of interest communities and K^{fl} numbers of influence communities. We perform IF-MF algorithm on each community to map the trusters and trustees into the reduced latent space of L respectively.

We organize every community c (c here refers an interest community and an influence community otherwise) as a matrix form $\tilde{M}_c$. Through performing IF-MF on each $\tilde{M}_c$, we obtain C_score (f,g,c) for every community c. Noted that x_f are latent feature vectors for trusters and y_g are latent feature vectors for trustees in community c.

$$C\_score(f, g, c) = x_f \cdot y_g \tag{21}$$

Thereafter, we separately take the maximum score of $C^{in}\_score(f, g, c^{in})$ among K^{in} communities and $C^{fl}\_score(f, g, c^{fl})$ among K^{fl} communities. They are denoted by $F^{in}\_score(f, g)$ and $F^{fl}\_score(f, g)$, respectively.

$$F^{in}\_score(f, g) = \underset{c \in C^{in}}{Maximum}(C\_score(f, g, c)) \tag{22}$$

$$F^{fl}\_score(f, g) = \underset{c \in C^{fl}}{Maximum}(C\_score(f, g, c)) \tag{23}$$

Following that, we generate two candidate lists for each truster f: a list of top-N candidates and a list of top-$(N+\delta)$ candidates. The former ranks N users with highest $F^{in}\_score(f, g)$ scores, denoted by list A^f; the latter list ranks $(N + \delta)$ users with highest $F^{fl}\_score(f, g)$ scores, denoted by list B^f. For each candidates g in the ordered set A^f, we check every element in set B^f to see if the same g exists. If it is, we will compare the $F^{in}\_score(f, g)$ score with $F^{fl}\_score(f, g)$, and choose the higher score to replace the original $F^{in}\_score(f, g)$ score in A^f. Until all the candidates in A^f are checked, we rerank A^f according to the updated scores and take it as the final top-N list for the target user f.

4 Experiments

4.1 Description of the Dataset

To validate the proposed method we conducted extensive experiments on Epinions dataset, which is taken from a public web site[2]. We deleted the users with less than five trusters/trustees. The preprocessed dataset is also extremely sparse and imbalanced containing 44852 users with 13008 trusters, 44711 trustees and the number of explicit trust relations between users is 442,175. Its density is 0.03% in terms of trust relations. For each truster, we randomly choose 90% trustees she has trusted as training set data and the remaining 10% trustees are used as testing set data. The evaluation metrics used in our experiments are Recall, Precision, F1 Score and NDCG.

[2] http://www.trustlet.org/wiki/Downloaded_Epinions_dataset.

4.2 Comparative Methods

To comparatively evaluate the performance of our proposed method *DuLDA-MF*, we take the following six related methods as competitors:

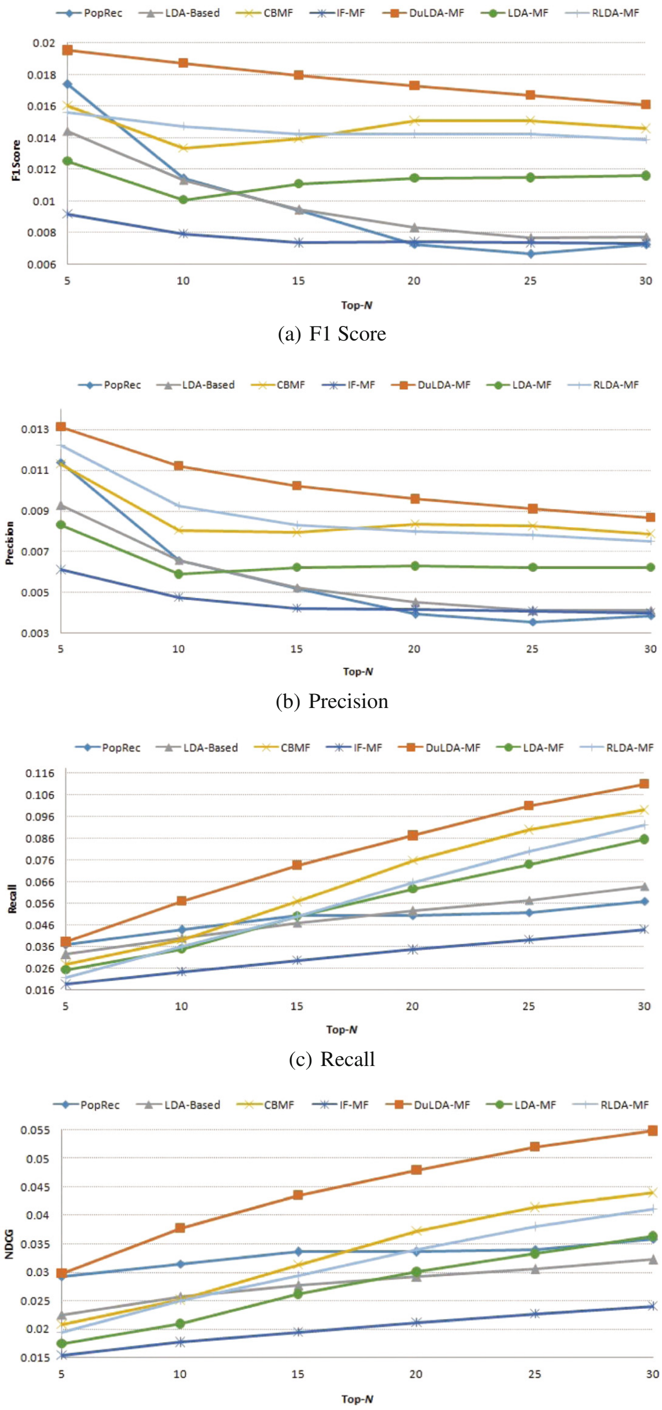
1. ***CB-MF*** [16]. A community-based user recommendation method.
2. ***IF-MF*** [4]. A state-of-the-art MF technique for implicit feedback data.
3. ***LDA-MF***. Unlike *DuLDA-MF*, only the truster-documents LDA process is conducted.
4. ***RLDA-MF***. Unlike *DuLDA-MF*, only the trustee-documents LDA process is conducted.
5. ***LDA-Based***. An LDA-based model proposed in [2] and we recommend trustees using the equation as Ref. [13,14]:
6. ***PopRec***. This method generates a non-personalized ranked trustee list based on how often the users are chosen as trustees among all users.

For LDA-based model, we set Dirichlet prior hyper-parameters as $\alpha^{in} = \alpha^{fl} = \beta^{in} = \beta^{fl} = 0.1$. We also set the number of latent topics $K^{in} = 5$ and $K^{fl} = 5$ for our *DuLDA-MF*, $K^{fl} = 10$ for $RLDA - MF$ and $K^{in} = 10$ for $LDA - MF$ and $LDA - Based$. We also empirically set thresholds $\gamma = 0.4$, $\zeta = 0.01$. For all the MF models, we set the number of latent factors $L = 10$. We experimentally set $\delta = 10$ in this paper.

4.3 Method Comparisons

Figure 3 presents the recommendation performance of all the comparison methods in terms of F1 Score@N, Precision@N, Recall@N and NDCG@N, respectively. Generally, our method *DuLDA-MF* obtains the best performance in comparison with all the other methods. Compared to the best performance of baseline methods *CB-MF*, *DuLDA-MF* averagely increases the F1 Score by 21.10%, the Precision by 19.98%, the Recall by 25.07%, and the NDCG by 19.98%. We attribute these results to the advantage of separately considering users' interest and influence instead of mapping them into the same latent space. This can help extracting higher quality of topics and thus significantly improving the effectiveness of trustee recommendation.

CB-MF outperforms other MF based methods (*LDA-MF*, *RLDA-MF*). These results again prove that integrating both interest of users and influence of users to learn user trust preferences can improve the recommendation performance. An interesting and important finding is that *RLDA-MF* outperforms *LDA-MF*. Previous research concentrated on truster's interest, as this paper mentioned about truster-documents LDA processing. However, our experiment discovered that recommend from trustee's influence (from trustee-documents LDA) yield even better results. Thus, we believe that incorporating users' influence positively boosting our results. On the other hand, the most basic, non-personalized *PopRec* method can achieve tolerable results in some cases. It may imply that users tend to trust popular trustees to some extent. We also find



(a) F1 Score

(b) Precision

(c) Recall

(d) NDCG

Fig. 3. Comparison of trustee recommendation on Epinions dataset.

that directly performing *IF-MF* on original data set outputs the worst results on various evaluation metrics, the reason we consider is the extremely sparsity of user-user trust relationships. It also confirms that the necessity of discovering communities before matrix factorization which helps to mitigate the data sparse problem.

5 Conclusion

This article proposed a simple but effective trustee recommendation method with the incorporation of truster's interest and trustee's influence. Technically, we organized truster-documents corpus and trustee-documents corpus for LDA processing. Based on extracted interest topics and influence topics of users, we picked qualified users to form interest communities and influence communities accordingly. After that, we performed matrix factorization on each community and merged the result to generate N ranked trustees toward a target user. We conducted experiments on a real-word data set, and demonstrated that our method performed the best in comparison with other counterparts. In the future work we plan to extend our approach by integrating user-user social trust information with user-item feedback history as to improve recommendation accuracy.

Acknowledgement. We would like to thank the anonymous reviewers for their comments and suggestions. This work is supported by the Fundamental Research Funds for the Central Universities, SCUT (No. 2017ZD0482015ZM136), Tiptop Scientific and Technical Innovative Youth Talents of Guangdong special support program (No. 2015TQ01X633), Science and Technology Planning Project of Guangdong Province, China (No. 2016A030310423), Science and Technology Program of Guangzhou (International Science and Technology Cooperation Program No. 201704030076 and Science and Technology Planning Major Project of Guangdong Province (No. 2015A070711001).

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
2. Cha, Y., Cho, J.: Social-network analysis using topic models. In: ACM Conference on Research and Development in Information Retrieval, SIGIR, pp. 565–574 (2012). https://doi.org/10.1145/2348283.2348360
3. He, X., Zhang, H., Kan, M.Y., Chua, T.S.: Fast matrix factorization for online recommendation with implicit feedback. In: ACM Conference on Research and Development in Information Retrieval, SIGIR, pp. 549–558 (2016). https://doi.org/10.1145/2911451.2911489
4. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: IEEE International Conference on Data Mining, ICDM, pp. 263–272 (2008). https://doi.org/10.1109/ICDM.2008.22
5. Li, L., Peng, W., Kataria, S., Sun, T., Li, T.: Frec: a novel framework of recommending users and communities in social media. In: ACM Conference on Information & Knowledge Management, CIKM, pp. 1765–1770 (2013). https://doi.org/10.1145/2505515.2505645

6. Lu, X., Li, P., Ma, H., Wang, S., Xu, A., Wang, B.: Computing and applying topic-level user interaction in microblog recommendation. In: ACM Conference on Research and Development in Information Retrieval, SIGIR, pp. 843–846 (2014). https://doi.org/10.1145/2600428.2609455
7. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: bayesian personalized ranking from implicit feedback. In: Conference on Uncertainty in Artificial Intelligence, UAI, pp. 452–461 (2009)
8. Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M., Oliver, N., Hanjalic, A.: CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering. In: ACM Recommender Systems, RecSys, pp. 139–146 (2012). https://doi.org/10.1145/2365952.2365981
9. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv. Artif. Intell. **2009**(12), 1–19 (2009). https://doi.org/10.1155/2009/421425
10. Wang, S., Gong, M., Li, H., Yang, J., Wu, Y.: Memetic algorithm based location and topic aware recommender system. Knowl.-Based Syst. **131**, 125–134 (2017). https://doi.org/10.1016/j.knosys.2017.05.030
11. Xie, H., Li, X., Wang, T., Chen, L., Li, K., Wang, F., Cai, Y., Li, Q., Min, H.: Personalized search for social media via dominating verbal context. Neurocomputing **172**, 27–37 (2016). https://doi.org/10.1016/j.neucom.2014.12.109
12. Xie, H., Li, X., Wang, T., Lau, R.Y., Wong, T.L., Chen, L., Wong, F.L., Qing, L.: Incorporating sentiment into tag-based user profiles and resource profiles for personalized search in folksonomy. Inf. Process. Manage. **52**, 61–72 (2016). https://doi.org/10.1016/j.ipm.2015.03.001
13. Xu, K., Cai, Y., Min, H., Zheng, X., Xie, H., Wong, T.L.: UIS-LDA: a user recommendation based on social connections and interests of users in uni-directional social networks. In: ACM Conference on Web Intelligence, WI, pp. 260–265 (2017). https://doi.org/10.1145/3106426.3106494
14. Xu, K., Zheng, X., Cai, Y., Min, H., Gao, Z., Zhua, B., Xie, H., Wong, T.L.: Improving user recommendation by extracting social topics and interest topics of users in uni-directional social networks. Knowl.-Based Syst. **140**, 120–133 (2018). https://doi.org/10.1016/j.knosys.2017.10.031
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: ACM Conference on Research and Development in Information Retrieval, SIGIR, pp. 83–92 (2014). https://doi.org/10.1145/2600428.2609579
16. Zhao, G., Lee, M.L., Hsu, W., Chen, W., Hu, H.: Community-based user recommendation in uni-directional social networks. In: ACM Conference on Information & Knowledge Management, CIKM, pp. 189–198 (2013). https://doi.org/10.1145/2505515.2505533

Correction to: Filtering Techniques for Regular Expression Matching in Strings

Tao Qiu, Xiaochun Yang, and Bin Wang

Correction to: Chapter "Filtering Techniques for Regular Expression Matching in Strings" in: C. Liu et al. (Eds.): *Database Systems for Advanced Applications*, LNCS 10829, https://doi.org/10.1007/978-3-319-91455-8_12

The original version of the chapter starting on p. 118 was revised.

The title of the chapter has been supplemented with "Invited Talk".

The acknowledgement "This paper constituted an invited talk, held at BDQM 2018, a DASFAA 2018 satellite workshop. The main techniques derive from our work cited in [11]" has been added.

The original chapter was corrected.

The updated online version of this chapter can be found at
https://doi.org/10.1007/978-3-319-91455-8_12

C. Liu et al. (Eds.): DASFAA 2018, LNCS 10829, p. E1, 2018.
https://doi.org/10.1007/978-3-319-91455-8_24

Author Index