Information Sciences 466 (2018) 258–269

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Identifying advisor-advisee relationships from co-author networks via a novel deep model

Zhongying Zhao[a,b,*], Wenqiang Liu[a], Yuhua Qian[c], Liqiang Nie[d], Yilong Yin[e,*], Yong Zhang[b]

[a] College of Computer Science and Engineering, Shandong Province Key Laboratory of Wisdom Mine Information Technology, Shandong University of Science and Technology, Qingdao, China
[b] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
[c] Institute of Big Data Science and Industry, Shanxi University, Taiyuan, Shanxi, China
[d] School of Computer Science and Technology, Shandong University, Qingdao, China
[e] Shandong Province Key Laboratory of Software Engineering, Shandong University, Jinan, China

ARTICLE INFO

Article history:
Received 24 September 2017
Revised 4 April 2018
Accepted 25 July 2018
Available online 27 July 2018

Keywords:
Social network analysis
Relationship mining
Co-author network
Advisor-advisee prediction

ABSTRACT

Advisor-advisee is one of the most important relationships in research publication networks. Identifying it can benefit many interesting applications, such as double-blind peer review, academic circle mining, and scientific community analysis. However, the advisor-advisee relationships are often hidden in research publication network and vary over time, thus are difficult to detect. In this paper, we present a time-aware Advisor-advisee Relationship Mining Model (tARMM) to better identify such relationships. It is a deep model equipped with improved Refresh Gate Recurrent Units (RGRU). Extensive experiments over real-world DBLP data have well verified the effectiveness of our proposed model.

1. Introduction

Social media (e.g., Twitter and Wechat) are gaining momentum in these years and become indispensable to most of us. Various social relationships are built in such social media, such as friendships on Facebook and trust relations on Epinions. It is well recognized that different types of social relationships are essentially affecting different aspects of people's life. For example, a Ph.D. candidate's research areas are largely influenced by his/her advisor [6], while his/her hobbies or behaviors are often affected by his/her families or friends to a high extent [7,18].

A research publication network comes into being in the process of research. It contains rich information of authors, paper titles, publication year, publication venue and etc, which imply abundant knowledge about advisor-advisee relationships. Identifying such relationships can benefit many significant applications [14]. With such relationships at hand, we can easily discover how researchers form different communities [5] [22] [25], how research topics emerge and evolve over time [9] [19] [24], and how a researcher influences the academic research community [8]. It is thus important and interesting to identify the advisor-advisee relationships from research publication networks.

To clearly describe the problem, Fig. 1 illustrates an example of the advisor-advisee relationship analysis over a research publication network. In the left part, a research publication network contains the information of authors, papers, and author-paper relationships. The middle part shows the preprocessing of the left one. The edge between Bob and Adi indicates that

* Corresponding author.
E-mail addresses: zyzhao@sdust.edu.cn, zzysuin@163.com (Z. Zhao), lwqiang@163.com (W. Liu), jinchengqyh@126.com (Y. Qian), nieliqiang@sdu.edu.cn (L. Nie), ylyin@sdu.edu.cn (Y. Yin), zhangyong@siat.ac.cn (Y. Zhang).

https://doi.org/10.1016/j.ins.2018.07.064
0020-0255/

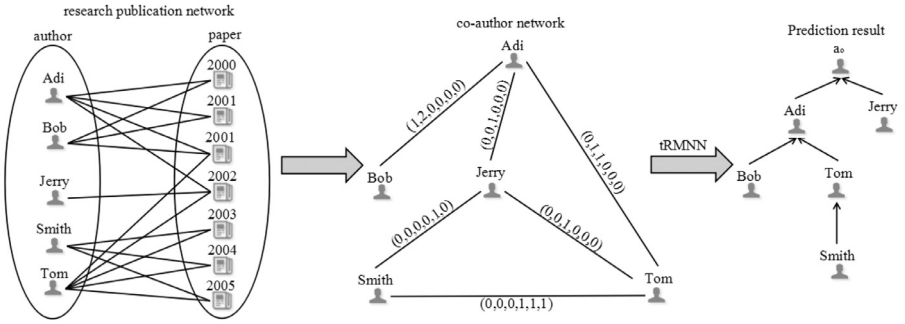

Fig. 1. Example of advising relationship analysis on the co-author network.

there is a co-author relationship between them. The vector (1,2,0,0,0,0) on this edge means that these two researchers have 1 co-authored paper in 2000, 2 co-authored papers in 2001, and have no collaboration since 2002. The right part shows the expected result by a visualized chronological hierarchy. The parent-child relation in the hierarchical structure is the advisor-advisee relationship. The advising path is from the root to the leaves.

However, in reality, identifying the advisor-advisee relationships faces some challenges. (1) Implicit. The advisor-advisee relationship is hidden in the research publication networks. The information that we can find from the network includes only authors, paper title, journal or conference name, and publication year. We have no direct or additional information or labels about such kind of relationship. (2) Time-dependent. The advisor-advisee relationship is highly time-dependent. When a postgraduate student joins his advisor's team, there is a strong advisor-advisee co-author relationship between them. However, after graduation, such relationship turns to be loosen gradually.

In this paper, we formulate the identification of the advisor-advisee relationship as a probabilistic ranking problem. An improved Refresh Gate Recurrent Unit (RGRU) is firstly presented. It contains only one gate unit, holding a simpler structure but performing better than Long Short-Term Memory (LSTM). Taking time dynamics into account, we propose a deep time-aware Advisor-advisee Relationship Mining Model(tARMM). The main contributions of this paper are summarized as follows.

- The RNN is reconstructed by devising a Refresh Gate Recurrent Unit (RGRU), inspired by the idea of variance Recurrent Neural Network (RNN) model like LSTM. RGRU is composed of only one gate unit, with a simpler structure but better performance than LSTM.
- Taking the time factor into account and combining RGRU and DNN together, we present tARMM to solve the problem of advisor-advisee relationship mining.
- In order to evaluate the performance of the proposed model, we conducted some extensive experiments on DBLP data. The experimental results show that tARMM achieves the best performance by comparing with several state-of-the-art models.

The remainder of the paper is organized as follows. Section 2 shows the related works. The problem definition is given in Section 3. Section 4 presents the time-aware Advisor-advisee Relationship Mining Model in detail. Extensive experimental results and discussions are presented in Section 5. Finally, Section 6 concludes this study and discusses the future work.

2. Related work

With the help of social media, people are more active than ever before with more diverse social activities, eg., making friends with each other, browsing and commenting others' posts, and forwarding interesting news [23]. Behind those social behaviors, there are many different types of social relationships and many research works have been done on the relationship mining from the social activities.

The relationships in social media are multi-dimensional since people always connect with each other in many different ways or for different reasons. To address this problem, Tang et al. [12] proposed a clustering based method to extract the latent social dimensions based on the network information. They then utilized them as features for discriminative learning which was capable of determining different social dimensions automatically. But they didn't study the semantic meanings related to the extracted dimensions. Wang et al. [17] proposed a novel collaborative filtering approach to predict the unobserved links in a network (or graph) by using the topological features. However, this approach was still not stable, and could not be extended to the large-scale networks effectively [26]. Factor graph was used to identify the social relationships in recent years. Tang et al. [13] proposed a partially labeled factor graph to predict the types of social relationships in large networks. They then developed a framework for classifying the types of social relationships by learning across heterogeneous networks [10]. Zhuang et al. [27] precisely defined the problem of inferring social ties and proposed a Partially Labeled Pairwise Factor Graph Model (PLP-FGM) to infer the types of social relationships. Tang et al. [11] developed a framework called TranFG to classify the types of social relationships by learning from the heterogeneous networks. The framework incorporated social theories into a factor graph model and thus effectively improved the accuracy of predicting the types of

Table 1
The notations used in this paper.

Notations	Description
$\mathcal{G}$	A graph denoting research publication network
$\mathcal{A}$	The set of authors in $\mathcal{G}$
$\mathcal{P}$	The set of papers in $\mathcal{G}$
$\mathcal{E}$	The set of relations between authors and papers in $\mathcal{G}$
n_a	The number of authors in $\mathcal{G}$
n_p	The number of papers in $\mathcal{G}$
p_i	The ith paper
a_j	The jth author
$\mathcal{G}'$	A graph denoting co-author network
$\mathcal{A}'$	The set of authors in $\mathcal{G}'$
$\mathcal{E}'$	The set of co-author relations between authors in $\mathcal{G}'$
y_x	The advisor of author x
pn_x^t	The number of publications of author x in the year of t
pn_{ij}	A T-dimensional vector to indicate the number of papers that a_i and a_j co-authored each year
T	The time span that the author x co-authored
m	The number of authors collaborated with x
C	The co-author matrix with dimensions $((m+1)\times T)$
$\mathcal{R}$	The set of advisor-advisee relationship
y_{ij}	A binary variable to denote whether a_j is the advisor of a_i
X_S	A $((m+1)\times T)$ dimensional matrix from the perspective of advisee
X_D	A $((m+1)\times T)$ dimensional matrix from the perspective of advisor
X_T	A $((m+1)\times T)$ dimensional matrix with time feature taken into account

social relationships in a target network by using the knowledge from a different source network. They also presented several active learning strategies to further enhance the inferring performance. However, all those above proposed algorithms were based on the factor graph, resulting in computation-intensive.

The problem of identifying the 'Advisor-advisee' relationship has been well studied in these years. Wang et al. [14] presented a two stage framework to transform a collaboration network step by step until achieving the advising hierarchy with ranking. Then they proposed a time-constrained probabilistic factor graph model (TPFG). It took a research publication network as input and modeled the advisor-advisee relationship mining problem using a jointly likelihood objective function. An efficient learning algorithm was also designed to optimize the objective function. This work had been proved with a competitive performance. However, the dynamic feature of the research network in nature was ignored. Li et al. [4] presented an algorithm based on the maximum entropy model to identify the advisor-advisee relationships. But this model had a disadvantage that the feature selection was required beforehand, and different combinations of features would lead to different results. Recently, deep learning based model received tremendous attention from both academia and industry especially in the era of big data [20,21]. Wang et al. [16] presented a deep learning based advisor-advisee relationship identification method. It took both personal properties and network characteristics into consideration. Wang et al. [15] proposed a deep learning based advisor-advisee relationship identification method, which considers the personal properties and network characteristics with a stacked auto-encoder model. However, none of the above methods improved the existing deep learning model.

The main difference between existing studies and our effort is that the previous works mainly focus on a static network, while our study takes the time-varying dynamics into consideration. We design the Refresh Gate Recurrent Unit (RGRU) first, and then present a deep time-aware Advisor-advisee Relationship Mining Model(tARMM).

3. Problem definition

To facilitate the formulations, we define some notations first. In particular, we use bold capital letters (e.g., $\boldsymbol{X}$) and bold small letters (e.g., $\boldsymbol{v}$) to denote matrices and vectors, respectively. The nonbold letters (e.g., x) represent scalars, and Greek letters (e.g., ω) denote parameters. We use flourish letters (e.g., $\mathcal{A}$) to denote sets. If not clarified, vectors are in column forms. The notations used in this paper are listed out in Table 1.

In this paper, we aim to solve the problem of identifying advisor-advisee relationships from the research publication networks. Thus, the research publication network should be defined first, which is shown in Definition 1. It contains all the authors, publications, and the corresponding publication year. An example of such a research publication network is shown in the left part of Fig. 1.

Definition 1. Research publication network $\mathcal{G}$.

The research publication network is represented as a bipartite graph $\mathcal{G} = (\mathcal{A}, \mathcal{P}, \mathcal{E})$, where $\mathcal{A} = \{a_1, a_2, \cdots, a_{n_a}\}$ is a set of all the authors involved in the research publication network; $\mathcal{P} = \{p_1, p_2, \cdots, p_{n_p}\}$ is a set of papers in the research publication network; $\mathcal{E} = \{e_{ik} | i = 1, 2, \ldots, n_a; k = 1, 2, \ldots, n_p\}$ is the edge set of the bipartite graph $\mathcal{G}$, each edge e_{ik} means that a_i is one of the authors of p_k.

Definition 1 is a bipartite graph which depicts the author-paper relationships. What we concern is the co-authoring relationships and how to fully utilize the above information to mine the advisor-advisee relationships. Therefore, we should define and extract the co-author network from the research publication network. The definition of the co-author network is presented in Definition 2. An example is shown in the middle part of Fig. 1.

Definition 2. Co-author network $\mathcal{G}'$.

We construct $\mathcal{G}' = (\mathcal{A}', \mathcal{E}', \{\boldsymbol{pn_{ij}}\}_{e_{ij}\in\mathcal{E}'})$ from $\mathcal{G}$, where $\mathcal{A}' = \{a_0, a_1, \cdots, a_{n_a}\}$ is a set of authors, in which a_0 is a virtual author. As to the author a_i, we assume y_{a_i} is the advisor. If $y_{a_i} \notin \mathcal{A}$, then we consider $y_{a_i} = a_0$. $\mathcal{E}'$ is the set of all the co-authoring relationships, $\mathcal{E}' = \{e_{ij} | i = 1, 2, \ldots, n_a; j = 1, 2, \ldots, n_a\}$. $\boldsymbol{pn_{ij}}$ is a vector related to e_{ij}. It indicates the number of papers that a_i and a_j co-authored during a certain time interval. Thus, given one author a_i, we can use all the $\boldsymbol{pn_{ij}}(j = 0, 1, \ldots, n_a)$ to indicate all his/her publications.

In order to fully utilize the co-author network and process it with the proposed model, we define the co-author matrix in Definition 3.

Definition 3. Co-author matrix $\boldsymbol{C}$

As to the author x in $\mathcal{A}$, it is assumed that x has collaborated with m authors. Then the set of co-authors of x could be represented by $\mathcal{A}_x$. $\mathcal{A}_x = \{b_0, b_1, \cdots, b_m\}$, where $b_0 = a_0$. Let $pn^t_{xb_j}$ denote the number of papers that x published with the co-author b_j in the year of t. Let T denote the time span that the author x co-authored. Then we can get the co-author matrix $\boldsymbol{C}$ for x as follows.

$$\boldsymbol{C} = \begin{bmatrix} pn^0_{xb_0} & \cdots & pn^{T-1}_{xb_0} \\ \vdots & \cdots & \vdots \\ pn^0_{xb_m} & \cdots & pn^{T-1}_{xb_m} \end{bmatrix}. \tag{1}$$

With the above three definitions (Definition 1, 2 and 3), our target is to predict the advisor-advisee relationships. The definition of the advisor-advisee relationship is given as follows.

Definition 4. Advisor-advisee relationship $\mathcal{R}$.

$$\mathcal{R} = \{y_{ij} | i = 0, 1, 2, \ldots, n_a; j = 0, 1, 2, \ldots, n_a\}. \tag{2}$$

$\mathcal{R}$ is used to indicate whether the authors i and j have advisor-advisee relationship. The value of each y_{ij} is defined as follows:

$$y_{ij} = \begin{cases} 1, & \text{if } a_j \text{ is the advisor of } a_i \\ 0, & otherwise \end{cases}. \tag{3}$$

The target of this paper is to predict the advisor for each author x. It needs to answer these two questions: who is the advisor of x and to what degree?

4. Our approach

In this section, we first give some preliminary knowledge as the prerequisite of our approach, then present our models and algorithms in detail.

4.1. Assumptions

Preliminary knowledge is needed for recognizing the interesting semantic relationships. Here we present two assumptions based on the commonsense knowledge about the advisor-advisee relationships.

Assumption 1.

$$First(y_x) < First(x), x = 1, 2, 3, \ldots, n_a. \tag{4}$$

This assumption is proposed by Wang et. al in literature [14]. It is reasonable that the advisor has a longer publishing history than the advisee in most cases. In other words, the time that an advisor publishes the first paper should be earlier than that of his/her advisee. In the above formula, $First(x)$ denotes the time when the author x published his/her first paper. It can be extracted from the vector pn_{xj} by identifying the first nonzero value. This assumption will be used in generating the co-author matrix $\boldsymbol{C}$, which can exclude those collaborators who are not likely to be x's advisors. In this way, each co-author appeared in matrix $\boldsymbol{C}$ is the advisor candidate of x. There is also a special case. If there is no advisor for x in $\{b_1, b_2 \ldots b_m\}$, then we let b_0 be x's advisor. That is, $y_x = b_0$. This assumption will be used in Section 4.3.

Assumption 2.

$$P(y_{ij} = 1) \propto \frac{1}{First(a_i, a_j)}. \tag{5}$$

This assumption means that the earlier an author collaborates with x, the more possible he/she is the advisor of x. It is suitable for general situations. Generally speaking, most students are engaged in the scientific research and publish their first research paper under the supervision of their advisors. In the above formula, $First(a_i, a_j)$ refers to the year when author a_i collaborates with a_j for the first time. This assumption is also in accordance with our common sense. Generally speaking, the first paper of a student is often published under the supervision of his/her advisor. So, the co-author of a student's first paper is more likely to be the advisor. This assumption will be used to determine the input sequence of the RGRU.

To handle the advisor-advisee relationship mining problem, we should first formalize the input by constructing three co-author sub-matrices, $\boldsymbol{X_S}$, $\boldsymbol{X_D}$ and $\boldsymbol{X_T}$. Then we propose a time-aware Advisor-advisee Relationship Mining Model (tARMM).

4.2. The construction of the co-author matrix

In this sub-section, we first construct the co-author network $\mathcal{G}'$ from the original publishing network $\mathcal{G}$, and then extract the co-author matrices.

The co-author network can be viewed from two aspects: co-authors and time. In the former situation, the collaborating relationships can be derived easily from the authors and the advisor candidates. The collaborating relationships are denoted by $\boldsymbol{C}$. Let the vector $\boldsymbol{s}$, shown in Eq. (6), represent the details of author x. Similarly, we use the matrix $\boldsymbol{D}$ to represent the collaborating details of all advisor candidates which is shown in Eq. (7). Then, we use $\boldsymbol{s}$ and $\boldsymbol{D}$ to normalize the co-author matrix $\boldsymbol{C}$ to get $\boldsymbol{X_S}$ and $\boldsymbol{X_D}$, as described in Eq. (8) and (9), respectively. Here, $\boldsymbol{X_S}$ is a matrix from the aspect of the advisee, while $\boldsymbol{X_D}$ is a matrix from the aspect of the advisor. Let X_{Sij} denote the ratio between the number of x' co-authoring papers with his advisor candidate b_i and x's total publications in year j. Let X_{Dij} denote the ratio between the number of x' co-authoring papers with his advisor candidate b_i and b_i's total publications in year j.

$$\boldsymbol{s} = \left[\ s^0, \quad \cdots, \quad s^{T-1}\ \right], \tag{6}$$

where,

$$s^t = \begin{cases} \frac{1}{pn_x^t}, & pn_x^t \neq 0 \\ 0, & pn_x^t = 0 \end{cases}$$

$$\boldsymbol{D} = \begin{bmatrix} D_{b_0}^0 & \cdots & D_{b_0}^{T-1} \\ \vdots & \cdots & \vdots \\ D_{b_m}^0 & \cdots & D_{b_m}^{T-1} \end{bmatrix}, \tag{7}$$

where,

$$D_{b_i}^t = \begin{cases} \frac{1}{pn_{b_i}^t}, & pn_{b_i}^t \neq 0 \\ 0, & pn_{b_i}^t = 0 \end{cases}$$

$$\boldsymbol{X_S} = \boldsymbol{C} \cdot \boldsymbol{s}. \tag{8}$$

$$\boldsymbol{X_D} = \boldsymbol{C} \cdot \boldsymbol{D}. \tag{9}$$

In the latter part, we consider the time factor. Let $\boldsymbol{X_T}$ be the co-author matrix with the publication time, which is shown in Eq. (10).

$$\boldsymbol{X_T} = \begin{bmatrix} X_{T_{b_0}}^0 & \cdots & X_{T_{b_0}}^{T-1} \\ \vdots & \cdots & \vdots \\ X_{T_{bm}}^0 & \cdots & X_{T_{b_m}}^{T-1} \end{bmatrix}, \tag{10}$$

where,

$$X_{T_{b_i}}^t = \begin{cases} 1, & pn_{xb_i}^t \neq 0 \\ 0, & pn_{xb_i}^t = 0 \end{cases}$$

4.3. The Construction of tARMM

In this sub-section, we propose a time-aware Advisor-advisee Relationship Mining Model(tARMM) illustrated in Fig. 2. This model considers $\boldsymbol{X_S}$, $\boldsymbol{X_D}$ and $\boldsymbol{X_T}$ respectively. It calculates the probability matrix based on the time structure and picture-like matrix. Then, it generates the final probability of advisor-advisee via a fully connected layer. As for the processing of $\boldsymbol{X_T}$, a Refresh Gate Recurrent Unit (RGRU) of the inverse time is implemented. As for the processing of $\boldsymbol{X_S}$ and $\boldsymbol{X_D}$, the deep neural network is used.

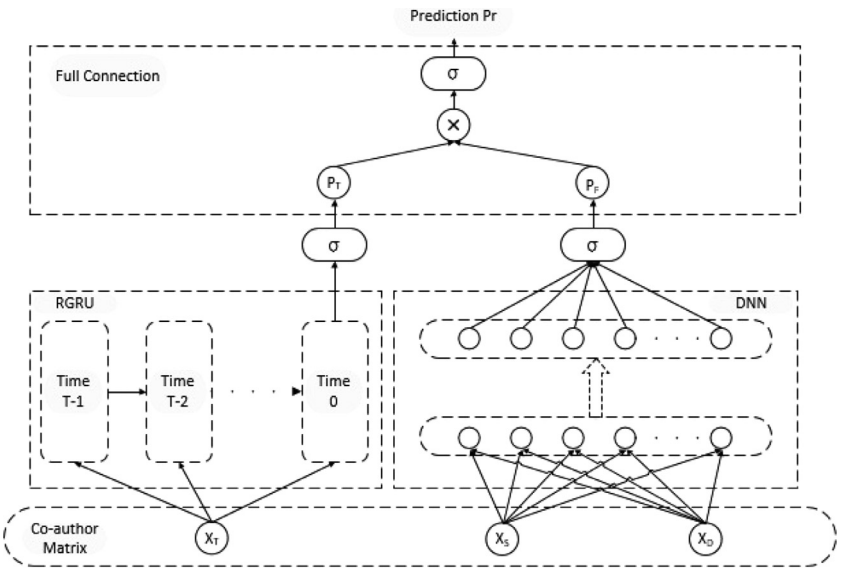

Fig. 2. Schematic illustration of our proposed tARMM model.

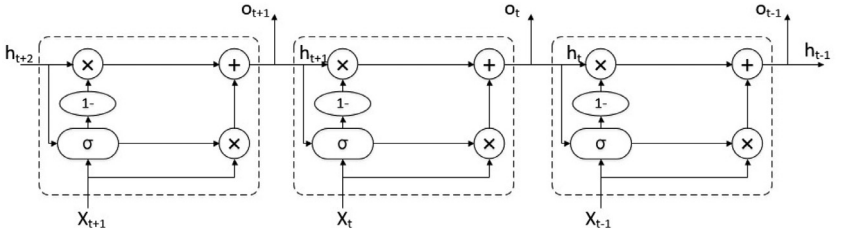

Fig. 3. An illustration of RGRU.

4.3.1. The Design of RGRU

As we all know, LSTM and GRU perform very well in dealing with sequence data, but their structures are still complex. We hope to establish a simple but effective model to handle the problem of the advisor-advisee relationship mining. According to [1], we have learned that RNN with gate structure is better than those without it. According to [2], we have learned that the forget gate and the input gate are more essential than the output gate in RNN. So we integrate the forget gate and the input gate, but ignore the output gate. Then we design a simple structure called the Refresh Gate Recurrent Unit (RGRU). RGRU has only one gate unit, however, the experimental results have shown that RGRU is effective on mining the advisor-advisee relationship.

Fig. 3 is an illustration of RGRU. As for the matrix $\boldsymbol{X_T}$ with the publication time, it can be seen from Assumption 2 that the earlier an author collaborates with x, the more likely he/she is an advisor of x. Thus, the matrix $\boldsymbol{X_T}$ is processed with respect to the columns by RGRU to obtain a time-based advisor probability matrix. In RGRU, we have the following equations as shown in Eq. (11) at time t.

$$\begin{cases} \boldsymbol{r_t} = \sigma(\boldsymbol{W_h h_{t+1}} + \boldsymbol{W_x x_t} + \boldsymbol{b_r}), \\ \boldsymbol{h_t} = (\mathbf{1} - \boldsymbol{r_t}) \circ \boldsymbol{h_{t+1}} + \boldsymbol{r_t} \circ \boldsymbol{x_t}, \\ \boldsymbol{o_t} = \boldsymbol{h_t}, \end{cases} \tag{11}$$

where

- $\boldsymbol{r_t}$ is the state of the refresh gate at time t;
- $\boldsymbol{W_h}$ and $\boldsymbol{W_x}$ are the weight matrices of the refresh gate;
- $\boldsymbol{h_{t+1}}$ is the state of the unit at time $t+1$;
- $\boldsymbol{x_t}$ is the input at time t;
- $\boldsymbol{b_r}$ is the biases of the refresh gate;
- $\boldsymbol{h_t}$ is the state of the unit at time t;
- $\boldsymbol{o_t}$ is the output at time t;

With RGRU, we can calculate the probability of one author being the advisor of x. The algorithm to mine the advisor-advisee relationship with the proposed RGRU is described in Algorithm 1.

Algorithm 1 Advisor-advisee relationship mining based on RGRU.

Input: co-author matrix X_T of author x
Output: advising probability of the author x (P_T)
1: $h_T \leftarrow 0$
2: $r_T \leftarrow 0$
3: **for** $t \leftarrow T-1$ to 0 **do**
4: $\quad r_t \leftarrow \sigma(W_h h_{t+1} + W_x x_t + b_r)$
5: $\quad h_t \leftarrow (1-r_t) \circ h_{t+1} + r_t \circ x_t$
6: $\quad o_t \leftarrow h_t$
7: **end for**
8: $P_T \leftarrow o_0$
9: return P_T

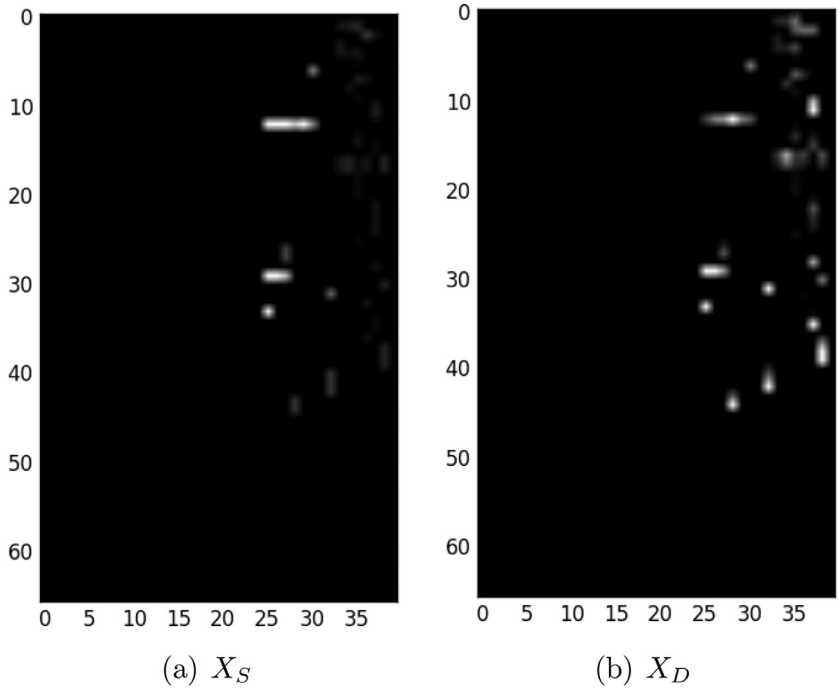

Fig. 4. An illustration of picture-like matrix.

In Algorithm 1, the main time overhead focuses on the state updating of the refresh gate and unit, which is in the *for* loop. Thus, the time complexity of the above algorithm is $O(T)$, where T is the spanning time of the input data.

4.3.2. Probability model based on picture-like matrix

As stated in Section 4.2, $\boldsymbol{X_S}$ and $\boldsymbol{X_D}$ denote the collaborating relationships from the aspect of the advisee and the advisor, respectively. Showing in the form of bitmap, $\boldsymbol{X_S}$, can be regarded as a 66×40 grayscale image. From the preliminary knowledge, we know that if a person is the advisor of x, there must be continuous pixel values constituting a special image similar to '-'. Note that different bitmaps lead to different characteristics. Therefore, we don't use the ordinary array traversal methods. Considering that this is a pixel-level problem, we use deep neural network to solve it. As shown in Fig. 4, it depicts the $\boldsymbol{X_S}$ bitmap and $\boldsymbol{X_D}$ bitmap of the author with ID '4386'. The advisor of the author '4386' is the person with ID '19748'.

In this part, $\boldsymbol{X_S}$ and $\boldsymbol{X_D}$ are combined together to form a double color channel bitmap, called the picture-like matrix $\boldsymbol{X}$. In this way, the objective is to find the line number of a particular picture in $\boldsymbol{X}$. Let U denote the number of hidden units, L denote the number of hidden layers, and U' denote the number of nodes in each layer. Then $U' = 2$ if it is in input layer; $U' = 1$ if it is in output layer; and $U' = U$ if it is in the hidden layer. Let y_n^l denote the output of the n^{th} node in the l^{th} layer in DNN, it can be calculated according to Eq. (12). The input layer is initialized by $y_1^0 = X_D$ and $y_2^0 = X_S$. The advising

probability $\boldsymbol{P_F}$ based on the picture-like matrix is the output of the DNN. It can be calculated according to the Eq. (13).

$$y_n^l = \sigma\left(\sum_{i=1}^{U'} y_i^{l-1} w_n^{l-1}{}_i + b_n^{l-1}\right). \tag{12}$$

$$P_F = ReLU\left(\sum_{i=1}^{U} x_i^L w_i^L + b^L\right). \tag{13}$$

The algorithm to compute the advisor probability based on the picture-like matrix is shown in Algorithm 2.

Algorithm 2 Advisor-advisee relationship mining based on DNN.

Input: co-author matrix X_S and X_D of author x
Output: advising probability of the author x (P_F)
1: //input layer
2: $y_1^0 \leftarrow X_D$
3: $y_2^0 \leftarrow X_S$
4: //hidden layer
5: **for** $n \leftarrow 1$ to U **do**
6: $\quad y_n^1 \leftarrow \sigma(\sum_{i=1}^{2} y_i^0 w_{ni}^0 + b_n^0)$
7: **end for**
8: **for** $l \leftarrow 2$ to L **do**
9: $\quad$ **for** $n \leftarrow 1$ to U **do**
10: $\quad\quad y_n^l \leftarrow \sigma(\sum_{i=1}^{U} y_i^{l-1} w_n^{l-1}{}_i + b_n^{l-1})$
11: $\quad$ **end for**
12: **end for**
13: //output layer
14: $P_F \leftarrow ReLU(\sum_{i=1}^{U} y_i^L w_i^L + b^L)$
15: return P_F

In Algorithm 2, the state of each node in DNN will be calculated in the nested '*for*' loop. Thus, the complexity of this algorithm is $O(L \times U^2)$, where L and U denote the number of hidden layers and the hidden units, respectively.

4.4. Relationship Identification with tARMM

Given $\boldsymbol{P_T}$ and $\boldsymbol{P_F}$, the final advisor probability matrix can be derived via a fully connected layer, as shown in the upper block of Fig. 2. The formula is given in Eq. (14). The person with the highest probability will be predicted to be the advisor.

$$\boldsymbol{Pr} = \sigma(\boldsymbol{P_F} \cdot \boldsymbol{P_T}). \tag{14}$$

In the model of tARMM, the cross entropy is used as the loss function, as shown in the following. The model parameters can be calculated by minimizing the loss function.

$$\begin{aligned}\boldsymbol{loss} = -\frac{1}{d}\sum_{i=0}^{n}(&\boldsymbol{y} \cdot \log P(\boldsymbol{y}|\boldsymbol{X_T}, \boldsymbol{X_S}, \boldsymbol{X_D}) \\ &+(\boldsymbol{1}-\boldsymbol{y}) \cdot \log(\boldsymbol{1} - P(\boldsymbol{y}|\boldsymbol{X_T}, \boldsymbol{X_S}, \boldsymbol{X_D}))).\end{aligned} \tag{15}$$

The algorithm to predict the advisor based on tARMM is presented as follows:

In Algorithm 3, all the authors should be traversed via co-author network. Since Algorithm 1 and Algorithm 2 are called as sub-routines, the time complexity of Algorithm 3 is $O(n_a \times (L \times U^2 + T))$, where n_a is the number of authors. Note that T is much smaller than $(L \times U^2)$. So, the time complexity of this algorithm is $O(n_a \times L \times U^2)$.

5. Experimental results

In this section, we present extensive experiments to evaluate the performance of the proposed algorithm.

5.1. Experiment settings

(1) Experimental Data

DBLP is the on-line reference with open bibliographic information on journals and proceedings in computer science. It is an important data source for researchers. The basic publication information includes authors, paper title, publication year,

Algorithm 3 Advisor-advisee relationship mining based on tARMM.

Input: the co-author network G'
Output: the set of advisor probability based on tARMM P_r
create the set $setP_r$
for each author $x \in G'$ **do**
  construct matrix C, D, S according to Equation (1), (6), (7)
  construct matrix X_S, X_D, X_T according to Equation (8), (9), (10)
  call Algorithm 1 to get P_T of author x
  call Algorithm 2 to get P_F of author x
  $P_r \leftarrow \sigma(P_F \cdot P_T)$
  add P_r to $setP_r$
end for
return $setP_r$

publication venue, volumes, and pages. Obtained from DBLP, our data set consists of 654,628 authors and 1,076,946 publications from 1970 to 2008. The labels indicating the advisor-advisee relationships are obtained from the home page of the advisors, the Mathematics Genealogy project and AI Genealogy project. By Assumption 1, we get 1322 valid advisor-advisee pairs for the model training. For each advisee, we calculate the co-author matrix based on the publication information in DBLP.

(2) Experimental Environment

We perform our experiments on a personal computer. The machine is equipped with Intel dual-core 2.4 GHz CPU, 16GB Memory and 1T Disk Storage. The operating system is Ubuntu 16.04. The software tools that we used are matlab and PyCharm. In the data preprocessing stage, we use Matlab to do data preprocessing. In other experiments, including the implementation of the proposed model and the comparison with other methods, our programming language is python.

(3) Performance Metrics

To evaluate our method, we use three performance metrics: precision (P), recall (R) and F_1 score (F_1). They are defined as follows:

$$\begin{cases} P = \dfrac{TP}{TP + FP}, \\ R = \dfrac{TP}{TP + FN}, \\ F_1 = \dfrac{2PR}{P + R}, \end{cases} \tag{16}$$

where

- TP denotes the number of true positives;
- FP denotes the number of false positives;
- FN denotes the number of false negatives.

(4) Parameter Settings

Parameter setting is an important procedure since tARMM has a good self-learning ability. Considering that the structure of tARMM is based on DNN, we need to determine the number of hidden layers L and the number of hidden units U first. As we all know, more hidden layers and units could achieve better performance. However, it would increase the number of parameters tremendously, resulting in a large overhead. Thus, the suitable hidden layers and hidden units should be determined. The other parameters in tARMM are initialized as random values following the normal distribution.

In this paper, the parameter L is tested from 1 to 8, and U is tested from 30 to 100. The performances of tARMM are shown in Fig. 6. As can be seen from Fig. 5(a), all the metrics including precision, recall and F1 score are improved with the increase of hidden layers. The measure of recall achieves the best value at '$L = 7$', while the precision and F1 achieve the best performance at '$L = 6$'. Thus, after a comprehensive consideration, the number of hidden layers is set to be 6. In the test of hidden units settings, we evaluate the performance every 10 units, and the results are shown in Fig. 5(b). The metric of recall gets the highest value at '$U = 70$'. The performance on the other two kinds of metrics becomes stable at '$U = 80$'. That is, with the number of hidden units increasing, the performance of tARMM tends to be stable. Considering the computing cost, the number of hidden units is set to be 80 in our following experiments.

5.2. Experimental results and discussion

(1) Effect of the Optimization Strategy

As we all know, different optimization strategies have different effects on the efficacy of the algorithm. Stochastic Gradient Descent (SGD) is a popular algorithm that has achieved good performance on a variety of machine learning tasks.

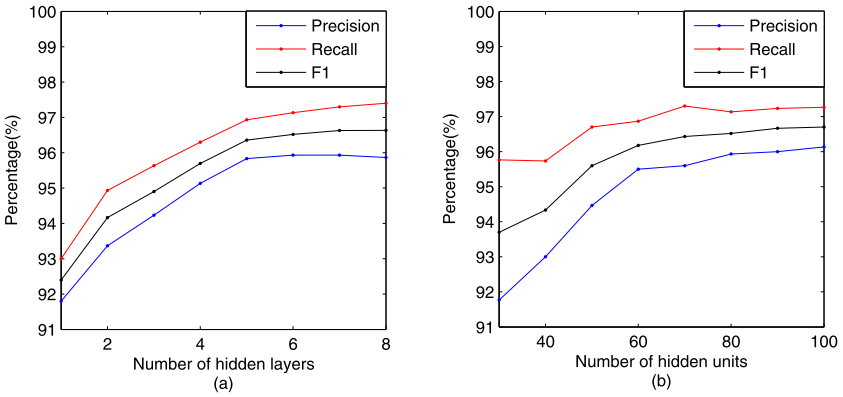

Fig. 5. The performance with different hidden layers and hidden units.

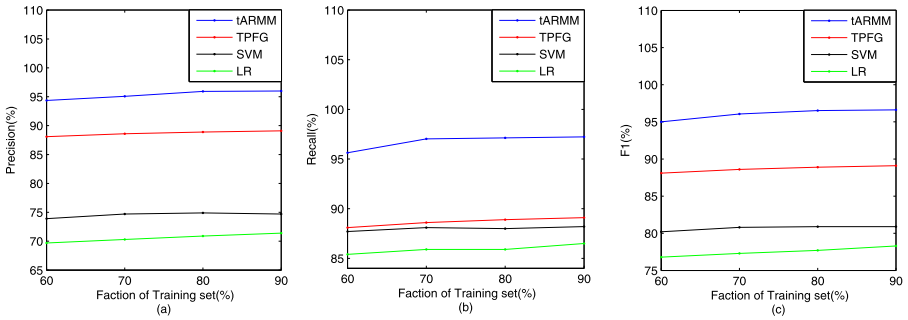

Fig. 6. Performance comparison between tARMM and other methods in terms of precision, recall and F1.

Table 2
Effects of the optimizing algorithms.

	Recall	Precision	F1-score
Adam	97.1%	95.9%	96.5%
SGD	92.6%	91.4%	92.1%

The Adam method is another algorithm for the first-order gradient-based optimization of stochastic objective functions. It is straightforward to implement, computationally efficient, and well suitable for handling problems with big data or large number of parameters [3].

In this paper, the above two types of optimization methods are compared to train the model separately. The comparative results are shown in Table 2. According to Table 2, we can see that the Adam algorithm outperforms SGD algorithm. Thus, in the following experiments, we adopt the Adam algorithm to learn parameters.

(2) Evaluation of tARMM

The following three methods including Logistic Regression, Support Vector Machine, and TPFG are selected as the baselines. Logistic Regression(LG) and Support Vector Machines(SVM) are traditional supervised learning approaches. If the advisor-advisee pairs are treated as positive examples and non advisor-advisee pairs are treated as negative examples, then the problem of advisor-advisee relationship mining is transformed into a classification problem. TPFG is another algorithm to identify adivsor-advisee relationship. It has been shown to be very efficient and accurate in some previous work [14].

In this paper, we compare the proposed tARMM with the above three methods. In the experiments, the data set is divided into two parts, training set and testing set. In order to investigate the performance of tARMM regarding to the size of training set, we set different proportions on these two parts. The comparative results are shown in Fig. 6. From Fig. 6, we can see that LR and SVM have made a fairly good job. But the proposed tARMM is 10-15% better than them. The main reason is that those methods do not explore the network structure or time dependence. TPFG has achieved a competitive performance, as it takes the network structure and time factor into account. But it is still 5–8% worse than the proposed tARMM in terms of precision and F_1 metrics. In summary, tARMM not only considers the network structure and the time factor, but also learns features automatically. Thus, it has the best performance among all these methods. Another observation is that all methods tend to make a better job with the increasing of the training set.

Table 3
Performance comparison of tARMM with other deep models.

	RNN	LSTM	RGRU	tARMM
P	82.4%	91.1%	92.2%	95.9%
R	94.5%	95.9%	96.8%	97.1%
F_1	88%	93.4%	94.4%	96.5%

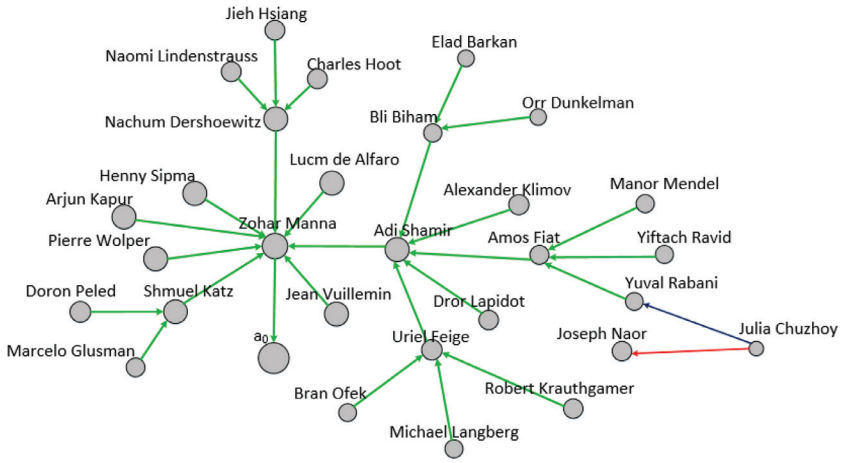

Fig. 7. Visualization of some advisor-advisee relationships.

(3) Comparison with Deep Models:

In order to demonstrate the effectiveness of the tARMM, we also compare it with some deep models like RNN and LSTM. In this experiment, the fraction of training data is set to be 80%. The experimental results are shown in Table 3. According to Table 3, we can see that the proposed tARMM performs better than RNN and LSTM.

The reason behind these results mainly lies in the design of RGRU. For one thing, the RGRU is designed with one gate unit, which makes it outperform RNN obviously. For another thing, it has a simpler structure than LSTM, which makes it more efficient. Furthermore, RGRU is designed with the inverse input, which could fully utilize the time features of the co-author network. Let us consider the co-author matrix X_T. According to Assumption 2, we can see that it is a sparse matrix, in which nonzero values distributed in those columns with larger index while zero elements are with smaller indexes. In RGRU, we adopt the inverse input of X_T, enabling RGRU to record nonzero elements first. The more zero elements it receives, the more quickly it decays. Thus, tARMM is able to fully utilize the time features of the co-author network to improve its performance.

(4) Case Study

Fig. 7 visualizes some identified advisor-advisee relationships. A vertex denotes an author. The directed edge in green from A to B means that B is correctly predicted to be the advisor of A. The directed edge in red means that the advisor-advisee relationship is predicted incorrectly. The advisor-advisee relationships in reality but not predicted with our approach are denoted by directed edges in blue. From Fig. 7, we can see that most of advisor-advisee relationships are predicted correctly. As for the author named 'Julia Chuzhoy', 'Joseph Naor' is predicted to be her advisor with our algorithms. However, 'Yuval Rabani' is the true advisor of 'Julia Chuzhoy'.

6. Conclusion and future work

This paper studies the identification of the advisor-advisee relationships in research publication networks. To accomplish this task, we firstly define and formalize the co-author matrices. Then a deep time-aware advisor-advisee relationship learning model equipped with the improved refresh gate recurrent units (RGRU) is presented. Three algorithms are also proposed to determine the probability of the advisor-advisee relationship. We evaluate our model over the publicly accessible DBLP data and compare its performance with that of several state-of-the-art models. The results show that the tARMM achieves the best performance. In our future work, we will study the effect of those hidden relationships on the influence analysis.

Acknowledgement

We sincerely appreciate the editor and anonymous reviewers for their valuable and constructive comments. We also would like to thank Dr. Zhang Chunyun for her suggestions during the writing of this paper. This research is supported

by the National Natural Science Foundation of China (Grant No. 61303167, 61433012, 61573219, 61672332 and 61702306), Ministry of Education in China Foundation for Humanities and Social Sciences (Grant No. 17YJCZH262 and 16YJCZH041), Natural Science Foundation of Shandong Province (Grant No. ZR2018BF013, ZR2017BF026), the Key R&D Plan of Shandong Province (Grant No. 2016ZDJS02A11, 2018GGX101045), San Jin Scholars of Shanxi Province (2016769), and Scientific Research Foundation of Shandong University of Science and Technology for Innovative Team (Grant No. 2015TDJH102).

References

[1] K. Greff, R.K. Srivastava, J. Koutnik, B.R. Steunebrink, J. Schmidhuber, Lstm: a search space odyssey, IEEE Trans Neural Netw Learn Syst 28 (10) (2016) 2222–2232.
[2] R. Jozefowicz, W. Zaremba, I. Sutskever, An empirical exploration of recurrent network architectures, in: Proceedings of the 32nd International Conference on Machine Learning, JMLR, 2015, pp. 2342–2350.
[3] D. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations, JMLR, 2015, pp. 1–15.
[4] Y.J. Li, Advisor-advisee relationship identification based on maximum entropy model, Acta Physica Sinica 62 (16) (2013) 581–586.
[5] Z. Li, R.S. Wang, S. Zhang, X.S. Zhang, Quantitative function and algorithm for community detection in bipartite networks, Inf Sci (Ny) 367 (C) (2016) 874–889.
[6] L. Liu, J. Tang, J. Han, M. Jiang, S. Yang, Mining topic-level influence in heterogeneous networks, in: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, ACM, 2010, pp. 199–208.
[7] Q. Liu, E. Chen, H. Xiong, C.H.Q. Ding, J. Chen, Enhancing collaborative filtering by user interest expansion via personalized ranking, IEEE Transactions on Systems, Man & Cybernetics, Part B Cybernetics 42 (1) (2012) 218–233.
[8] Q. Liu, B. Xiang, N.J. Yuan, E. Chen, H. Xiong, Y. Zheng, Y. Yang, An influence propagation view of pagerank, ACM Trans Knowl Discov Data 11 (3) (2017) 1–30.
[9] V.C. Storey, J.C. Trujillo, S.W. Liddle, Research on conceptual modeling: themes, topics, and introduction to the special issue, Data & Knowledge Engineering 98 (2015) 1–7.
[10] J. Tang, T. Lou, J. Kleinberg, Inferring social ties across heterogenous networks, in: Proceedings of the 5th ACM international conference on Web search and data mining, ACM, 2012, pp. 743–752.
[11] J. Tang, T. Lou, J. Kleinberg, S. Wu, Transfer learning to infer social ties across heterogeneous networks, ACM Trans. Inf. Syst. 34 (2) (2016) 1–43.
[12] L. Tang, H. Liu, Relational learning via latent social dimensions, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 817–826.
[13] W. Tang, H. Zhuang, J. Tang, Learning to infer social ties in large networks, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2011, pp. 381–397.
[14] C. Wang, J. Han, Y. Jia, J. Tang, D. Zhang, Y. Yu, J. Guo, Mining advisor-advisee relationships from research publication networks, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge discovery and data mining, ACM, 2010, pp. 203–212.
[15] W. Wang, J. Liu, F. Xia, I. King, H. Tong, Shifu: Deep learning based advisor-advisee relationship mining in scholarly big data, in: Proceedings of the 26th International Conference on World Wide Web Companion, in: WWW '17 Companion, International World Wide Web Conferences Steering Committee, 2017, pp. 303–310.
[16] W. Wang, J. Liu, S. Yu, C. Zhang, Z. Xu, F. Xia, Mining advisor-advisee relationships in scholarly big data: A deep learning approach, in: Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries, ACM, 2016, pp. 209–210.
[17] X. Wang, D. He, D. Chen, J. Xu, Clustering-based collaborative filtering for link prediction, in: Proceedings of the 29th AAAI Conference on Artificial Intelligence, AAAI, 2015, pp. 332–338.
[18] L. Wu, Y. Ge, Q. Liu, E. Chen, R. Hong, J. Du, M. Wang, Modeling the evolution of users' preferences and social links in social networking services, IEEE Trans Knowl Data Eng 29 (6) (2017) 1240–1253.
[19] Y. Wu, S. Venkatramanan, D.M. Chiu, Research collaboration and topic trends in computer science: An analysis based on ucp authors, in: Proceedings of the 24th International Conference on World Wide Web, ACM, 2015, pp. 1045–1050.
[20] Q. Zhang, L.T. Yang, Z. Chen, P. Li, Pphopcm: privacy-preserving high-order possibilistic c-means algorithm for big data clustering with cloud computing, IEEE Trans. Big Data PP (99) (2017) 1–11.
[21] Q. Zhang, L.T. Yang, Z. Chen, P. Li, A survey on deep learning for big data, Information Fusion 42 (2018) 146–157.
[22] Z. Zhao, S. Feng, Q. Wang, J.Z. Huang, G.J. Williams, J. Fan, Topic oriented community detection through social objects and link analysis in social networks, Knowl Based Syst 26 (2012) 164–173.
[23] Z. Zhao, C. Li, Y. Zhang, J.Z. Huang, J. Luo, S. Feng, J. Fan, Identifying and analyzing popular phrases multi-dimensionally in social media data, International Journal of Data Warehousing & Mining 11 (3) (2015) 98–112.
[24] Z. Zhao, Y. Zhang, C. Li, L. Ning, J. Fan, A system to manage and mine microblogging data, Journal of Intelligent & Fuzzy Systems 33 (1) (2017) 1–11.
[25] Z. Zheng, F. Ye, R.H. Li, G. Ling, T. Jin, Finding weighted k-truss communities in large networks, Inf Sci (Ny) 417 (C) (2017) 344–360.
[26] A. Zhiyuli, X. Liang, X. Zhou, Learning structural features of nodes in large-scale networks for link prediction, in: Proceedings of the 30th AAAI Conference on Artificial Intelligence, 2016, pp. 4286–4287.
[27] H. Zhuang, J. Tang, W. Tang, T. Lou, A. Chin, X. Wang, Actively learning to infer social ties, Data Mining & Knowledge Discovery 25 (2) (2012) 270–297.