PAPER

HeteroRWR: A Novel Algorithm for Top-*k* Co-Author Recommendation with Fusion of Citation Networks

Sufen ZHAO[†,††a)], ***Member***, **Rong PENG**[†b)], **Meng ZHANG**[††], ***and*** **Liansheng TAN**[††,†††], ***Nonmembers***

SUMMARY It is of great importance to recommend collaborators for scholars in academic social networks, which can benefit more scientific research results. Facing the problem of data sparsity of co-author recommendation in academic social networks, a novel recommendation algorithm named HeteroRWR (Heterogeneous Random Walk with Restart) is proposed. Different from the basic Random Walk with Restart (RWR) model which only walks in homogeneous networks, HeteroRWR implements multiple random walks in a heterogeneous network which integrates a citation network and a co-authorship network to mine the *k* mostly valuable co-authors for target users. By introducing the citation network, HeteroRWR algorithm can find more suitable candidate authors when the co-authorship network is extremely sparse. Candidate recommenders will not only have high topic similarities with target users, but also have good community centralities. Analyses on the convergence and time efficiency of the proposed approach are presented. Extensive experiments have been conducted on DBLP and CiteSeerX datasets. Experimental results demonstrate that HeteroRWR outperforms state-of-the-art baseline methods in terms of precision and recall rate even in the case of incorporating an incomplete citation dataset.

key words: *heterogeneous networks, social networks, friend recommendation, co-author recommendation, random walk with restart*

1. Introduction

Recommender Systems (RSs) have been one of the hottest research topics in artificial intelligence for more than ten years. Due to the fact that, the RSs' techniques can significantly enhance the business value of enterprises and efficiently reduce the information overloaded for users, many enterprises and companies use RSs to recommend products and services to customers in the domains, such as music recommendation, news recommendation, image recommendation, personalized point-of-interest (POI) recommendation and friend recommendation, etc. [1].

In this study, we focus on top-*k* co-author recommendation problem in the heterogeneous bibliographic network, for a large number of studies have shown that scholars with more collaborative relationships tend to publish more papers with better quality [2]. Recommending the *k* most potential co-authorship partnerships will help scholars to establish more and better cooperative relations and promote more scientific research results. On the other hand, co-author recommendation problem is closely related to link prediction problem. Many works treat them as the same while the difference is recommendation tasks output a ranking list. Link prediction is the heart of social graph mining, which can help us discover the intrinsic mechanism of social relationship building and reveal the essence of social network evolution. In addition, the link recommendation and prediction models based on one complex network can usually be generalized to other types of complex networks. Therefore, this research of co-author recommendation has both important commercial and theoretical value.

However, even though the co-author recommendation problem has attracted the attention of some scholars, its research still faces some challenges:

- real-world social networks are usually extremely sparse [3], which causes many traditional recommendation models to face the problem of cold start. Hence, in order to overcome the sparsity problem, how best to use additional information in recommendation models is a challenge.
- it's imperative that recommendation models need to be highly time efficient on large-scale networks, because in the real world, academic collaborative RSs are often built on large-scale graph structures.
- recommendation models need to be more adaptable to incomplete and noisy data sets, for real-world data are often incomplete and noisy.

As we know, RWR is a kind of random walk model where we pick a node and move following a random walk with probability α or we return to the starting node with probability $1 - \alpha$ [4]. Compared to other local similarity-based measures, RWR can capture the whole network topology information well and easily integrate more features into the model. However, the basic RWR algorithm is defined on homogeneous networks, hence it is difficult to overcome the data sparsity of the co-authorship network.

Due to the abundant information contained in the citation network, we design a novel method named HeteroRWR. Different from the basic RWR which only defines random walk on homogeneous networks, HeteroRWR combines the citation network with the co-authorship network and performs multiple random walks in the heterogeneous

Manuscript received April 14, 2019.
Manuscript revised August 18, 2019.
Manuscript publicized September 26, 2019.

[†]The authors are with the School of Computer Science, Wuhan University, Wuhan, China.
[††]The authors are with the School of Computer Science, Central China Normal University, Wuhan, China.
[†††]The author is with the Discipline of ICT, School of Technology, Environments and Design, University of Tasmania, Hobart, TAS 7001, Australia.
a) E-mail: s.zhao@mail.ccnu.edu.cn
b) E-mail: rongpeng@whu.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2019EDP7108

bibliographic network. In the case of sparse data of the co-authorship network, HeteroRWR will choose the authors who were cited in the target authors' papers in the past as the candidates for recommendations. For the citation network contains a wealth of information, such as the relevance of the topics between papers and the academic status of authors, the recommended scholars will not only have high topic similarities with target users, but also have good centralities in the research communities. It should be emphasized that in order to ensure the convergence of the random walk process, we modify the original heterogeneous network structure by adding a virtual paper node and a small number of associated edges. We provide an insightful analysis and discuss the convegence of the proposed algorithm.

Extensive experiments have been performed on DBLP and CiteSeerX datasets. The experimental results show that HeteroRWR outperforms the state-of-the-art baseline methods such as Common Neighbors (CN), Adamic/Adar (AA), Random Walk with Restart (RWR) [4] and MVCWalker [6] models in terms of precicion and recall rate even for the incomplete citation dataset. Nevertheless, the proposed HeteroRWR approach is time efficient and is thus faciliated to be performed in large-scale networks.

To summarize, the main contributions of this paper are as follows:

- We apply the idea of heterogenous random walk with restart into the field of co-author recommendation application and explore the convergence property of the model. This has an advantage of fully using the citation network information and overcoming the difficulty of the data sparsity.
- A new edge weighting metric for citation networks has been proposed. The metric can guide the proposed algorithm to mine more potentially valuable co-authors with high topic-similarity and good community centralities.
- The proposed HeteroRWR algorithm improves recommendation performance compared to the state-of-the-art baseline methods in terms of precision and recall rate, and meanwhile, it has reasonable time efficiency.
- Unlike traditional approaches which require many citations, our method can effectively improve the recommendation performance particularly for the incomplete citation datasets.

2. Related Work

In this section, we will classify the related studies into two categories and then review them: friend recommendation and random walk in heterogeneous networks.

Friend Recommendation. At a high level, existing friend recommendation work can be classified into unsupervised methods and supervised methods.

Most unsupervised friend recommendation algorithms produce a ranked list in decreasing order of a "similarity" measure between two nodes, relative to the network topology [7], [8], [40], such as Common Neighbors, Jaccard Coefficient, Adamic/Adar, SimRank, RWR, $Katz_\beta$ etc [4]. Some other studies consider features about authors' affiliated information or geographic information [2], or research interests extracted from text information [9]. Unsupervised methods usually have low computational complexity, except for several global measures such as $Katz_\beta$, but it is difficult for one single similarity metric to model complex social network relationships, which results in inferior results. Some studies use weighted methods to integrate multiple features, but the performance is still unsatisfactory.

From the prospective of features used by algorithms, supervised friend recommendation approaches fall into three categories: explicit feature-based models, implicit feature-based models and hybrid models. Explicit feature-based methods extract explicit features related to the network topologies or users' information, and then use machine learning algorithms, such as SVM, logistic regression, Adaboost, neural network etc., to combine different features, to train binary classifiers, and to predict new links [10]–[12], [41]. Other supervised recommendation methods are mainly based on implicit features. Matrix factorization is the most frequently used technique for extracting the implicit features [13], [14], [42]. It projects users and items into a shared lower latent space and applies an inner product on the two latent real-valued feature vectors. It's the most popular one in traditional Collaborative Filtering (CF) approaches. In recent years, with the vigorous development of deep learning in speech recognition, natural language processing and other fields, some works use neural network models to extract implicit features for users and items [16] rather than matrix factorization. Other works combine deep learning technique with matrix factorization to extract the implicit features of users or items [17], [18]. For neural networks can extract the non-linear expression of the features and abstract them at a higher level, these deep learning-based recommendation models typically have better performance than pure matrix factorization or factorization machine models. It is natural for people to consider fusing the explicit features and the implicit features into one model to get better results, and such models are called hybrid models [20]–[22], [25].

The essence of recommendation problem is lying its sorting procedure. From the prospective of learning to rank [23], supervised friend recommendation approaches can be divided into pointwise learning, pairwise learning and listwise learning methods. Pointwise learning models transfer the sorting problem into a multi-classification problem or a regression problem [11], [12], and the disadvantage is that these models cannot deal with the high skewness of data very well. Pairwise learning models treat friend recommendation as a learning to rank problem based upon pairwise comparisons [14], [19], [24]–[26], [43]. Such kind of approaches can better overcome the problem of data skewness, but often suffers the problem of too large training dataset size and high time complexity. Listwise models aim to learn a ranking function by taking individual lists as in-

stances and minimizing a loss function defined on the predicted list and the ground truth list [22]. These models can directly optimize the ranking evaluation measures such as MAP and NDCG, however, for large data sets with binary classification labels, these methods are more complex to be performed.

Random Walk in Heterogeneous Networks. Heterogeneous information network (HIN) is a newly emerging research direction. It can model different objects and their rich relations in RSs, in which objects are of different types and links among objects represent different relations. HIN-based recommendation model can better overcome the data sparsity problem because of the ability to integrate abundant information into the model [19], [27]–[29].

RWR is an excellent global similarity-based unsupervised model for link recommendation. Many existing research works use RWR-based models to model user relationship strength [6], [30]–[33]. However, the basic RWR defined in homogeneous networks limits its performance, as mentioned. Some works extend RWR to heterogeneous networks, such as gene-disease networks [5], [34], and bibliographic networks [35], etc. Random walk in HIN enlarges the extension of RWR and enhances its applicability.

However, although there have been some studies on random walk in HIN, to our best knowledge, the application of co-author recommendation in heterogeneous bibliographic networks has not been explored yet, and the convergence property of the HIN-based RWR models has not been thoroughly analyzed. Therefore, we propose a framework for multiple random walks in a heterogeneous bibliographic network to address the sparsity problem in the co-author recommendation task.

3. Prelimiliary

This section begins with a description of the data model followed by a definition of the research problem. Since there are many symbols used in the article, we list the main symbols in Table 1.

3.1 Data Model

A co-authorship network can be modeled by an undirected graph $G_a = (V^a, E^{aa})$ from the data set of academic publications. V^a is the set of authors, while each link in E^{aa} represents two authors have co-authored at least one paper. The number of authors is $n = |V^a|$, and the set of authors is $V^a = \{a_1, a_2, \cdots, a_n\}$.

A citation network can be also modeled as a directed graph $G_p = (V^p, E^{pp})$ from the citation dataset. V^p is the set of papers, while each directed edge in E^{pp} denotes a citation relationship between papers. The number of papers is $m = |V^p|$, and the set of papers is $V^p = \{p_1, p_2, \cdots, p_m\}$.

An authorship network can be modeled as a bipartite graph $G_{ap} = (V^a \cup V^p, E^{ap})$. Edges in E^{ap} connect each paper with all of its authors.

Inspired by the existing studies [35], [36], we consider that the promising way for co-author recommendation is to concatenate G_a, G_p and G_{ap} as a whole to form a heterogenous network G. In this regard, it enables us to capture rich information across G_a, G_p and G_{ap} for friend recommendation.

Table 1 Symbolic description

Symbol	Definition
$G = (V, E)$	heterogenous bibliographic network
$\|V\|$	number of nodes in G
G_a	co-authorship network
G_p	citation network
G_{ap}	authorship network
A	adjacency matrix for G
A^{aa}	adjacency matrix for G_a
A^{pp}	adjacency matrix for G_p
A^{ap}	adjacency matrix for G_{ap}
A^{pa}	transpose of A^{ap}
α	$1-\alpha$ is the restart probability
λ	probability assignment of random walk in G_a
β	probability assignment of random walk in G_p
s	target user
$\vec{\mathcal{R}_{\mathcal{A}}}$	probability distribution vector for authors
$\vec{\mathcal{R}_{\mathcal{P}}}$	probability distribution vector for papers
$\vec{Q}$	restart vector
M_{AA}	transition matrix for A^{aa}
M_{PP}	transition matrix for A^{pp}
M_{AP}	transition matrix for A^{ap}
M_{PA}	transition matrix for A^{pa}
e_{ij}	edge formed by node i and node j
w_{ij}	edge weight for e_{ij}
$p(i \to j)$	probability of moving from node i to j in one step
p_{ij}	1-step transition probability from state j to state i
$p_{ij}^{(x)}$	x-step transition probability from state j to state i
M	transition matrix for G
M'	the revised new transition matrix for G
$\tilde{M}$	transition matrix for HeteroRWR
$\tilde{M}'$	the revised transition matrix for HeteroRWR
n	number of authors in G_a
m	number of papers in original G_p
$gcd\{\ldots\}$	greatest common divisor for a set
k	number of recommending authors
$\tilde{t}$	number of iterations of convergence for HeteroRWR

Definition 1. **Heterogenous Network**: $G = (V, E)$. Where V is the vertex set and $E = V \times V$ is the edge set. $V = V^a \cup V^p$, $E = E^{aa} \cup E^{ap} \cup E^{pp}$. i.e. the heterogeneous network G consists of three networks: an undirected co-authorship network $G_a = (V^a, E^{aa})$, a bipartite authorship network $G_{ap} = (V^a \cup V^p, E^{ap})$, and a directed citation network $G_p = (V^p, E^{pp})$.

Figure 1 is an example. Four authors A, B, C, D collaborate to publish papers p_1, p_2, p_3, while p_1 cites p_2, p_3, and p_2 cites p_3. The three different types of relationships between authors and papers generate three different networks G_a, G_p and G_{ap}, as depicted in Fig. 1.

3.2 Adjacency Matrix

For the purpose of storing G, we use $|V| \times |V|$ adjacency matrix A to represent G:

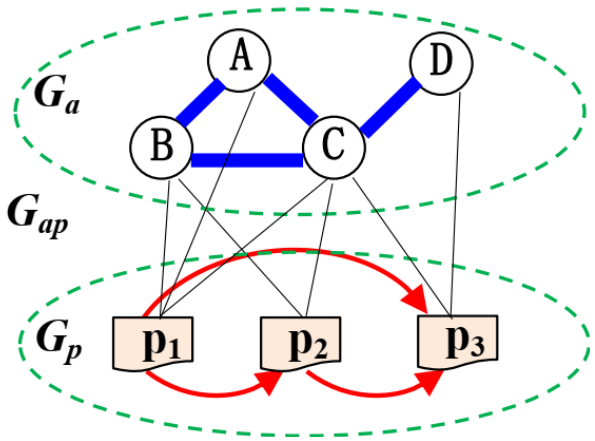

Fig. 1 Example of a heterogeneous network

$$A = \left[\begin{array}{c|c} A^{aa} & A^{ap} \\ \hline A^{pa} & A^{pp} \end{array}\right] \tag{1}$$

Where A^{aa}, A^{ap}, A^{pa} and A^{pp} each denotes a different relation between authors (a) and papers (p). That is, A^{aa} denotes the co-author relationships in G_a, A^{pp} denotes the citation relationships in G_p, and A^{ap} denotes the authorship relationships in G_{ap}. A_{pa} is the transpose of A_{ap}.

For the example in Fig. 1, if G_a, G_{ap} and G_p are set to be unweighted, the generated adjacency matrix A is:

$$A = \begin{array}{c} \\ A \\ B \\ C \\ D \\ \hline p_1 \\ p_2 \\ p_3 \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \left[\begin{array}{cccc|ccc} 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{array}\right] \\ \begin{array}{ccc} p_1 & p_2 & p_3 \end{array} \end{array} \tag{2}$$

The weight setting for G counts a great deal for the proposed algorithm. We give a detailed description of weight setting in Sect. 4.

3.3 Problem Definition

Definition 2. **Top-k co-author recommendation problem**: Given a snapshot of heterogeneous network G which consists of a co-authorship network G_a, an authorship network G_{ap}, and a citation network G_p between time $[t_0, t_1]$, for each target user s, recommending the k most potentially valuable co-authors during the interval from t_1 to a given future time t_2.

4. Proposed Model

In this section, we describe and explain our proposed top-k co-author recommendation algorithm in detail.

4.1 HeteroRWR Framework

As we know, citation networks contain a wealth of information. Citation relationships imply topic similarities between papers. Authors whose papers are heavily cited are likely to have relatively high centralities in research communities. Therefore, in the case of an extremely sparse co-authorship network, if random walk in the co-authorship network can be extended to the citation network, more partners with potential cooperative value can be found. Based on this intuition, we design a novel model named HeteroRWR for co-author recommendation in the heterogeneous bibliographic network G. The proposed model is briefly sketched in Fig. 2.

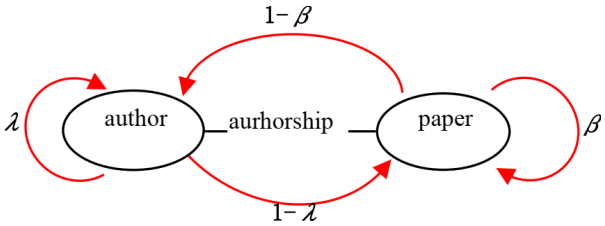

Fig. 2 HeteroRWR framework

Different from the basic RWR which only defines random walk in homogeneous networks, there are three different types of random walks in HeteroRWR: One on G_a, one on G_p and the other on G_{ap}. Specifically, for recommending co-authors to the target user s, a random walker starts from s. At each step of the random walk, we define:

- when the random walker moves from one node to another, it chooses to move to the neighbor nodes of the current node with probability α, while return to s with probability $1 - \alpha$.
- If the random walker swims to an author node, at the next step, it will move to the homogeneous nodes (author nodes) with probability $\alpha \cdot \lambda$, and to the heterogeneous nodes (paper nodes) with probability $\alpha \cdot (1 - \lambda)$.
- If the random walker swims to a paper node, at the next step, it will move to the homogeneous nodes (paper nodes) with probability $\alpha \cdot \beta$, and to the heterogeneous nodes (author nodes) with probability $\alpha \cdot (1 - \beta)$.

It can be seen that at each step of random walk, there are two different scenarios: homogeneous random walk and heterogeneous random walk. Homogeneous random walk includes the random walks within G_a or G_p, while heterogenous random walk means the random walk on G_{ap}. The two homogeneous random walks on G_a and G_p are coupled by the heterogeneous random walk on G_{ap}.

At each step, the random walker returns to the initial node s with the probability of $\alpha \in (0, 1)$. Parameter α controls the restart probability. Parameters λ and β regulate the coupling and their values reflect the extent to which we want to use the citation network information for co-author recommendation, and they also range from 0 to 1. In order to maintain symmetry, we usually set $\lambda + \beta = 1$. In this case, the value of a single parameter λ (or β) can reflect the probability of random walk assigned to G_p in the random walk process. But it needs to be stated that, $\lambda + \beta = 1$ does not necessarily have to be satisfied.

4.2 Formulation

As can be seen from Sect. 4.1, there are two coupled random walk processes in HeteroRWR: one is on G_a and the other is on G_p.

Let $\vec{\mathcal{R}}_{\mathcal{A}} \in \mathbb{R}^n$ denote the probability distribution column vector for author nodes, and $\vec{\mathcal{R}}_{\mathcal{P}} \in \mathbb{R}^m$ denote the probability distribution column vector for paper nodes. Define $\vec{\mathcal{R}} = \begin{pmatrix} \vec{\mathcal{R}}_{\mathcal{A}} \\ \vec{\mathcal{R}}_{\mathcal{P}} \end{pmatrix}$, then $\vec{\mathcal{R}} \in \mathbb{R}^{(m+n)}$ is a probability distribution vector for all nodes in G. $\|\vec{\mathcal{R}}_{\mathcal{A}}\|_1 + \|\vec{\mathcal{R}}_{\mathcal{P}}\|_1 = 1$ should be satisfied during the whole random walk process.

Based on the HeteroRWR framework defined above, the transition iterative formulas for the two mutual coupling random walks can be written as:

$$\begin{aligned} \vec{\mathcal{R}}_{\mathcal{A}}^{(t+1)} &= \alpha \cdot [\lambda M_{AA} \vec{\mathcal{R}}_{\mathcal{A}}^{(t)} + (1-\beta) M_{AP} \vec{\mathcal{R}}_{\mathcal{P}}^{(t)}] \\ &\quad + (1-\alpha) \cdot \vec{Q}_{\mathcal{A}} \qquad (3) \\ \vec{\mathcal{R}}_{\mathcal{P}}^{(t+1)} &= \alpha \cdot [(1-\lambda) M_{PA} \vec{\mathcal{R}}_{\mathcal{A}}^{(t)} + \beta M_{AP} \vec{\mathcal{R}}_{\mathcal{P}}^{(t)}] \\ &\quad + (1-\alpha) \cdot \vec{Q}_{\mathcal{P}} \qquad (4) \end{aligned}$$

where M_{AA}, M_{PA}, M_{AP} and M_{PP} each represents the probability transition matrix whose ij^{th} element is the probability of moving from node j to node i in G at one step, i.e. $p(j \to i)$. We give the detailed computation in Sect. 4.6. $\vec{Q} = \begin{pmatrix} \vec{Q}_{\mathcal{A}} \\ \vec{Q}_{\mathcal{P}} \end{pmatrix}$ is a restart vector, in which the corresponding value of s is 1, while all the other elements are zeros, i.e. $\vec{Q} = (0, \ldots 0, 1, 0, \ldots 0) \in \mathbb{R}^{(n+m)}$. The initial value of $\vec{\mathcal{R}}$ is equal to $\vec{Q}$.

Next, we merge Eq. (3) and Eq. (4) into one iterative formula:

$$\begin{aligned} \vec{\mathcal{R}}^{(t+1)} &= \alpha \cdot \begin{bmatrix} \lambda M_{AA} & (1-\beta) M_{AP} \\ (1-\lambda) M_{PA} & \beta M_{PP} \end{bmatrix} \vec{\mathcal{R}}^{(t)} \\ &\quad + (1-\alpha) \cdot \vec{Q} \qquad (5) \end{aligned}$$

let $e^T \in \mathbb{R}^{(m+n)}$ denote the all-ones row vector, i.e. $e^T = (1, 1, \ldots, 1)$, then $e^T \cdot \vec{\mathcal{R}}^{(t)} = 1$ holds due to the fact that the 1-norm for $\vec{\mathcal{R}}^{(t)}$ is always equal to 1. Therefore, we get:

$$\begin{aligned} \vec{\mathcal{R}}^{(t+1)} &= \alpha \begin{bmatrix} \lambda M_{AA} & (1-\beta) M_{AP} \\ (1-\lambda) M_{PA} & \beta M_{PP} \end{bmatrix} \vec{\mathcal{R}}^{(t)} \\ &\quad + (1-\alpha) \vec{Q} \cdot e^T \vec{\mathcal{R}}^{(t)} \\ &= \left\{ \alpha \begin{bmatrix} \lambda M_{AA} & (1-\beta) M_{AP} \\ (1-\lambda) M_{PA} & \beta M_{PP} \end{bmatrix} + (1-\alpha) \vec{Q} e^T \right\} \\ &\quad \cdot \vec{\mathcal{R}}^{(t)} \qquad (6) \end{aligned}$$

Define

$$\tilde{M} = \alpha \begin{bmatrix} \lambda M_{AA} & (1-\beta) M_{AP} \\ (1-\lambda) M_{PA} & \beta M_{PP} \end{bmatrix} + (1-\alpha) \vec{Q} e^T \qquad (7)$$

then $\tilde{M}$ is the transition matrix for the HeteroRWR random walk.

The stationary distribuion $\vec{\mathcal{R}}^{\bar{t}}$, solution of the $\vec{\mathcal{R}} = \tilde{M} * \vec{\mathcal{R}}$ represents the probability for the random walker to be located at a specific node after a sufficiant amount of time. However, Markov chains do not always converge. Can the transition matrix $\tilde{M}$ satisfy the convergence condition of Markov chains? If not, what changes can be made to meet the required conditions? Next, we will present the convergence analysis of the HeteroRWR model.

4.3 Convergence Analysis

Random walk can be regarded as a special case of Markov chains. A Markov chain converges to a stationary distribution if the transition matrix $P = (p_{ij})$ satisfy the following conditions [37], [38], [45]:

- **stochastic**. A matrix P is stochastic means: for all i, j, $p_{ij} \geq 0$, and $\sum_i p_{ij} = 1$.
- **irreducible**. A Markov chain is said to be irreducible if: for all i, j, there exists a positive integer x such that $p_{ij}^{(x)} > 0$. That is, all states communicate with each other, as one can always go from any state to any other state.
- **aperiodic**. A Markov chain is aperiodic when the number of steps required to move between two states is not required to be multiple of some integer. In other words, the chain is not forced into some cycle of fixed length between certain states.

Before analyzing the convergence, we first give the assumptions of this paper.

Assumption 1. G_a is a strongly connected network.

Many complex networks may not necessarily be strongly connected. But most of them contain a large connection component that contains most of the nodes in the network. We assume that the co-authorship network G_a in this paper is the largest connected component of the whole co-authorship network. Next, we explore the convergence property of the transition matrix $\tilde{M}$.

Firstly, we analyze the stochasticness of $\tilde{M}$. We begin with the stochasticness of four sub transition matrices $M_{AA}, M_{AP}, M_{PA}, M_{PP}$. For convenience, we use M to union them as a whole:

$$M = \left[\begin{array}{c|c} M_{AA} & M_{AP} \\ \hline M_{PA} & M_{PP} \end{array} \right] \qquad (8)$$

For a strongly connected network G_a and a bipartite graph G_{ap}, each node in G_a and G_{ap} has at least one outgoing link within each network, which ensures that the column sum for each transition matrix M_{AA}, M_{AP}, M_{PA} equals 1, i.e. M_{AA}, M_{AP} and M_{PA} are stochastic.

However, that's not the case for M_{PP}. In M_{PP}, there are always some columns with all zeros. For example, we record the earliest published papers in the time interval $[t_0, t_1]$ as $\mathcal{X}$ set. If a paper in $\mathcal{X}$ contains any reference information, the related cited papers are certainly not published during $[t_0, t_1]$, but earlier. Therefore, each paper in $\mathcal{X}$ has no outgoing links in G_p, which result in the corresponding

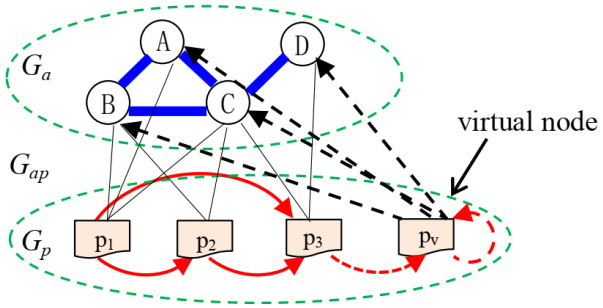

Fig. 3 Example of a modified heterogeneous network

column values in M_{PP} all being zeros.

Take Fig. 1 as an example. Assume the publication years of papers p_1, p_2 and p_3 are 2011, 2009 and 2000, respectively. If G is set to be unweighted, the original transition matrix M is as following:

$$M = \left[\begin{array}{c|c} M_{AA} & M_{AP} \\ \hline M_{PA} & M_{PP} \end{array}\right]$$

$$= \begin{array}{c} \\ A \\ B \\ C \\ D \\ \hline p_1 \\ p_2 \\ p_3 \end{array} \begin{array}{c} \begin{array}{ccccccc} A & B & C & D & p_1 & p_2 & p_3 \end{array} \\ \left[\begin{array}{cccc|ccc} 0 & 1/2 & 1/3 & 0 & 1/3 & 0 & 0 \\ 1/2 & 0 & 1/3 & 0 & 1/3 & 1/2 & 0 \\ 1/2 & 1/2 & 0 & 1 & 1/3 & 1/2 & 1/2 \\ 0 & 0 & 1/3 & 0 & 0 & 0 & 1/2 \\ \hline 1 & 1/2 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 1/3 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/3 & 1 & 1/2 & 1 & 0 \end{array}\right] \end{array} \tag{9}$$

For p_3 has no outgoing links in G_p, the column vector for p_3 in M_{PP} is a zero vector, which causes M_{PP} to violate the stochastic property.

We use a trick here. An addtional virtual paper node p_v is added to G_p to ensure that when the random walker swims to a paper node without outgoing links in G_p, it chooses to move to each author node with roughly equal probability. Specifically, each paper node without outgoing links generates a directed link to the virtual node p_v, and p_v generates directed links from itself to each author node. To ensure M_{PP} is stochastic, p_v also generates a directed link to itself. We present the revised heterogeneous network structure for Fig. 1 in Fig. 3.

The revised transition matrix M' for G therefore is:

$$M' = \left[\begin{array}{c|c} M_{AA} & M'_{AP} \\ \hline M'_{PA} & M'_{PP} \end{array}\right]$$

$$= \begin{array}{c} \\ A \\ B \\ C \\ D \\ \hline p_1 \\ p_2 \\ p_3 \\ p_v \end{array} \begin{array}{c} \begin{array}{cccccccc} A & B & C & D & p_1 & p_2 & p_3 & p_v \end{array} \\ \left[\begin{array}{cccc|cccc} 0 & 1/2 & 1/3 & 0 & 1/3 & 0 & 0 & 1/4 \\ 1/2 & 0 & 1/3 & 0 & 1/3 & 1/2 & 0 & 1/4 \\ 1/2 & 1/2 & 0 & 1 & 1/3 & 1/2 & 1/2 & 1/4 \\ 0 & 0 & 1/3 & 0 & 0 & 0 & 1/2 & 1/4 \\ \hline 1 & 1/2 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 1/3 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 1 & 1/2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right] \end{array} \tag{10}$$

As observed, the network structures of G_p and G_{ap} have been modified due to the addition of p_v. The transition matrices M_{PP}, M_{AP}, M_{PA}, and the restart vector $\vec{Q}_\mathcal{P}$ are also modified accordingly. Therefore, the revised transition matrix for HeteroRWR is as following:

$$\tilde{M}' = \alpha \begin{bmatrix} \lambda M_{AA} & (1-\beta)M'_{AP} \\ (1-\lambda)M'_{PA} & \beta M'_{PP} \end{bmatrix} + (1-\alpha)\vec{Q}' e^T \tag{11}$$

For four transition matrices $M_{AA} \in \mathbb{R}^{n\times n}$, $M'_{AP} \in \mathbb{R}^{n\times(m+1)}$, $M'_{PA} \in \mathbb{R}^{(m+1)\times n}$, and $M'_{PP} \in \mathbb{R}^{(m+1)\times(m+1)}$ are all stochastic, we can easily prove that $\sum_i \tilde{M}'_{ij} = 1$ for $j \in [1, (m+n+1)]$, i.e. the revised transition matrix $\tilde{M}'$ is stochastic.

Next, we analyze the irreducibility for the HeteroRWR Markov chain. Since G_a is a strongly connected network and the original G_{ap} is a bipartite graph, the original heterogeneous network G must be strongly connected regardless of whether G_p is connected. Even if we add an additional virtual node p_v to the network, it does not change the strong connectivity of G because p_v has both the outgoing links and ingoing links to the original network. That is to say, every node in G can reach any other node in G (all states communicate with each other), which proves the irreducibility of the HeteroRWR Markov chain.

Finally, we explore the aperiodicity of the Markov chain. What we need to prove is that for any state i in the Markov chain, the greatest common divisor of steps required to return to state i from itself is equal to 1 [44], that is to say, the period of each state in the Morkov chain is 1. From [37], we know that if state i and state j belong to the same class (state i and state j are mutually reachable), the periods of state i and state j are equal. Hence, for a strongly connected network (all nodes belong to the same class), as long as we prove that the period of one state in the Markov chain is 1, then the periods of all the other states are also 1.

This is obvious for the initial node s. Let $\left\{x : x \geq 1, p_{ii}^{(x)} > 0\right\}$ denote the set of all the steps that the Markov chain can return to state i when it starts from state i. As defined in [44], the period for state i can be denoted as: $d_i = gcd\left\{x : x \geq 1, p_{ii}^{(x)} > 0\right\}$. In the HeteroRWR random walk, node s has a link to itself, if the random walker moves to other nodes, it can return to s at each step with the probability of $1-\alpha$. That is, starting from the first step, the random walker can return to s at each step. Therefore, $d_s = gcd\{1, 2, 3, 4, 5 \ldots\} = 1$. In this way, we prove that $\tilde{M}'$ is aperiodic.

Based on the above anaysis, the revised transition matrix $\tilde{M}'$ in Eq. (11) satisfies all convergence conditions, and the Markov chain will reach a stable distribution in finite steps.

4.4 HeteroRWR Recommending Algorithm

It's known that a right eigenvector associated with the eigenvalue equals to 1 of the stochastic transition probability matrix of a Markov chain is its stationary probability vector.

Table 2 HeteroRWR Recommendation Algorithm

Algorithm 1. HeteroRWR Recommendation Algorithm.

Input: target author s; adjacency matrix A; parameters $\alpha, \beta, \lambda, k$;
Output: top-k recommending author list for author s.

1. **Initialization**
   compute transfer matrix $M_{AA}, M'_{AP}, M'_{PA}, M'_{PP}$
   $\vec{Q'} \leftarrow (0, \ldots 0, 1, 0, \ldots 0)$
   $\vec{\mathcal{R}}^{(0)} \leftarrow \vec{Q'}$
2. **Repeate**
3. $\quad \vec{\mathcal{R}_{\mathcal{A}}}^{(t+1)} = \alpha\lambda M_{AA}\vec{\mathcal{R}_{\mathcal{A}}}^{(t)} + \alpha(1-\beta)M'_{AP}\vec{\mathcal{R}_{\mathcal{P}}}^{(t)} + (1-\alpha)\vec{\mathcal{Q}_{\mathcal{A}}}$
4. $\quad \vec{\mathcal{R}_{\mathcal{P}}}^{(t+1)} = \alpha(1-\lambda)M'_{PA}\vec{\mathcal{R}_{\mathcal{A}}}^{(t)} + \alpha\beta M'_{PP}\vec{\mathcal{R}_{\mathcal{P}}}^{(t)} + (1-\alpha)\vec{\mathcal{Q}_{\mathcal{P}}}$
5. **Until Convergence**
6. sort $\mathcal{R}_{\mathcal{A}}{}^{\bar{t}}$
7. recommending list $\leftarrow top_k(\mathcal{R}_{\mathcal{A}}{}^{\bar{t}})$

The numerical solution of Markov chain can be solved with an algebraic method, but it always has a heavy workload for a large transition matrix. We choose to solve it with an iterative algorithm by successively multiplying some initial probability distribution vector by the matrix of transition probabilities [38]. Let $\vec{\mathcal{R}}^{\bar{t}} = \begin{pmatrix} \vec{\mathcal{R}_{\mathcal{A}}}^{\bar{t}} \\ \vec{\mathcal{R}_{\mathcal{P}}}^{\bar{t}} \end{pmatrix}$ be the stationary distribution, then $\vec{\mathcal{R}_{\mathcal{A}}}^{\bar{t}}$ can be used to describe the similarity between s and each author node, and it is the basis for us to recommend collaborators.

The detailed description of proposed HeteroRWR recommendation algorithm is shown in Table 2.

4.5 Time Complexity Analysis

Due to the addition of the virtual paper node p_v, the dimensions of the transition matrices M'_{AP}, M'_{PA}, M'_{PP} are changed to $n \times (m+1)$, $(m+1) \times n$, and $(m+1) \times (m+1)$, respectively. Therefore, in one iteration, the computational time complexity of the HeteroRWR algorithm is $O(n \cdot n + n \cdot (m+1) + (m+1) \cdot n + (m+1) \cdot (m+1)) = O((n+m+1)^2) = O((|V|+1)^2)$. Let $\bar{t}$ denote the number of iterations of convergence, then the total time complexity of recommending a top-k collaborator list for a single target user is $O(\bar{t}(|V|+1)^2)$.

It should be noted that, since the three networks in G are all extramely sparse, we use sparse matrices to store the four sub transition matrices. When performing matrix multiplication, only non-zero elements participate in the operation. Let C denote the maximum number of non-zero elements in each row vector of the transition matrix M', then the time complexity of each iteration is reduced to $O(C \cdot (m+n+1))$. The total time complexity of recommending a top-k collaborator list for a single user is reduced to $O(\bar{t}C(|V|+1))$.

4.6 Weight Setting

In this section, we introduce the weight setting for G. After that, we give the calculation method for the transition matrix.

4.6.1 Co-Authorship Network G_a

For G_a, the obvious defect of $0-1$ matrix is that the adjacent matrix cannot express the relationship strength between two authors. Therefore, we set the weight of edge $e_{ij} \in E^{aa}$ to be the number of papers co-authored by author a_i and author a_j. In this way, when performing a random walk algorithm, those neighbors who have co-authored more papers with the current author will have higher probabilities of being selected for random walk.

4.6.2 Authorship Network G_{ap}

For G_{ap}, the order of the authors largely reflects the contribution of different authors to the same paper, so it is very important information. If G_{ap} is set to be unweighted, all the authors of one paper will be treated equally. Such a defect cannot highlight the contribution difference between the authors.

How to measure the contribution of different authors in a multi-authored paper is a widely discussed issue in the reseach field of bibliometrics. In this paper, we use the fractional counting method [39] for determining individual credit of coauthors, i.e. if author a_i is one of the authors of paper p_j, the edge weight for edge $e_{ij} \in E^{ap}$ is:

$$w_{ij} = 1/r \tag{12}$$

where r is author a_i's order in the paper p_j's naming list.

The advantage of edge weighting is that, when performing HeteroRWR, if the random walker moves from a paper node to author nodes, the top-ranked authors will be selected with higher probabilities. And if the random walker moves from an author node to paper nodes, it is more likely to choose papers that the current author has made more contributions.

4.6.3 Citation Network G_p

If G_p is unweighted, recommendations tend to be made in favor of authors with a lot of ingoing links, i.e. the authors' papers have a large number of citations. Such scholars may have good centralities in the reseach communities, but they are likely to be academic bulls who published papers very earlier, and they may be old enough. On the other hand, papers that are widely cited but published for a long time may be relatively basic research. That's to say, when the publication time interval between two papers is too large, even though there is a citation relationship between them, it does not mean that the research interests between the related authors are very similar.

Therefore, we believe the publication time interval between the papers and the cited papers is a significant indicator. The closer the publication time of the two papers is, the more similar the research interests of the related authors are. Hence, we define a new edge weighting metric for G_p.

If paper p_i cites paper p_j, the weight for the edge $e_{ij} \in E^{pp}$ is:

$$w_{ij} = \frac{1}{\log(2 + t_i - t_j)} \tag{13}$$

In Eq. (13), we use log as the abbreviation of $\log_2$. t_i and t_j are the publishing years of papers p_i and p_j, respectively. Since $t_i \geq t_j$, i.e. the minimum value of $t_i - t_j$ is 0, we add a constant 2 to the denominator to ensure that the edge weighting metric is positive. It can be easily found that, the weight decays with the increase of the publication time interval between the papers, and it ranges from 0 to 1. This weight metric will guide the HeteroRWR algorithm to mine the potential valuable authors whose topic-related papers have been published in a more recent time.

4.6.4 Calculating the Transition Matrix

Based on the weight setting of G, we use the following formula to calculate the 1-step transition probability from node i to node j in each G_a, G_p and G_{ap}:

$$p(i \to j) = \frac{w_{ij}}{\sum_j w_{ij}} \tag{14}$$

Let the value of row i column j element of matrix M be $p(j \to i)$. Based on Eq. (14), we know that the ij^{th} entry in each transition matrix M_{AA}, M_{AP}, M_{PA}, and M_{PP} is the column normalization of the four adjacency matrices respectively.

It needs to be stated that, due to the addition of p_v, the final transition matrix of HeteroRWR is the revised version (in Eq. (11)).

4.7 Special Case Analysis

In HeteroRWR algorithm, parameters λ and β control the ratio of homogeneous random walk to heterogeneous random walk. Here, we analyze some extreme cases:

(1) $\lambda = 1, \beta = 0$: HeteroRWR random walk is performed only in G_a, G_p and G_{ap} do not work at all. HeteroRWR degenerates into a single homogeneous random walk in G_a, which is equivalent to the basic RWR.
(2) $\lambda = 0, \beta = 1$: HeteroRWR random walk is performed only in G_p, G_a and G_{ap} are completely useless. HeteroRWR degenerates into a single random walk algorithm in G_p. In this case, only the value of $\vec{\mathcal{R}_{\mathcal{P}}}^{(t)}$ is updated in each iteration, while the value of $\vec{\mathcal{R}_{\mathcal{A}}}^{(t)}$ keeps the intitial value and does not change. The HeteroRWR algorithm fails in this case.
(3) $\lambda = 1, \beta = 1$: the case is equivalent to the case (1).
(4) $\lambda = 0, \beta = 0$: HeteroRWR random walk occurs only in G_{ap}, while G_a and G_p do not work at all. HeteroRWR degenerates into a completely heterogeneous random walk, and no homogeneous random walk is allowed.

From the above analysis, we can see that when setting the values of parameters λ and β, we should try to avoid using such extreme values.

5. Experiment and Results

In this section, we first describe the experimental settings and then report the experimental results.

5.1 Datasets

The datasets we use includes DBLP[†] dataset and CiteSeerX dataset[††].

DBLP dataset is in XML format, and it contains metadata information about the computer-related academic publications, including author, title, booktile, url, crossref, etc. We use DBLP dataset to generate G_a and G_{ap}. The DBLP dataset we use is the last version of January 2018.

However, DBLP fails to provide citation information between publications, so we consider using CiteSeerX database which is a famous citation system. We crawl the citation information and reference information of the papers on CiteSeerX website to generate G_p.

5.2 Preprocessing

Firstly, we parse the original XML file of the DBLP dataset, which contains 2104606 conference papers and 1758872 journal papers ranging from year 1969 to 2018. Next, we select 23 well-known journals[†††] and 22 conferences[††††] papers related to the topics of data mining and machine learning and remove the papers published before 1990, 96174 papers are left.

Then, we divide the selected time interval into training interval and test interval with 2011 as the divding point. That is, the papers published in 1990.1 ∼ 2011.12 constitute the training set, and the papers published in 2012.1 ∼ 2018.1 constitute the test set. We count all the authors in the selected papers, and the number of papers they had published during the training and the test periods, respectively.

Next, we select $3 - core$ authors [11] from all authors, i.e. the authors who had published at least three papers during the training and the test intervals, respectively. The authors who fail to meet the requirement are excluded, 4146 authors are left. After that, based on the 4146 authors' publications in the training interval, we generate the coauthorship network of these authors. The network is not strong connected, and the largest connected component consists of 3609 authors. Since random walk in a disconnected

[†]http://dblp.uni-trier.de/xml/
[††]http://citeseerx.ist.psu.edu/index
[†††]AI, JMLR, DKE, IS, TNN, AMAI, ESWA, IDA, IJIS, KBS, NCA, PAA, WIAS, TODS, TOIS, TKDD, IEEE TKDE, DKE, DMKD, IS, KIS, IJIS, JIIS.
[††††]AAAI, CIKM, ECIR, EDBT, ICDE, ICDM, ICDT, ICML, IJCAI, JCDL, KDD, NIPS, PAKDD, PKDD, PODS, SDM, SIGIR, SIGMOD CONFERENCE, UAI, VLDB, WSDM, WWW.

Table 3 Statistics of training data

Networks	G_a	$\bar{G}_{ap}$	$\bar{G}_p$
Number of Nodes	3609	V^a:3609 V^p:19673	19673
Number of Edges	12971	36182	96326
Average Degree	7.19	V^a:10.03 V^p:1.84	9.29
Density of Networks	0.00199	0.000134	0.000249

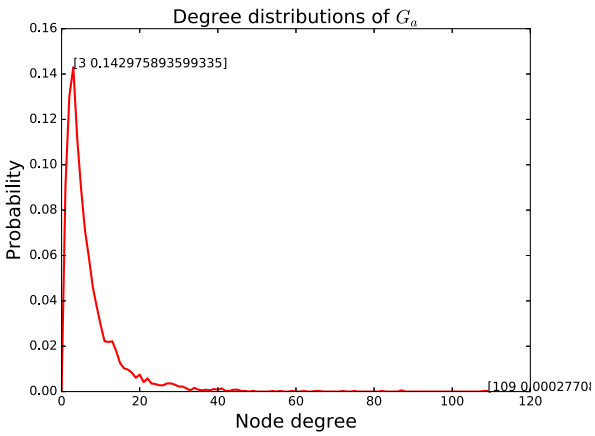

Fig. 4 Degree distributions of G_a

network does not guarantee convergence, we remove the authors who are not in the largest connected component. Finally, the total papers published by the 3609 authors are taken as the final data set, which contains 29877 publications in the training interval and 22293 publications in the test interval.

Using all the training papers as the seeds, we crawl the citation information from CiteSeerX website to generate G_p. However, real-world data are usually incomplete. The data we have crwaled from CiteSeerX website contains only the citation information of 19673 papers. Therefore, the citation network we ultimately use is $\bar{G}_p$, which consists of about 67 percent papers. We give the statistical information of the final training data set in Table 3 (in which $\bar{G}_p$, $\bar{G}_{ap}$ means incomplete dataset, and the virtual node p_v and the associated edges are excluded). As can be seen from the table, G_a, $\bar{G}_p$ and $\bar{G}_{ap}$ are all extremely sparse.

In Fig. 4, we give the degree distributions of G_a in the training period. From the chart presented, we can see that the degree distribution curve of the sampled authors reaches its maximum value of 3, followed by a sharp decline, and shows a trend of long tail distribution. Many complex networks follow the long tail distribution, which shows that the dataset we sampled has catholicity.

5.3 Experimental Setup

5.3.1 Experimental Environment

All the experiments are implemented in Python 2.7. The computer we used is Dell workstation, with 100G hard disk, 4G memory, Intel Core i7-4790CPU. The operating system is Ubuntu 16.04.

Table 4 Default values of parameters

Parameter	Range	Default Value
α	[0,1]	0.15
λ	[0,1]	0.6
β	[0,1]	0.4
k	5~50	5,10

Table 5 Confusion matrix

	Collaborated	Not Collaborated
Recommend	TP	FP
Not Recommend	FN	TN

5.3.2 Comparison Methods

We compare the performance of the HeteroRWR algorithm with several classical baseline methods, including Common Neighbors (CN), Adamic/Adar (AA) [4], the basic RWR and MVCWalker [6]. Since the basic RWR method to be compared is unweighted, there are two versions of the HeteroRWR algorithm: the unweighted version (HeteroRWR-U) and the weighted version (HeteroRWR).

Here, it should be noted that, all the baselines we compare use only G_a network information for recommending collaborators.

5.3.3 Setup and Evaluation

Based on G_a, $\bar{G}_p$ and $\bar{G}_{ap}$, we firstly generate the adjacency matrices A^{aa}, A^{pa}, A^{ap} and A^{pp}, then we use Eq. (14) to generate the four transition matrices M_{AA}, M'_{PA}, M'_{AP} and M'_{PP}. After that, the algorithm shown in Table 2 is used to recommend the top-k potentially valuable co-authors for target users. When evaluating the recommendation performance, 219 authors who had no co-authors in the test interval are excluded, and the remaining 3390 authors are the final test subjects.

The experimental parameters include k (the number of recommended authors), α, β and λ. When we explore the effect of one parameter, all the other parameters are set to default values. Table 4 shows the ranges and default values of the parameters, which are the relatively better values obtained by a large number of experiments.

We use precision@k and recall@k as measures of recommendation performance. The data set in the training interval is used to run various algorithms to generate top-k co-author recommendation lists for each target author, and the co-author relationships built in the test interval are used as the ground truth to evaluate the recommendation performance of the algorithms. For a target user s, we divide all users into four sets (as shown in Table 5):

- TP = {users who are recommended to s and had collaborated with s in the test interval};
- FP = {users who are recommended to s but hadn't collaborate with s in the test interval};

Table 6 Performance comparison

Algorithm	pre@5	rec@5	pre@10	rec@10
CN	0.201	0.184	0.142	0.251
AA	0.230	0.212	0.157	0.277
MVC	0.288	0.303	0.180	0.350
RWR	0.308	0.328	0.204	0.400
HeteroRWR-U	0.317	0.339	0.207	0.405
HeteroRWR	0.322*	0.344*	0.210*	0.408*

- FN = {users who are not recommended to s but had collaborated with s in the test interval};
- TN = {users who are not recommended to s and hadn't collaborate with s in the test interval}.

Then, the precision rate is defined as:

$$precision@k = \frac{|TP|}{|TP| + |FP|} \tag{15}$$

and the recall rate is:

$$recall@k = \frac{|TP|}{|TP| + |FN|} \tag{16}$$

In each experiment, the precision and recall rates are obtained by calculating the average of all the target users.

5.4 Experimental Results and Discussion

5.4.1 Comparing HeteroRWR with Baselines

We compare the proposed HeteroRWR algorithm with the baselines. The results under the default parameter values are shown in Table 6. As observed from Table 6, when the number of recommendation authors is 5 and 10, the weighted version HeteroRWR achieves the best performance, followed by the unweighted version HeteroRWR-U.

The result verifies that citation network information is helpful to improve the performance of the basic RWR algorithm. The main reason lies in that citing others' publications implies a certain degree of recognition of their work, so it is reasonable for us to use the referenced authors as the potential co-author candidates. Experimental results confirm this intuition. In addition, the weighted version outperforms the unweighted version, which indicates that reasonably setting the edge weight for the heterogeneous bibliographic network counts a great deal for the HeteroRWR algorithm.

5.4.2 Effect of Parameter α Setting

In this part, we explore the sensitivity of the parameter α. Related algorithms include the basic RWR, MVCWalker, and HeteroRWR. We present the experimental results in Fig. 5, Fig. 6, Fig. 7 and Fig. 8.

From the charts of Fig. 5 and Fig. 6, we observe that the proposed algorithm performs better when α is in the range of [0.05, 0.15]. When α is greater than 0.15, the precision gradually decreases as α increases. When α is greater than 0.8, the precision curve becomes steeper. Meanwhile, the

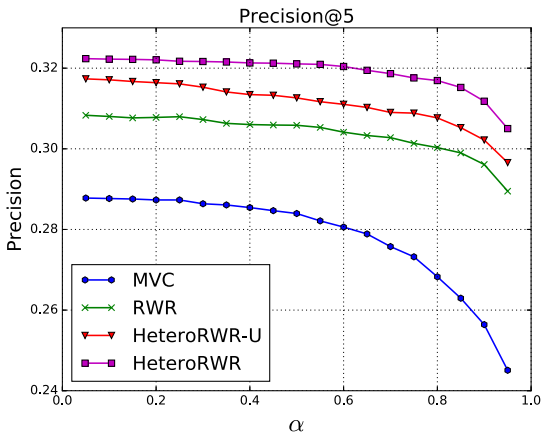

Fig. 5 The curve of the precision@5 with the change of α

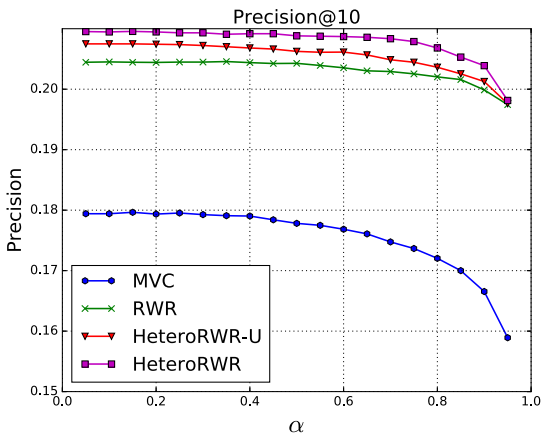

Fig. 6 The curve of precision@10 with the change of α

recall rates in Fig. 7 and Fig. 8 are similar trends.

We give the following analysis. As we know that $1 - \alpha$ is the restart probability. When α is larger, the restart probability of the random walk decreases, so the probability of walking to farther nodes increases. However, in the real world, social networks usually have the characteristic of small-world [40]. When the graph distance between two persons is too large, it is difficult for them to establish a real connection. Hence, larger α may lead to inferior performance, and we should try to avoid such a situation.

Based on the experimental results, we consider that 0.15 is the optimal value for the parameter α, so it is set as the default value in our experiments.

5.4.3 Effect of Parameters λ and β Setting

In this part, we explore the sensitivity of parameters λ and β. We always set $\lambda + \beta = 1$ in order to maintain the symmetry of the multiple random walks.

As can be seen from Fig. 9, Fig. 10, Fig. 11 and Fig. 12, when λ varies within the range of [0.05, 0.95], the shape of the precision and recall curves is like an arch. In other words, when λ is too large or too small, the precision and recall rate are relatively low, and when λ is medium, the

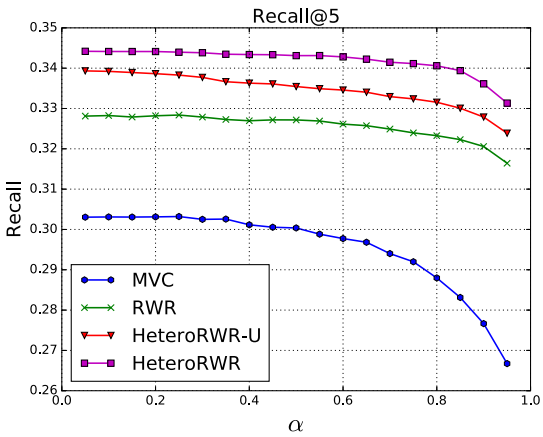

Fig. 7 The curve of recall@5 with the change of α

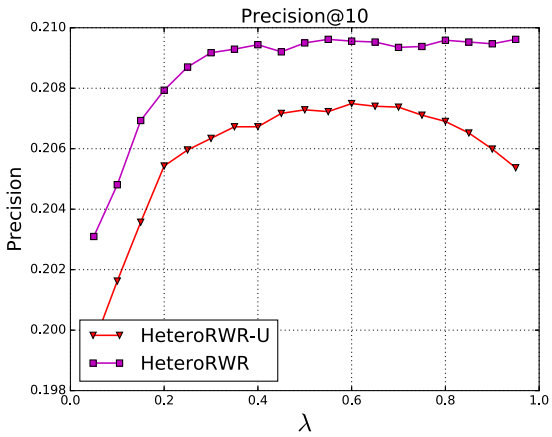

Fig. 10 The curve of precision@10 with the change of λ

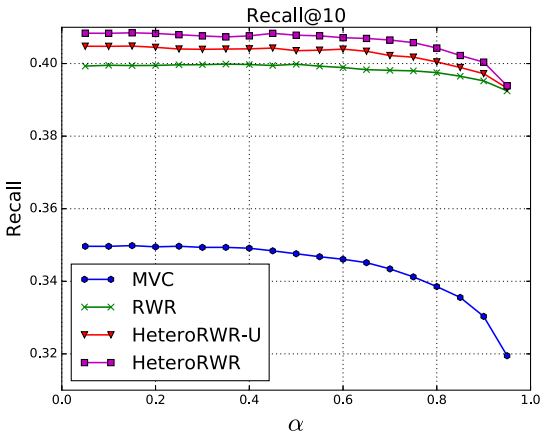

Fig. 8 The curve of recall@10 with the change of α

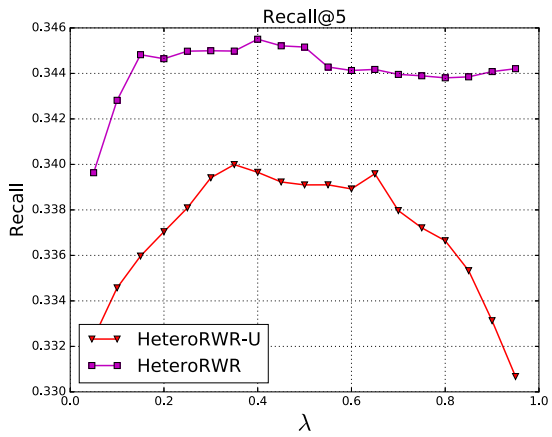

Fig. 11 The curve of recall@5 with the change of λ

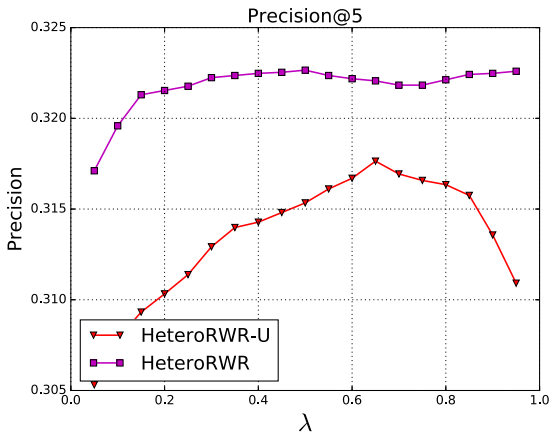

Fig. 9 The curve of the precision@5 with the change of λ

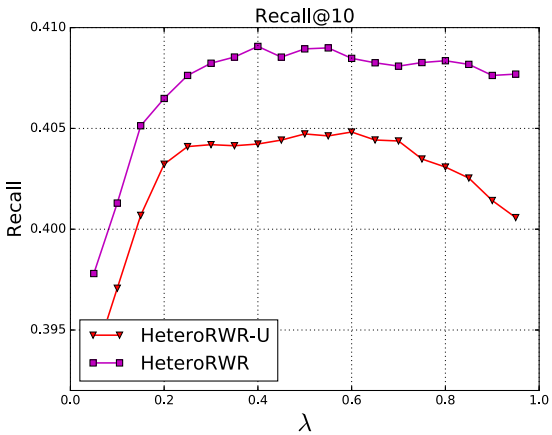

Fig. 12 The curve of recall@10 with the change of λ

precision and recall rate are relatively high. Specifically, there are slight differences between the two versions. Firstly, the weighted version always performs better than the unweighted version. Secondly, when λ is larger than 0.6, the performance of the unweighted version shows a significant downward trend, while the performance of the weighted version is relatively stable.

As a whole, it is more appropriate when λ falls in the range of [0.4,0.65]. We give the following explanations. When λ falls in [0.4, 0.65], it represents that the probabilities of random walks allocated in G_a and G_p are roughly equal. Therefore, HeteroRWR algorithm can make full use of the topology information of both the co-authorship network and the citation network. Otherwise, HeteroRWR random walk

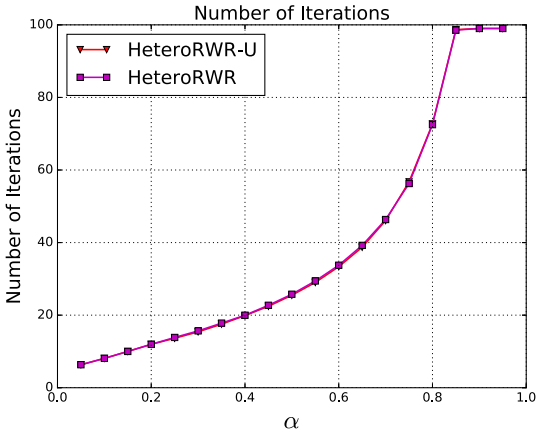

Fig. 13 Number of iterations with the change of α

will focus only on one network, which will result in poor overall effect. Hence, we set the default value of λ to 0.6 (β is 0.4) in the experiments.

5.4.4 Analysis of Convergence Iteration Number

In this section, we explore the convergence iteration number of the HeteroRWR algorithm. We set the maximum iteration number of HeteroRWR to be 100, and the termination condition $\|\vec{\mathcal{R}}_{\mathcal{A}}^{(t+1)} - \vec{\mathcal{R}}_{\mathcal{A}}^{(t)}\|_1 < \Delta = 10^{(-10)}$.

Based on a lot of experiments, we find that the convergence rate of HeteroRWR is not sensitive to λ and β, but very sensitive to the parameter α. As can be seen from Fig. 13, the number of iterations of the two versions of the HeteroRWR algorithm increases with the increase of α. What's more, the iteration numbers are almost exactly the same for the two different versions. We give the following explanations. When α becomes larger, the restart probability becomes smaller. As mentioned, it means that the random walks will go farther and reach more nodes in the network. Therefore, it needs more time for the whole random walk process to be stable.

In addition to investigating the number of convergence iterations, we also measure the actual execution time of the proposed algorithm (parameters are set to the default values). We find that it takes an average of 60 seconds for the proposed algorithm to generate a top-10 collaborator recommendation list for all 3390 test subjects. Therefore, on average, it takes 0.018 seconds to recommend a top-10 co-author list for each target user. In a real-world RS, this is a fully acceptable response time for users, which verifies the proposed method has high time efficiency on large datasets.

6. Conclusions and Future Work

In this paper, we have proposed a novel algorithm named HeteroRWR for attacking top-k co-author recommendation problem. To our best knowledge, HeteroRWR is the first co-author recommendation algorithm by multiple random walks in a heterogeneous bibliographic network which is composed of a co-authorship network and a citation network. We give the detailed analysis for the convergence property and time efficiency of the proposed HeteroRWR model. Numerous experiments on DBLP and CiteSeerX datasets have been conducted. The experimental results validate the efficiency and the effectiveness of the proposed algorithm. It should be noted that we only use an incomplete citation dataset. If more complete citation information can be obtained, the experimental results are expected to be better.

Our work can be further extended in several aspects. First, we expect that more features can be integrated into the model, such as users' geographic location information, and so on. Secondly, social networks are highly dynamic. A time-sensitive recommendation model is promising and worthy of being developed. Lastly, the experimental dataset used in the paper covers the major machine learning and data mining conferences and journals, we'd like to extend the experiments to consider papers in other fields.

Acknowledgments

This work was supported by the National Key Research and Development Plan of China (2017YFB0503700, 2016YFB0501801); the National Natural Science Foundation of China (61170026); the Fundamental Research Funds for the Central Universities (CCNU18QN019).

References

[1] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: a survey and new perspectives," ACM Comput. Surv., vol.52, no.1, pp.5:1–5:38, Feb. 2019.

[2] M.A. Brandão, M.M. Moro, G.R. Lopes, and J.P.M. Oliveria, "Using link semantics to recommend collaborations in academic social networks," Proc. 22nd International World Wide Web Conference (WWW'13), Companion Volume, pp.833–840, Rio de Janeiro, Brazil, May 2013.

[3] Y. Dong, J. Tang, S. Wu, J. Tian, N.V. Chawla, J. Rao, and H. Cao, "Link prediction and recommendation across heterogeneous social networks," Proc. IEEE 12th International Conf. Data Mining (ICDM'12), pp.181–190, Brussels, Belgium, Dec. 2012.

[4] V. Martínez, F. Berzal, and J.-C. Cubero, "A survey of link prediction in complex networks," ACM Comput. Surv., vol.49, no.4, p.69, Feb. 2017.

[5] A. Valdeolivas, L. Tichit, C. Navarro, S. Perrin, G. Odelin, N. Levy, P. Cau, E. Remy, and A. Baudot, "Random walk with restart on multiplex and heterogeneous biological networks," Bioinformatics, vol.35, no.3, pp.497–505, Feb. 2019.

[6] F. Xia, Z. Chen, W. Wang, J. Li, and L.T. Yang, "MVCWalker: random walk-based most valuable collaborators recommendation exploiting academic factors," IEEE Trans. Emerging Topics Comput., vol.2, no.3, pp.364–375, Sept. 2014.

[7] H.-H. Chen, L. Gou, X. Zhang, and C.L. Giles, "Collabseer: a search engine for collaboration discovery," Proc. 11th Joint International Conference on Digital Libraries (JCDL'11), pp.231–240, Ottawa, ON, Canada, June 2011.

[8] D. Guo, J. Xu, J. Zhang, M. Xu, Y. Cui, and X. He, "User relationship strength modeling for friend recommendation on Instagram," Neurocomputing, vol.239, pp.9–18, Feb. 2017.

[9] X. Kong, H. Jiang, T.M. Bekele, W. Wang, and Z. Xu, "Random

walk-based beneficial collaborators recommendation exploiting dynamic research interests and academic influence," Proc. WWW'17 Companion, pp.1371–1377, Perth, Australia, April 2017.
[10] Z. Yang, D. Li, R. Lin, Y. Tang, W. Li, and H. Liu, "An academic social network friend recommendation algorithm based on decision tree," Proc. 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, pp.1311–1316, Guangzhou, China, Oct. 2018.
[11] G.P. Gimenses, H. Gualdron, T.R. Raddo, and J.F. Rodrigues, "Supervised-learning link recommendation in the DBLP co-authoring network," Proc. 12th PerCom Workshops, pp.563–568, San Diego, CA, USA, March 2014.
[12] N. Benchettara, R. Kanawati, and C. Rouveirol, "A supervised machine learning link prediction approach for academic collaboration recommendation," Proc. 4th ACM Conference on Recommender Systems (RecSys'10), pp.253–256, Barcelona, Spain, Sept. 2010.
[13] D.D. Lee and H.S. Seung, "Algorithms for non-negative matrix factorization," Proc. NIPS 2000, pp.556–562, Denver, CO, USA, Nov. 2000.
[14] A.K. Menon and C. Elkan, "Link prediction via matrix factorization," Proc. 15th Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2011), pp.437–452, Athens, Greece, Sept. 2011.
[15] D. Ding, M. Zhang, S.-Y. Li, J. Tang, X. Chen, and Z.-H. Zhou, "BayDNN: friend recommendation with bayesian personalized ranking deep neural network," Proc. 26th ACM International Conference on Information and Knowledge Management (CIKM'17), pp.1479–1488, Singapore, Nov. 2017.
[16] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," Proc. 26th International Conference on World Wide Web (WWW'17), pp.173–182, Perth, Australia, April 2017.
[17] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," Proc. 10th ACM Conference on Recommender Systems (RecSys'16), pp.233–240, Boston, MA, USA, Sept. 2016.
[18] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," Proc. 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15), pp.1235–1244, Sydney, NSW, Australia, Aug. 2015.
[19] J. Verma, S. Gupta, D. Mukherjee, and T. Chakraborty, "Heterogeneous edge embeddings for friend recommendation," CoRR, vol.abs/1902.03124, Feb. 2019.
[20] C.C. Chen, S.-Y. Shih, and M. Lee, "Who should you follow? Combining learning to rank with social influence for informative friend recommendation," Decision Support Systems, vol.90, pp.33–45, June 2016.
[21] T. Chen, W. Zhang, Q. Lu, K. Chen, Z. Zheng, and Y. Yu, "SVDFeature: A toolkit for feature-based collaborative filtering," Journal of Machine Learning Research, vol.13, pp.3619–3622, Jan. 2012.
[22] D. Song, D.A. Meyer, and D. Tao, "Top-k Link Recommendation in Social Networks," Proc. 15th IEEE Internatinal Conference on Data Mining (ICDM'15), pp.389–398, Atlantic City, NJ, USA, Nov. 2015.
[23] H. Li, "A short introduction to learning to rank," IEICE Transactions on Information and Systems, vol.94-D, no.10, pp.1854–1862, Oct. 2011.
[24] D. Song, D.A. Meyer, and D. Tao, "Efficient latent link recommendation in signed networks," Proc. 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15), pp.1105–1114, Sydney, NSW, Australia, Aug. 2015.
[25] H. Ying, L. Chen, Y. Xiong, and J. Wu, "Collaborative deep ranking: a Hybrid pair-wise recommendation algorithm with implicit feedback," Proc. 20th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'16), vol.9652, pp.555–567, Auckland, New Zealand, April 2016.
[26] D. Rafailidis and F. Crestani, "Friend recommendation in location-based social networks via deep pairwise learning," Proc. 10th IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'18), pp.421–428, Barcelona, Spain, Aug. 2018.
[27] J. Yu, M. Gao, J. Li, H. Yin, and H. Liu, "Adaptive implicit friends identification over heterogeneous network for social recommendation," Proc. 27th ACM International Conference on Information and Knowledge Management (CIKM'18), pp.357–366, Torino, Italy, Oct. 2018.
[28] H. Chen, H. Yin, W. Wang, H. Wang, Q.V.H. Nguyen, and X. Li, "PME: Projected metric embedding on heterogeneous networks for link prediction," Proc. 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'18), pp.1177–1186, London, UK, Aug. 2018.
[29] H. Yin, L. Zou, Q.V.H. Nguyen, Z. Huang, and X. Zhou, "Joint event-partner recommendation in event- based social networks," Proc. 34th IEEE International Conference on Data Engineering (ICDE'18), pp.929–940, Paris, France, April 2018.
[30] Z. Yin, M. Gupta, T. Weninger, and J. Han, "A unified framework for link recommendation using random walks," Proc. International Conference on Advances in Social Networks Analysis and Mining (ASONAM'10), pp.152–159, Odense, Denmark, Aug. 2010.
[31] J. Gong, X. Gao, H. Cheng, J. Liu, Y. Song, M. Zhang, and Y. Zhao, "Integrating a weighted-average method into the random walk framework to generate individual friend recommendations," SCIENCE CHINA Information Sciences, vol.60, no.11, pp.110104:1–110104:22, Nov. 2017.
[32] L. Backstrom and J. Leskovec, "Supervised random walks: predicting and recommending links in social networks," Proc. 4th International Conference on Web Search and Data Mining (WSDM'11), pp.635–644, Hong Kong, China, Feb. 2011.
[33] J. Jung, W. Jin, L. Sael, and U. Kang, "Personalized ranking in signed networks using signed random walk with restart," Proc. 16th International Conference on Data Mining (ICDM'16), pp.973–978, Barcelona, Spain, Dec. 2016.
[34] Y. Li and J.C. Patra, "Genome-wide infering gene-phenotype relationship by walking on the heterogeneous network," Bioinformatics, vol.26, no.9, pp.1219–1224, March 2010.
[35] D. Zhou, S.A. Orshanskiy, H. Zha, and C.L. Giles, "Co-ranking authors and documents in a heterogenous network," Proc. 7th International Conference on Data Mining (ICDM'07), pp.739–744, Omaha, NE, USA, Oct. 2007.
[36] Q. Meng and P.J. Kennedy, "Discovering influential authors in heterogeneous academic networks by a co-ranking method," Proc. 22nd ACM international conference on Information & Knowledge Management (CIKM'13), pp.1029–1036, San Francisco, CA, USA, Oct. 2013.
[37] B. Zhang and H. Shang, Applied Stochastic Processes, China Renmin University Press, Beijing, 2016.
[38] A.N. Langville and C.D. Meyer, "deeper inside pagerank," Internet Mathematics, vol.1, no.3, pp.335–380, Jan. 2004.
[39] J. Kim and J. Diesner, "A network-based approach to coauthorship credit allocation," Scientometrics, vol.101, no.1, pp.587–602, Feb. 2014.
[40] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," Journal Of The American Society for Information Science and Technology, vol.58, no.7, pp.1019–1031, March 2007.
[41] H. Siyao and X. Yan, "Friend recommendation of microblog in classification framework: Using multiple social behavior features," International Conference on Behavioral, Economic, and Socio-Cultural Computing (BESC'14), pp.169–174, Shanghai, China, Oct. 30-Nov. 1, 2014.
[42] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," Proc. 14th ACM SIGKDD international conference on Knowledge discovery and data mining

(KDD'08), pp.426–434, Las Vegas, Nevada, USA, Aug. 2008.

[43] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personized ranking from implicit feedback," Proc. 25th Conference on Uncertainty in Artificial Intelligence (UAI'09), pp.452–461, Montreal, QC, Canada, June 2009.

[44] L.-C. Canon, M.E. Sayah, and P.-C. Heam, "A markov chain monte carlo approach to cost matrix generation for scheduling performance evaluation," CoRR, vol.abs/1803.08121, March 2018.

[45] B. Walsh, "Markov chain Monte Carlo and Gibbs sampling," Lecture Notes for EEB 581, University of Arizona, http://nitro.biosci.arizona.edu/courses/EEB581-2004/handouts/Gibbs.pdf, accessed 2004.

Sufen Zhao received her B.S. and M.S. degrees both in Computer Science from Central China Normal University in 2002 and 2005, respectively. She was a research assistant at Department of Electronic Engineering, City University of Hong Kong from 2006.5 to 2006.11. Currently, she is an instructor at School of Computer Science in Central China Normal University, PR China and a PhD candidate majoring in Computer Software and Theory in Wuhan University. Her research interests include social networks, data mining and machine learning.

Rong Peng received her PhD degree in Computer Science from Wuhan University in 2003. She is now working as a Professor in Wuhan University, China. Her research interests include software engineering, knowledge engineering, big data analysis and processing.

Meng Zhang received his PhD degree in Computer Science from Wuhan University in 2005. He is now working as an Associate Professor in Central China Normal University, PR China. He is interested in data mining and machine learning.

Liansheng Tan received his PhD degree in Mathematical Science from Loughborough University in the UK in 1999. He is now a Professor at Department of Computer Science in Central China Normal University. He was a research fellow in Research School of Information Sciences and Engineering, the Australian National University, Australia from 2006 till 2009, and a postdoctoral research fellow in 2001 in School of Information Technology and Engineering at University of Ottawa, Canada. He also held a number of visiting research positions at Loughborough University, University of Tsukuba, City University of Hong Kong and University of Melbourne. Dr. Liansheng Tan is currently the Editor-in-Chief of Journal of Computers, an Editor of International Journal of Computer Networks and Communications. He was an Editor of Dynamics of Continuous, Discrete & Impulsive Systems (Series B: Applications & Algorithms) (2006-2008), and an Editor of International Journal of Communication Systems. He has published 130 papers in international journals and conference proceedings including 22 in IEEE and ACM journals and two monographs with Elsevier and Taylor & Francis. His research interests include cloud computing, Internet of Things, computer networks, wireless sensor networks.