MVCWalker: A Random Walk Model for Recommending the Most Valuable Collaborators through Injection of Academic Factors

Feng Xia, *Senior Member, IEEE,* XXX, XXX, XXXX, and XXX

Abstract—In academia, scientific research achievements would be inconceivable without academic collaboration and cooperation among researchers. Previous studies have discovered that productive scholars tend to be more collaborative. However, it is often difficult and time-consuming for researchers to find the most valuable collaborators (MVCs) from a large volume of big scholarly data. In this work, we present MVCWalker, an innovative model, that stands on the shoulders of RWR (Random Walk with Restart) to recommend collaborators to scholars, based on big scholarly data. Three academic factors (i.e. coauthor order, latest collaboration time and times of collaboration) are exploited to define link importance in academic social networks for the sake of recommendation quality. We conducted extensive experiments on DBLP data set in order to compare MVCWalker to the basic model of RWR and the common neighbor-based model FOF in various aspects (e.g. the impact of critical parameters, the effect of academic factors and the performance over the entire DBLP dataset). Our experimental results show that incorporating the above factors into RWR can improve the precision, recall rate and coverage rate of academic collaboration recommendations.

Index Terms—Most valuable collaborator, Scientific recommendation, Big scholarly data, Random Walk model, Link importance.

1 Introduction

In an academic environment, collaboration among researchers has been increasingly popular and necessary. Previous studies confirm that there is a strong relationship between collaboration and productivity, and productive scholars tend to be more collaborative [1], [2]. Therefore, it would be instrumental for scholars to get acquainted with their most valuable collaborators (MVCs) [3]. Meanwhile, research on big scholarly data and academic social networks [4], [5] shows that, scholars in a collaborative context prefer to find valuable collaborators not yet known to them, or be in contact with distant researchers, in addition to staying in touch with their close colleagues. Considering the inherently social element, there has been difficulty in finding and recommending the MVCs on Academic Social Networks (ASN) [6].

Unfortunately, the huge size of big scholarly data makes it a significant challenge to find more valuable collaborators or totally new valuable collaborators. Common approaches to the problem are to proactively make personalized link predictions by predicting future connections, which is similar to what friend recommendation systems do in social networking sites (SNS). Specifically, a feature in SNS namely "People You May Know" has been proved meritorious in recommending users based on a common neighbor-based model FOF (friend of friends) [7], [8], which is popular in some social sites. Typical SNS such as Facebook, usually recommend friends that users already know offline [9]. However, recommending researchers in ASN based on big scholarly data is dissimilar from traditional recommendation of friends in social networks. In the academic context, traditional friend recommended systems have inherent weaknesses in satisfying scholars' requirements of discerning valuable collaborators. For instance, when making a decision about a collaborator, researchers often have to consider questions such as whether he/she has common research interests, if he/she is valuable in research from a collaboration perspective, and how to get connected with him/her. In order to satisfy scholars' special requirements, it becomes vital to develop special collaboration recommendation methods based on ASN.

A co-author network is an extraordinary social network due to its academic property of co-authorship, which can be modeled by a simple graph evolving from the author-paper binary graph as shown in Fig. 1. The links represent the relationships between researchers, and it should be noticed that the importance of the links are different. There are many factors which influence the measurement of relationships between researchers, e.g. latest collaboration time, times of collaboration and coauthor order. Consequently, as pointed out in [10], when recommending new co-authors in academic social networks, social interactions and its relational aspects should be taken into consideration.

In this paper, we propose a model named *A Random*

- *F. Xia, XXX, XXX, and XXX are with School of Software, Dalian University of Technology, Dalian 116620, China. Email: f.xia@ieee.org.*
- *L.T. Yang is with the School of Computer Science and Technology, Huazhong University of Science and Technology, China, and the Department of Computer Science, St. Francis Xavier University, Canada.*

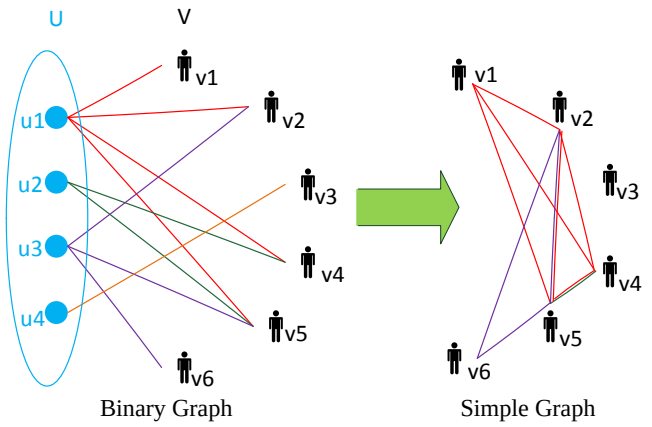


Fig. 1. Extraction from a Binary Graph to a Simple Graph. U is a list of papers, V is the list of authors. The figure shows three cases: (1) If a scholar has no collaborator, he/she is an isolated node, just like v_3; (2) If two authors coauthor a paper, there is a link between them, such as (v_1, v_2) and (v_6, v_5). (3) If two scholars coauthored multiple papers, the number of links between them increases, like (v_2, v_5) and (v_5, v_4).

Walk Model for Recommending the Most Valuable Collaborators through Injection of Academic Factors (MVCWalker), which enhances from our previous work [11]. MVCWalker is a kind of Random Walk (RW) model in which the rich information of both nodes and links are taken into account when being used for recommendation technology [12]. Random Walk with Restart (RWR) is a classic instance of RW model, which can provide a good relevance score between two nodes in a weighted graph, it has been successfully used in numerous settings, such as personalized PageRank and social recommendation [13]. Hence, we take advantage of RW model, by utilizing RWR and optimizing it through guidance for the random walker on the coauthor networks. Additionally, we define link importance based on some academic factors (i.e. the latest collaboration time point, the times of collaboration and the coauthor order). We further explore these three factors to measure co-authors' link importance. This can provide the random walk with more possibility to visit the MVCs on a weighted network and help improve the recommendation quality and accuracy.

In summary, we make the following contributions in this paper:

- To deal with scientific collaborator recommendation in the context of big scholarly data, we develop a model based on random walk with restart that learns how to bias a random walk on the network so that it can visit the potential collaborators with more probability than others.
- In order to improve the recommendation quality and accuracy, we propose and define the link importance of researchers in academic social networks by exploiting three specific factors including coauthor order, the latest collaboration time and collaboration times.
- We conduct extensive experiments on DBLP data set to evaluate the performance of the proposed method in various scenarios and compare it to RWR and FOF. Promising results are presented and analyzed.

The remainder of the paper is structured as follows. Section 2 briefly surveys the related work regarding of social recommender systems, features of coauthor networks, link prediction and RWR. We discuss the details of our proposed model in Section 3, which highlights our problem statement, workflow and computation of link importance. In Section 4 we discuss our experimental settings and analyze the results achieved. Finally, Section 5 concludes the paper.

2 Related Work

2.1 Academic Social Recommender Systems

Social networks have been studied for decades in an effort to comprehend the relationships between people and detect patterns in such interactions [14]. Recently much research work has been done on how to utilize social network information to improve recommender systems [15], [16]. For instance, Ma *et al.* [17] elaborated on how social networks information can benefit recommender systems and provided a general method for improving recommender systems by incorporating social network information. Perugini *et al.* [18] suggested that recommendation has an intrinsic social element and is essentially intended to connect people.

In this paper, we specifically aim to recommend MVCs in academic social networks (based on e.g. big scholarly data), which is different from recommending normal friends or items in academic context. Chin *et al.* [19] learned how Offline to Online interactions in a conference can help link people together, improve friend recommendations. Lee *et al.* [20] studied how well content-based, social and hybrid recommendation algorithms predicted coauthor relationship, and the result show that a hybrid algorithm combining expertise and social network information outperformed better. Pavlov and Ichise [21] proposed a method that extracts structural attributes from graph of past collaborations and uses them to train a set of predictors using supervised learning algorithms, these predictors can then be used to predict future links between existing nodes in the graph. Lopes *et al.* [10] considered researchers' publications area and the vector space model to make collaboration recommendation in academic social networks. A search engine for collaboration discovery named Collabseer was proposed in [3]. In academic social networks, to make MVC recommendations, there are many aspects we should consider, the influential of academic collaboration relationships. In this work, we utilize three specific academic factors into consideration.

2.2 Coauthor Networks

Scientific collaboration is a complex social phenomenon in research that has been systematically studied since the 1960 s. Furthermore, co-authorship is one of the most tangible and well documented forms of scientific collaboration [22]. As a kind of complex social networks, coauthor networks have been studied comprehensively. In these networks two scholars are normally connected if they have coauthored one or more papers together. By mapping the electronic database containing all relevant journals in mathematics and neuro-science for an 8-year period (1991-1998), Barabasi *et al.* [23] inferred that the dynamic and the structural mechanisms govern the evolution and topology of coauthor networks. They analyzed the basic network properties of academic social networks in terms of degree distribution, average separation, clustering coefficient, average degree and so on. Their results indicated that the scientific collaboration network is scale-free, and that the network evolution is governed by preferential attachment, affecting both internal and external links.

Newman [24] studied a variety of statistical properties of scientific collaboration networks, including the number of papers written by authors, numbers of authors per paper, numbers of collaborators that scientists have, existence and size of a giant component of connected scientists, and degree of clustering in the networks. At the same time Newman [24] found out that a number of differences are apparent among the fields studied. Researchers in different disciplines have different numbers of collaborators on average and the degrees of network clusters are also different.

Coauthor networks can be regarded as rich-information and weighted graphs that can fit many famous models. Liu *et al.* [25] conducted a coauthor network, for which he defined AuthorRank as an indicator of the impact of an individual author in the network. Their results show clear advantages of PageRank and AuthorRank over degree, closeness and betweenness centrality metrics. Based on these information of coauthor networks and theoretical support, we propose our model and design our metrics.

2.3 Link prediction

We can formalize academic collaboration recommendation as a link-prediction problem. Many approaches have been proposed for various link predictions [26]. For instance, David *et al.* [27] defined the link-prediction problem as follows: given a social network at time t, how to accurately predict the edges that will be added to the network in the future time t'. They developed approaches to link prediction based on measures for analyzing the ”proximity” of nodes in a network. Their approaches were applied to large social networks and the results suggested that fairly subtle measures for detecting node proximity can outperform direct measures.

Lichtenwalter *et al.* [28] examined important factors for link prediction in networks and provided a general framework for the prediction task. They cast link prediction as a problem in class imbalance. As a result, their consideration of some important factors leads to a general framework that outperformed unsupervised link prediction methods.

The work more closely related to ours is [29], which emphasizes recommending academic friends and considering the link semantics. The authors proposed two new metrics respectively representing the institutional affiliation and the geographic location of the researchers for recommending new collaborators. Our work differs from [29] in that we consider the details of coauthor relationship.

2.4 Academic Random Walk Model

Random walk model is often used in daily recommendation scenarios, etc. Mohsen *et al.* [30] proposed a random walk model that combines the trust-based and collaborative filtering approaches for recommendation. They took advantage of random walk to define and measure the confidence of a recommendation. Fouss *et al.* [31] presented a new perspective of characterizing the similarity among elements of a graph. Their method was based on a Markov-chain model involving random walk through the database. In this paper, our work stand on the shoulder of RWR, a famous random walk model, which provides a good way to measure how closely related two nodes are in a graph [32]. It has been successfully used in numerous areas including e.g. collaborative recommendation and link prediction. Konstas *et al.* [33] created a collaborative recommendation system that adopts the generic framework of RWR in order to provide with a more natural and efficient way to make recommendation. They conduct some comparative experiments between RWR and CF (collaborative filtering). Their experimental results show that the graph model benefits from the additional information embedded in social knowledge and outperforms the standard CF method. Backstrom *et al.* [12] proposed a supervised random walk based on RWR, to predict and recommend links in social networks.

These studies are quite close to our previous work ACRec [11]. The main goal of ACRec is to model both the attributes of authors and co-authorship at recommended MVCs so that further recommendation can be generated for researchers. By absorbing the advantages of these research mentioned above, we extend ACRec, inject academic factors and build MVCWaler to guide random walk. To be exact, we do more experimental work in this paper than in ACRec. We measure the influence of two more parameters, iteration times in random walking process and partition of training and testing data sets. At the same time, we do more optimizations over these two parameter settings. In addition, to further verify the superiority of our work, we compare MVCWalker

TABLE 1
List of Symbols

Symbol	Definition
MR	$N * 1$ ranking score vector of MVCs
p_i	Node i
$L(p)$	Number of all the neighbors of node p
$N(p)$	Set of nodes incident to node p
α	Damping coefficient: the probability of a walker walking to the next neighbor
N	Total number of nodes in a graph
$\mathbf{S}$	Transfer matrix
I	$N * 1$ starting vector for RWR
q	$N * 1$ starting vector for personalized RWR
t	Iteration times
$LIM(p_i, p_j)$	Link importance of p_i and p_j
$DCL(p_i, p_j)$	Distance in coauthor list of p_i and p_j
$k(t)$	A monotonically increasing function defined by coauthoring time
P	Precision of the recommendation result
R	Recall rate of the recommendation result
C	Coverage rate of the recommendation result

to both the basic RWR and the popular FOF model in this paper.

3 MVCWalker

In this section, we describe the details of MVCWalker. Following problem statement, we give an overview of MVCWalker. Furthermore, we explain how to compute the link importance by considering the academic factors one by one. The symbols used in this paper are listed in Table 1.

3.1 Problem Statement

In this paper, our goal is to find and recommend collaborators to researchers. According to section 2, we know that, the aforementioned researches exploit collaboration recommendation systems by considering varying information in big scholarly data, some factors will be helpful when making collaboration recommendation based on academic social networks, in addition random walk model can be adopt for its remarkable characteristic of integrating the rich information of both nodes and links. However, here comes the problem.

Problem 1: Academic collaboration recommendation is different from the traditional social recommendation. Scholars need collaborators who have common research interests and is valuable to coauthor and is connectable. To recommend MVCs for a scholar, what factors can be considered? How can a recommender algorithm (or model) be designed to achieve this goal?

Problem 2: As mentioned in Section 1, social interactions and its relational aspects can help to recommend collaborators. There are varying available academic social network features should be considered. However, what relations are featured in academic social networks? Which of these features are available?

3.2 Overview of MVCWalker

The MVCWalker collaborator recommendation model is inspired by the truth that scholars usually desire to cooperate with people who have high academic value. Such people normally have fruitful high-quality papers, which can generally be used to represent their academic achievements. Besides, as the RWR model has been proved to be competent for calculating the similarity of nodes in network, we use it as a basic model for the coauthor social network. Furthermore, we introduce edge attributes information into the network structure to bias the random walk such that it will more easily traverse to the positive nodes.

In MVCWalker recommendation, finding one's MVCs depends on the importance of other nodes to the target node. According to the importance, each recommended node has a rank score, which is determined by two factors, the number of nodes connected to this node and the importance of these nodes. It can be described as:

$$MR(p_i) = \frac{1-\alpha}{N} + \alpha \sum_{p_j \in M(p_i)} \frac{MR(p_j)}{L(p_j)} \tag{1}$$

where MR represents the rank score vector, and $MR(p_i)$ is the rank score of node p, which is the quantized importance of node p_i to the target node. $M(p_i)$ is the set of nodes incident to node p_i, with $L(p_j)$ being the number of all the neighbors of node p_j. α denotes the probability that the walker will continue to walk to the next neighbor. Above all, in MVCWalker model, the walker has the probability to randomly skip to any other nodes.

Equation (1) represents only the step to get the rank score of a node. With respect to each node in the whole graph, the personalized random walk process is defined by (2), which is an iterative process.

$$MR^{(t+1)} = \alpha \mathbf{S} MR^{(t)} + (1-\alpha)q \tag{2}$$

$\mathbf{S}$ is the transfer matrix, representing the probability for each node to skip to other nodes. $MR^{(t)}$ represents the rank score vector at step t, and q is the row vector, and its form is $(0, \ldots, 1, \ldots, 0)$. In fact, at the beginning, $MR^{(0)} = q$, and the rank score of target node is 1, while others' are 0.

Consider a single random walker that starts from node p_i. The walker iteratively transmits to its neighborhood with the probability $\alpha S_{i,j}$, which is proportional to their link importance. At each step, it has the probability of $(1-\alpha)q_i$ to return to node p_i.

The relevance score defined by MVCWalker has many good properties: as compared with those common neighbor models, it can capture the global structure of a graph; while compared with those traditional short distance models, it can capture the multi-facet relationship between two nodes [34].

Basic random walk models usually assume that the weights of edges are the same [13], and define the cells

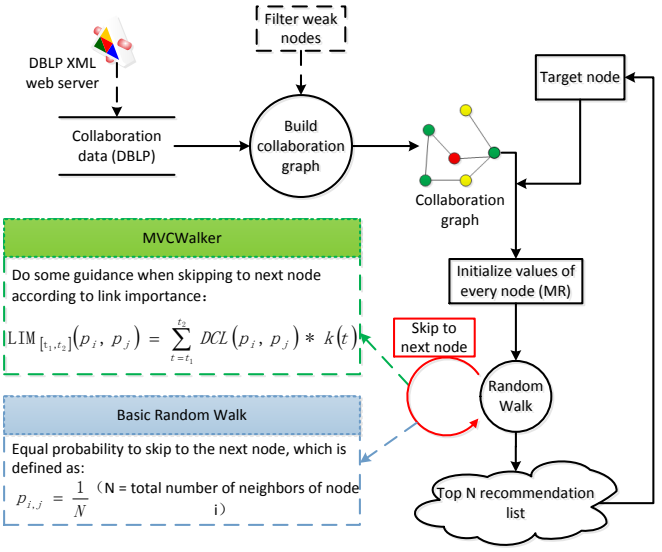


Fig. 2. Structure of MVCWalker.

of matrix $\mathbf{S}$ as $S_{i,j} = \frac{1}{L(p_j)}$. In contrast, here we define matrix $\mathbf{S}$ by link importance based on academic factors. We will introduce the link importance in section 3.3.

The detailed process of MVCWalker is described below and the corresponding pseudocode is shown in Algorithm 1. The structure of MVCWalker is illustrated in Fig. 2.

- The initial input data of MVCWalker is a set of several years' papers published by many scholars. To extract the coauthor network, we regard authors as nodes in the network. Before that, it is necessary to filter out the isolated nodes and weak nodes (with small degree). We define the graph G as the coauthor network, and P as the set of the nodes.
- If some authors have worked for one publication together, add edges among all those authors, which are defined as set E. In this step, we calculate the weight of each edge which is called the link importance in MVCWalker. Some attributes of collaboration should be taken into account (e.g. coauthor order, latest collaboration time point, and collaboration times), which will be described in the next section. We define the link importance between the nodes P_i and P_j as $w_{i,j}$.
- Before starting the random walk, we need to acquire the transfer matrix S as described in Algorithm 1. To be specific, we denote P_i as the current node while P_j as the next node. $\mathbf{S}$ is the set of probabilities for each P_i in G skipping to next node P_j. This can be described as $S_{i,j} = \frac{W_{i,j}}{\sum_{P_k \in N(P_i)} W_{i,k}}$, while $N(P_i)$ is the set of neighbors of P_i.
- MVCWalker starts with initializing the rank score vector $MR^{(0)}$ and the restart probability vector q as (0,...,1,...,0). The target node P_i is set to 1 while others are set to 0. MVCWalker iterates the traversal starting with node P_i until random walk stops walking and assigns each candidate node P_k a stable probability MR_k. Thus we get the rank score vector MR. Then sort the nodes in accordance to their corresponding rank scores.
- Finally, the TOP N nodes in the MR list are recommended to the target nodes. Of course, we can take out the nodes which have been in its coauthor list before recommending. That is the new co-authors recommendation.

Algorithm 1 MVCWalker(R, a, MaxIteration, MinDelta)

```
S ← ComputeTransferMatrix()
MR_0 ← R
Q ← R
for k ← 0 to MaxIteration − 1 do
    diff ← 0
    for i ← 0 to len(Q) − 1 do
        MR_{k_i} = α Σ_{j=0}^{len(Q)} S_{i,j} MR_j + (1 − α) Q_i
        diff ← diff + (MR_k − MR_{k−1})
    end for
    if diff < MinDelta then
        break
    end if
end for
Predictions ← predictions(MR)
return Predictions
```

In addition to the above, we present below, the details of how the link importance is computed by taking into account the three academic factors.

3.3 Link Importance

As mentioned in Fig. 1, when extracting a simple graph from a binary graph, considering the information about link features will be better, which indicate the importance of cooperation relationship between nodes. In reality, people might be more willing to choose certain nodes with high feature value for them. Therefore, a critical requirement of the MVC recommendation algorithm is to assign cooperation graph edge with related weight, to measure the cooperation relationship strengths between one user and his/her (potential) collaborators. We define the edge weight as *LIM* (Link Importance of MVCWalker).

Common sense depicts that, when choosing a collaborator, a scholar might prefer a researcher with whom he/she coauthored papers within the past year, rather than ten years ago. Consequently it seems that, closer relationships between authors are established through frequent collaboration, because coauthored papers can reflect the interest similarity of the authors. Even the coauthor order can reflect the similarity of the authors to some extent. This is why we choose these three factors, coauthor order, latest collaboration time point, and times of collaboration, to calculate LIM.

3.3.1 Coauthor Order

There is always a list of (co)authors for one paper. Normally, their contributions to the paper differ from each other, which we can measure generally by the author order. For example, the first two authors usually make more contributions than the rest authors. In such cases, the cooperation relationship between the first two authors is competently strong. That is, the coauthor order can reflect cooperation relationship strength. As a general rule, the contribution value is inversely proportional to the coauthor order, and the weight of relationship is contributed by the relevant two nodes. Therefore, we propose a measure of the link importance based on the coauthor order: *DCL* (distance in coauthor list).

Consider two nodes p_i, p_j in a coauthor list. Assume that $j > i$, and there are more than one author of a paper. For the sake of simplicity, we calculate $DCL(p_i, p_j)$ as follows.

$$DCL(p_i, p_j) = \begin{cases} \frac{1}{i} + \frac{1}{j} & j \leq 3 \\ \frac{1}{i} + \frac{2}{j} & j > 3, i \leq 3 \\ \frac{2}{i} + \frac{2}{j} & i > 3 \end{cases} \quad (3)$$

According to this definition, it is clear that the DCL value between the first and the second authors is 1.5, which is the maximum. The relationship between first two authors is the closest, while the relationship between the first author and the rest authors is relatively weak.

3.3.2 Latest Collaboration Time Point

In recent years, more and more social recommendation systems introduced time dependence models (e.g. [35]). Especially, academic social networks are obvious time-varying, where the links among scholars change over time. For instance, scholars might be more willing to collaborate with whom coauthored a paper last year, as compared to the co-authors of ten years ago. Hence, we measure the link dynamics using $LIM_t(p_i, p_j)$ (i.e. Link Importance):

$$LIM_t(p_i, p_j) = DCL(p_i, p_j) * k(t) \quad (4)$$

where $k(t)$ is a monotonically increasing function over time. We can measure the impact of different coauthoring time points by adjusting the parameter $k(t)$. Here, we define $k(t)$ as:

$$k(t) = \frac{t_i - t_0}{t_c - t_0} \quad (5)$$

where t_i is the published time of the paper i t-wo researchers coauthored (in year here), t_c is the current time (i.e. 2014 in this paper) and $t_0 = Published\ year\ of\ the\ first\ coauthored\ paper - 1$.

3.3.3 Times of Collaboration

In academic social networks, if two authors coauthor a paper, there will be a link between them. Furthermore, these two authors may collaborate many times. It is necessary to take the times of collaboration into account when measuring the link importance between two authors. Here we measure the impact of different times of coauthoring as follows:

$$\begin{aligned} LIM_{[t_1,t_2]}(p_i, p_j) &= \sum_{t=t_1}^{t_2} LIM_t(p_i, p_j) \\ &= \sum_{t=t_1}^{t_2} DCL(p_i, p_j) * k(t) \end{aligned} \quad (6)$$

According to above equation, we will calculate the sum of each link importance if there are n links between p_i and p_j during time period $t_1 \sim t_2$ $(t_0 < t_1 \leq t_2 < t_c)$.

TABLE 2
Seven papers by five researchers and their details

Paper ID	Authors (orderly)	Published year
1	Mark, Feng, Lisa, Wei	2009
2	Feng, Jing	2010
3	Mark, Lisa, Wei	2013
4	Jing, Wei, Feng	2012
5	Feng, Wei, Mark, Lisa	2013
6	Lisa, Feng, Mark	2014
7	Feng, Jing, Wei	2011

3.4 An example of MVCWalker

In order to better understand MVCWalker, we give an example.

As shown in Table 2, there are seven papers created by five researchers including paper id, orderly authors and published year. We can get the coauthor network (Fig. 3.) according to the coauthor activities, and then, calculate the LIM by making full use of these information. e.g. Wei and Jing have coauthored two papers (4 and 7) in 2011, 2012. The authors order are respectively (2, 1) and (3, 2). According to equation (6) and these data from their first cooperation time to now, We can compute that the LIM of Wei and Jing is $\frac{17}{12}$. Similar to the other authors, we can get the LIM values of all coauthor relationships, which have been displayed in Fig. 3.

Assuming that we make some academic cooperation recommendations for Jing. We know that she has two co-authors now, Feng and Wei. Regarding Jing as a random walker with initial MR value 1 (others 0), when deciding next jump, she has two equiprobable choices in basic RWR model, i.e. 0.5 probability to Feng and 0.5 probability to Wei. In MVCWalker model, we utilize a guidance procedure for this decision making process. She has probability of 0.602 (i.e. $\frac{77/36}{17/12+77/36}$) jumping to the node Feng, and 0.398 (i.e. $\frac{17/12}{17/12+77/36}$) to node Wei. The result of our experiment has proved the effectiveness of the guidance work. After limited times of iteration (19 times in this case), each of the nodes have have their own MR values which represent the importance to Jing. In this case, The MR values of Feng, Lisa, Wei, and Mark

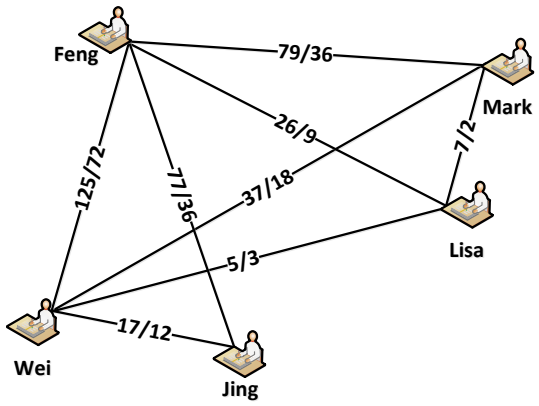


Fig. 3. Coauthor network with link importance over these five researchers

are $0.500, 0.391, 0.384, 0.376$ respectively. Hence we know that Feng is the most suitable partners in this circle, and Lisa can be recommended to Jing as a new cooperator (even though Jing has never worked with Lisa).

4 Evaluation and Analysis

We conducted extensive experiments using data from DBLP [36], a computer science bibliography website hosted at University of Trier. In this section, we describe the processing of DBLP data set, the evaluation metrics we employed and our experimental procedure for evaluating the performance of MVCWalker, as well as detailed analysis of the results.

To improve the effectiveness and efficiency of our MVCWalker model, We examined the influence of different parameters by conducting a series of experiments and optimization of some parameters. Within the same period, we embarked on different experiments to compare MVCWalker to the RWR and FOF models in terms of multiple metrics (i.e. precision, recall rate and coverage rate).

Similar to popular random walk models, the details and verification method of RWR is just like MVCWalker (described as section 4), except the definition of link importance. FOF is a common neighbor-based model. Its basis of recommending collaborators is the number of common neighbors. For two researchers, the more common neighbors they have, the more suitable to recommend to each other.

All experiments were performed on a 64-bit Linux-based operation system, Ubuntu 12.04 with a 4-duo and 3.2-GHz Intel CPU, 4-G Bytes memory. All the programs are implemented with Python.

4.1 Data Set

The data set we chose is from DBLP. The DBLP Computer Science Bibliography of the University of Trier grows from a very specialized small collection of bibliography information at the end of 1993 [36] and is expected to be the most comprehensive bibliography of the computer science field with support from the Anthology project [37]. After over 10 years' development, DBLP indexes more than 2.3 million articles on computer science and contains many links to home pages of computer scientists.

TABLE 3
Statistics of Data Set of Data Mining from DBLP

Statistics	Nodes	Edges	Average Degree
Number	59659	90282	1.513

Generally, each DBLP record contains these attributes, authors, title, pages, years, crossref, (in)proceedings or journals, etc. According to the methods provided by [36], to conduct our experiments, we can extract a subset of the entire data set using the required information. The reasons why we use a subset of the data set are as follows.

- It is possible that the subset of the researchers' publications be represented by a social network [29] and the analysis of published papers are of great significance to recommend most valuable collaborators.
- In a coauthor graph, there are some isolated authors who write publications without any cooperation. Thus, they nearly have no relationship with other scholars. Furthermore, we define these isolated authors as the weak nodes, since their degree values are 0. It is clear that the weak nodes have little impact on the random walk. Therefore, to measure the performance of MVCWalker better, we ignore the weak nodes whose degrees are less than 1.

The data sets we extracted are all in the field of data mining involving 34 journals and 49 conferences altogether. The statistics about the data sets are shown in Table 3, covering the number of nodes, edges and average degree.

4.2 Evaluation Metrics

We chose three popular metrics, precision, recall rate and coverage rate, to evaluate the performance of MVCWalker [38], [39]. Usually, the output of a recommendation system including MVCWalker model is a recommendation list. After some time, there will be a new list coauthors for the target node. We can divide all nodes into four groups according to the following four cases (as shown in Table 4):

- A: collaborating with target nodes and recommended;
- B: collaborating with target nodes but not recommended;
- C: not collaborating with target nodes but recommended;
- D: not collaborating with target nodes and not recommended.

TABLE 4
Possible Results of Recommendation

	Recommended	Unrecommended
Collaborated	A	B
No collaborated	C	D

TABLE 5
Simulation Parameters

Parameter	Range	Default
Target nodes' degree	≥ 0	≥ 30
Partitioning Time Point	2008∼2012	2011
Iteration times	10∼100	25
Damping Coefficient	0.1∼0.95	0.8
Number of recommended nodes	5∼100	10

The metric precision is defined as:

$$P = \frac{A}{\langle A + C \rangle} \tag{7}$$

The metric recall rate is defined as:

$$R = \frac{A}{\langle A + B \rangle} \tag{8}$$

From the definitions. We can see that the higher precision and recall rate, the better performance.

In this work, we modified the general definition of the metric coverage rate, the average of shortest path from recommended nodes to the target node. We believe that, it will be a pleasantly surprised and interesting recommendation if we get the high "coverage rate", which means we recommend more "file-new" and "wide-selected" possible cooperators to the target researcher.

$$c = \frac{\sum d}{n} \tag{9}$$

where d denotes the shortest path from recommended node to target node, and n is the total number of recommended nodes. With this definition, a higher c means a better coverage.

4.3 Impact of Various Experimental Parameters

In this section, we examine the impact of different experiment parameters setting, including range of target nodes' degree, time point of data set partition, iteration times, damping coefficient and the number of recommended nodes. The ranges and default values of the parameters are summarized in Table 5. When the effect of a parameter is under examination, the other parameters are set to the default values. For each experiments, we randomly chose 100 constant nodes as recommended targets, and compute the average of precision, recall rate and coverage rate. Through the experiments, we can attain the best values of them for later tests.

4.3.1 Target Nodes' Degree

In an academic social network, there are some "strong nodes" and "weak nodes", which are defined by the number of collaborators in our model. Strong nodes have many more collaborators than weak nodes. In other words, the degree of a strong node is larger than that of a weak node. To examine the influence of the target nodes' degree in the experiments, we conduct four experiments in this part. When choosing the 100 target nodes each time, we controlled the degree respectively in range of $0 \sim 10$, $10 \sim 20$, $20 \sim 30$, > 30. Other parameters are set to the default values as Table 5. The results of the four experiments are displayed in Fig. 4.

As shown in Fig. 4, the target node's degree has an influence on the metrics with a clear trend. From a practical perspective, it is different to recommend co-authors to those who have different number of collaborators. In terms of precision in Fig. 4(a), the larger the target node's degree, the better the model's performance. Besides, we can see that MVCWalker has relatively higher precision than RWR. At the range from 0 to 10, MVCWalker performs similarly to RWR. But when the target node's degree gets larger than 30, the precision can be as high as 18.1%, much more than RWR. Thus we can conclude that, as compared against RWR, MVCwalker has higher precision for strong nodes, but performs almost the same for weak nodes.

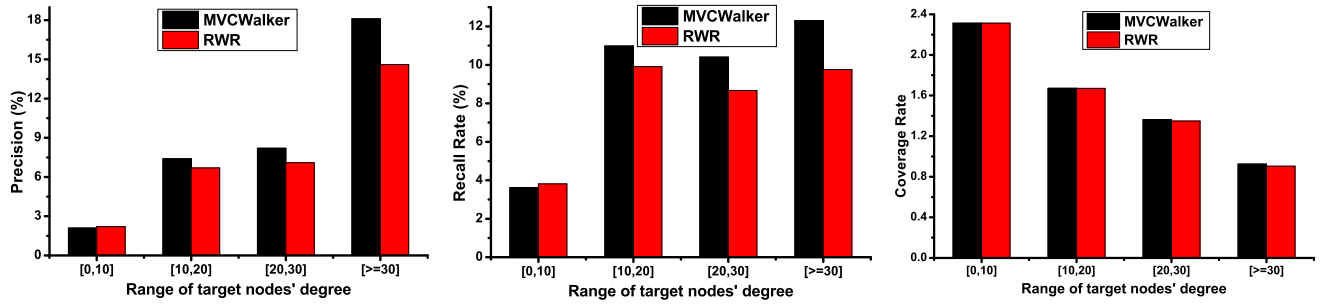
Fig. 4(b) shows the comparison of recall rate with the changing degree. The first two columns are almost the same for recall rate, while the gaps between the two models get larger for other columns. Similar to the results on precision, when the degree becomes larger than 30, the corresponding recall rate of MVCWalker is 12.3%, much higher than that of RWR (10.4%). Hence it can be realized that MVCWalker performs better than RWR on recall rate with varying target node's degree.

We can see the effect of target nodes' degree on the coverage rate from Fig. 4(c). The overall trend of coverage is distinct from the former metrics. The values of both models decrease respectively from 2.3 to 0.95 and 2.3 to 0.9. The results indicate that, for weak nodes, the neighbouring network becomes sparser with less valuable information, leading to the random walk going further; while for strong nodes, there are enough valuable nodes to be recommended in the neighbouring network.

This phenomenon is also due to that, weak nodes are not so active as strong nodes, and there is not enough valuable information for analysis and making recommendation. The analysis above leads us to the conclusions that MVCWalker outperforms RWR and it can make a better recommendation especially for strong nodes.

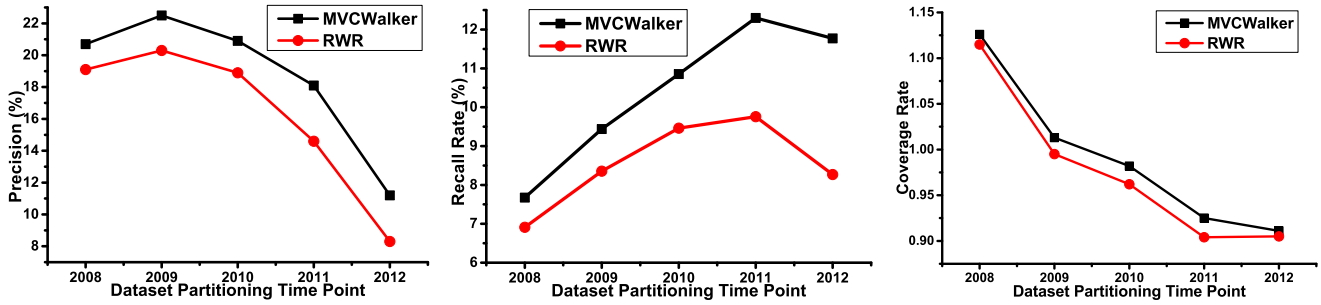
4.3.2 Partition of Training and Testing data sets

The DBLP data set contains the information ranging from 1970 to 2013. In the experiments, the data set was



(a) Precision (b) Recall Rate (c) Coverage Rate

Fig. 4. Performance of MVCWalker and Basic RWR over Target Nodes' Degree



(a) Precision (b) Recall Rate (c) Coverage Rate

Fig. 5. Performance of MVCWalker and Basic RWR over Dataset Partitioning Time Point

divided into two subsets (Training Set and Test Set) by the year of publication, based on the concept of split [40]. In this paper, we call the year as the partitioning time point. For example, the value of 2010 on X-axis in Fig. 5. means that the data before 2010 constitutes the training set while the data after 2010 make up the test set.

The effect that partitioning time point has on the performance of MVCWalker and RWR is depicted in Fig. 5. From the figures, we can see that MVCWalker performs better than RWR.

The results in Fig. 5(a) and 5(b) illustrate that both lines of precision and recall rate are similar to parabolas. In the case of MVCWalker, it recommends with higher precision and recall rate than RWR when partitioning time point ranges from 2008 to 2012. But in Fig. 5(a), the peak point is 2009, which means that when the partitioning time point is 2009, we can get the highest precision 22.5%. While the recall rate in Fig. 5(b) gets the best performance with the value 12.3% at the point of 2011. It is worth mentioning that, the trend of precisions for MVCWalker and RWR are similar. But for recall rate, the gap between MVCWalker and RWR is larger for the last three partitioning time points than before. In terms of coverage rate, it drops when the partitioning time point is increasing as shown in Fig. 5(c). The explanation for it is that the academic social network extracted from the data set enlarges in scale and tightens its topology, when the partitioning time point increases, making it faster to find a collaborator. Fig. 5(c) validates our thought that there is a trade off between precision and coverage.

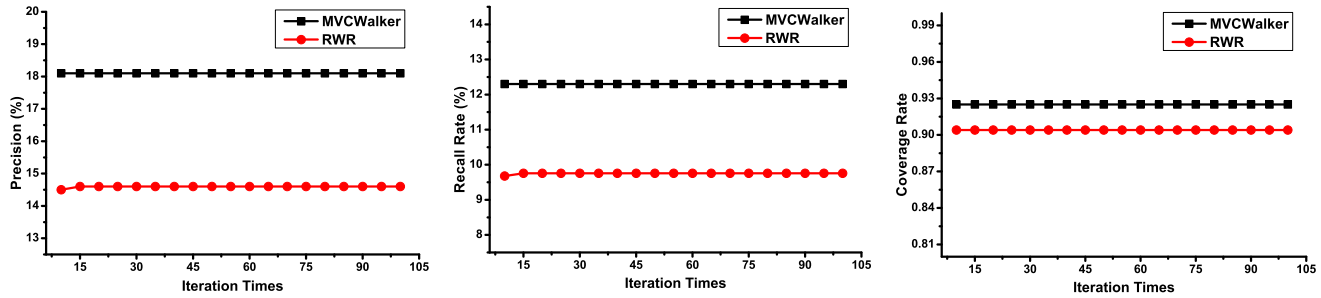
4.3.3 *Iteration Times*

Fig. 6 describes the performance of recommendation under different iteration times, which represent the number of matrix multiplication operations in the relevant equation. A higher number of iterations means that the random walk will conduct more matrix multiplication operations before getting the recommended list.

The three sub-figures share one feature in common. The three metrics show no significant changes when the iteration times get bigger. But after having a close-up view of the results, we can come up with some details. In the case of RWR, according to Fig. 6(a) and Fig. 6(b), both precision and recall rate are lower than common values until 15 iteration times, then the lines become horizontal. The common values of precision and recall rate are respectively 14.6 % and 9.76%, which means that after the random walker conducts 15 times matrix multiplication operations, the MR will become convergent. So we don't need to execute too many iterations. Since several nodes are not able to converge in fewer iterations, we set the iteration times to 25.

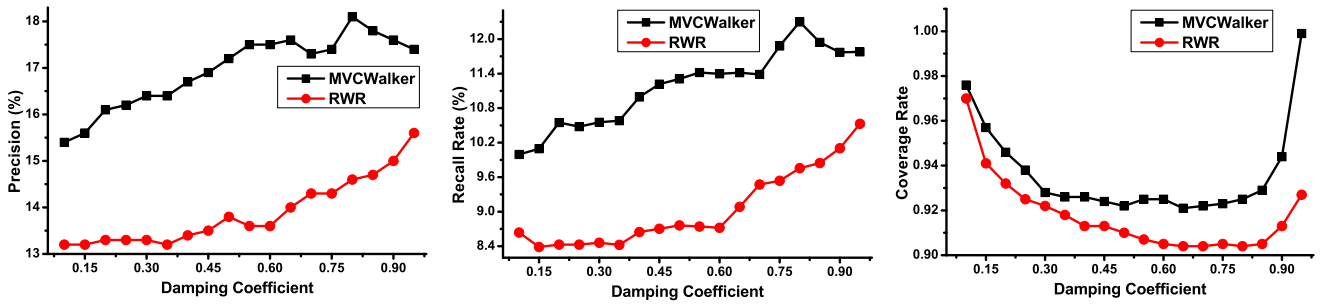
4.3.4 *Damping Coefficient*

In Random Walk model, there is a damping coefficient, which is usually set to 0.85, e.g. in PageRank. According to equation (3), the value of damping coefficient determines the probability of the walker continuing walking to the next neighbor. This parameter has a realistic significance as it controls how far the MR value will be dispersed. In this section, we analyze how the



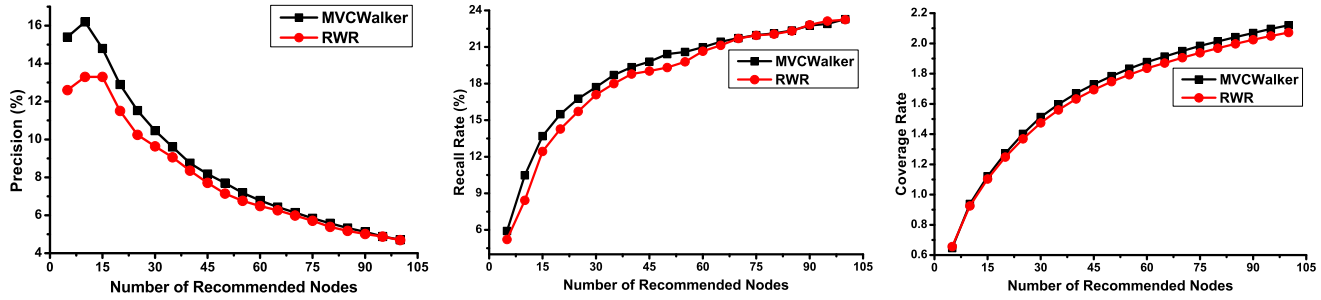
(a) Precision (b) Recall Rate (c) Coverage Rate

Fig. 6. Performance of MVCWalker and Basic RWR over Iteration Times



(a) Precision (b) Recall Rate (c) Coverage Rate

Fig. 7. Performance of MVCWalker and Basic RWR over Damping Coefficient



(a) Precision (b) Recall Rate (c) Coverage Rate

Fig. 8. Performance of MVCWalker and Basic RWR over Number of Recommended Nodes

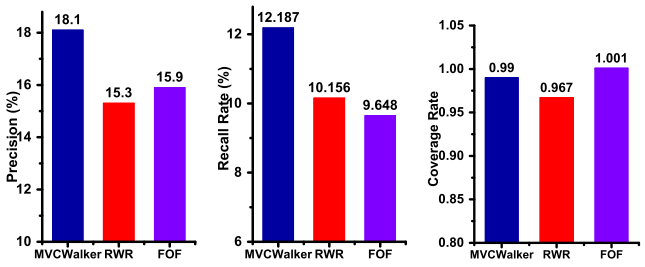
damping coefficient influences the performance of the two algorithms in terms of the three metrics.

Generally, as depicted in Fig. 7, MVCWalker and RWR almost share the same trend for the majority of tested data, while MVCWalker keeps recommending with higher precision, recall rate and coverage rate, as compared against the RWR approach. Thus we prefer to focus on describing the features of MVCWalker, instead of both of them.

Fig. 7(a) shows that the influence of damping coefficient on precision is significant. We can see that, the precisions are generally increasing with the growth of damping coefficient. In the case of MVCWalker, it can be as high as 18.1%, corresponding to the damping coefficient of 0.8. In the case of RWR, we can find that the precision is also high at this point. According to Fig. 7(b), the recall rate reaches the highest value of 12.3% when the damping coefficient is 0.8. From Fig. 7(a) and Fig. 7(b) we can see that both precision and recall rate decrease when damping coefficient becomes larger than 0.8 for MVCWalker. Moreover, from Fig. 7(c), we can see that the coverage rate generally decreases until the damping coefficient is over 0.8, and then increases rapidly. Since the point 0.8 is exactly the peak of precision and recall rate for MVCWalker, it can be verified again that there is a trade-off between recommendation precision, recall rate and coverage. Considering the importance of precision and recall rate, we regard 0.8 as a better damping coefficient for MVCWalker.

4.3.5 *Number of Recommended Nodes*

Fig. 8 illustrates how the number of recommended nodes influences the performance of MVCWalker and RWR with respect to precision, recall rate and coverage rate.



(a) Precision (b) Recall Rate (c) Coverage Rate

Fig. 9. Compare the Performance of MVCWalker, RWR and FOF Model

Fig. 8(a) shows the trend of precision. We can easily find that the precision decreases dramatically with the number of recommended nodes increasing. The highest precision of MVCWalker is 16.2% when we recommend 10 nodes to a target node while the highest precision of RWR is 13.3% when we return a 10-node recommendation list. The reason behind this is that according to equation (7), if we recommend more nodes, both the values of A and C rise, but C grows faster than A, resulting in the precision becomes smaller.

In terms of the performance of recall rate, Fig. 8(b) shows that the recall rate increases gradually. The result is opposite to that of precision. According to equation (8), the increase of the number of recommended nodes makes A grow while $(A + B)$ remains the same. Consequently, the recall rate increases.

Fig. 8(c) also depicts clearly that precision is almost inverse to coverage. Additionally, it is shown by the figure that MVCWalker performs slightly better than RWR.

4.4 Comparison with Other Methods

Reflecting on the experiments above, the consideration of academic social factors (i.e. coauthor order, latest collaboration time point and collaboration times) helps MVCWalker recommend more precisely with higher recall rate, in a wider scope in a coauthor network, and it performs better than the benchmark model RWR. Besides, the parameters we take into account affect the performance in diverse manners and we have found their best values for MVCWalker.

In order to further prove its better availability, we carry out experiments to compare the performance of MVCWalker, RWR and FOF. We run 100 times for each model, and keep the 100 target nodes same for each model. The MVCWalker and RWR are conducted over the five optimized parameters in previous experiments, i.e. target node's degree: > 30; partitioning time point: 2011; iteration times: 25; damping coefficient: 0.8; and number of recommended nodes: 10.

The results are shown in Fig. 9. The precision of MVCWalker is 18.1%, in comparison to 15.3% of RWR and 15.9% of FOF. In the case of recall rate, MVCWalker has the best value, 12.187%, which is higher than the recall rate of RWR and FOF, i.e. 10.156% and 9.648% respectively. It is clear that both precision and recall rate of MVCWalker are higher than those of RWR and FOF. Moreover the precision of FOF is a little higher than that of RWR, but its recall rate is a little lower than RWR. The coverage rate of MVCWalker is not so good as the first two indicators, better than that of RWR for our optimizing and lower than that of FOF. However, as can be seen in Fig. 9. the differences between their coverage rates are very small.

One more thing worth noting is the time complexity. We run 100 times for each model, and record the average running time. In FOF, making recommendation one time spend 1.2 seconds on average, in comparison to 3 seconds in both MVCWalker and RWR. According to section 3.3, to acquire the value of LIM, MVCWalker does more extra computation than RWR, but why is the recommendation efficiency similar? This is because, the guidance work in MVCWalker can enable the walker to encounter the most valuable node quicker and make the iteration stop at a faster rate, which we can identify in Fig. 6, the lines for MVCWalker become smooth earlier. In terms of time complexity, MVCWalker are not dominant comparing to FOF. But for most recommendation system, this disadvantage in time complexity is acceptable and does not much affect the recommend effectiveness.

In summary, we can still claim that the three factors we explore perform quite well, and MVCWalker is more effective than RWR and FOF in terms of its precision and recall rate.

5 Conclusion

In this paper, we focused on how to find scholars' MVCs based on coauthor networks (i.e. big scholarly data) which is rarely studied in the literature. To this end, we proposed a new model named MVCWalker, by injecting three academic factors into RWR. These factors are coauthor order, latest collaboration time point and collaboration times, constituting the weight of link importance between two authors for recommendation. We conducted extensive experiments on a subset of DBLP data set to examine the performance of MVCWalker with respect to various aspects, such as varying parameters and impact of the factors. We also conducted the RWR model and FOF model on the data set as comparisons. The experimental results show that our proposed approach performs better than RWR and FOF.

Nonetheless, there is still room for future study in this direction. We only count on three academic factors while many other features such as citation relation exist and should be explored in the direction of MVCWalker. Besides, there are more reasons for two scholars with no collaboration before to cooperate. For example, they might attend the same meeting and get acquainted to each other by chance, or they are from the same institution. The relationship among co-authors of a paper is

far more complicated than what we have imagined. As a future work, more experiments on the entire DBLP data set should be conducted.

REFERENCES

[1] J. S. Katz and B. R. Martin, "What is research collaboration?," *Research Policy*, vol. 26, no. 1, pp. 1–18, 1997.

[2] S. Lee and B. Bozeman, "The impact of research collaboration on scientific productivity," *Social Studies of Science*, vol. 35, no. 5, pp. 673–702, 2005.

[3] H.-H. Chen, L. Gou, X. Zhang, and C. L. Giles, "Collabseer: A search engine for collaboration discovery," in *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries (JCDL'11)*, (Ottawa, Ontario, Canada), pp. 231–240, 2011.

[4] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'08)*, (Las Vegas, Nevada, USA), pp. 990–998, ACM, 2008.

[5] J. Tang, S. Wu, J. Sun, and H. Su, "Cross-domain collaboration recommendation," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'12)*, (Beijing, China), pp. 1285–1293, ACM, 2012.

[6] G. R. Lopes, M. M. Moro, L. K. Wives, and J. P. M. De Oliveira, "Collaboration recommendation on academic social networks," in *Advances in Conceptual Modeling–Applications and Challenges*, pp. 190–199, Springer, 2010.

[7] J. Herlocker, J. A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," *Information Retrieval*, vol. 5, no. 4, pp. 287–310, 2002.

[8] A. Töscher, M. Jahrer, and R. Legenstein, "Improved neighborhood-based algorithms for large-scale recommender systems," in *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition (NETFLIX'08)*, (Las Vegas, Nevada), pp. 4:1–4:6, ACM, 2008.

[9] D. M. Boyd and N. B. Ellison, "Social network sites: Definition, history, and scholarship," *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007.

[10] G. Lopes, M. M. Moro, L. K. Wives, and J. P. M. de Oliveira, "Collaboration recommendation on academic social networks," in *Advances in Conceptual Modeling C Applications and Challenges*, vol. 6413 of *Lecture Notes in Computer Science*, pp. 190–199, Springer Berlin Heidelberg, 2010.

[11] J. Li, F. Xia, W. Wang, Z. Chen, N. Y. Asabere, and H. Jiang, "Acrec: A co-authorship based random walk model for academic collaboration recommendation," in *The 23rd International World Wide Web Conference (WWW), Companion Volume*, (Seoul, Korea), 2014.

[12] L. Backstrom and J. Leskovec, "Supervised random walks: Predicting and recommending links in social networks," in *Proceedings of the fourth ACM international conference on Web search and data mining (WSDM'11)*, (Hong Kong, China), pp. 635–644, 2011.

[13] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *Proceedings of the Sixth International Conference on Data Mining*, pp. 613–622, 2006.

[14] A.-L. Barabási and J. Frangos, *Linked: The New Science of Networks Science Of Networks*. Basic Books, 2002.

[15] J. Freyne, S. Berkovsky, E. M. Daly, and W. Geyer, "Social networking feeds: Recommending items of interest," in *Proceedings of the fourth ACM conference on Recommender systems (RecSys'10)*, pp. 277–280, 2010.

[16] J. He and W. W. Chu, "A social network-based recommender system (snrs)," in *Data Mining for Social Network Data*, vol. 12 of *Annals of Information Systems*, pp. 47–74, Springer US, 2010.

[17] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the fourth ACM international conference on Web search and data mining (WSDM'11)*, (Hong Kong, China), pp. 287–296, 2011.

[18] S. Perugini, M. A. Gonçalves, and E. A. Fox, "Recommender systems research: A connection-centric survey," *Journal of Intelligent Information Systems*, vol. 23, no. 2, pp. 107–143, 2004.

[19] A. Chin, B. Xu, H. Wang, and X. Wang, "Linking people through physical proximity in a conference," in *Proceedings of the 3rd international workshop on Modeling social media*, pp. 13–20, ACM, 2012.

[20] D. H. Lee, P. Brusilovsky, and T. Schleyer, "Recommending collaborators using social features and mesh terms," *Proceedings of the American Society for Information Science and Technology*, vol. 48, no. 1, pp. 1–10, 2011.

[21] M. Pavlov and R. Ichise, "Finding experts by link prediction in co-authorship networks.," *FEWS*, vol. 290, pp. 42–55, 2007.

[22] W. Glänzel and A. Schubert, "Analysing scientific networks through co-authorship," in *Handbook of quantitative science and technology research*, pp. 257–276, Springer, 2005.

[23] A. L. Barabási, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek, "Evolution of the social network of scientific collaborations," *Physica A: Statistical Mechanics and its Applications*, vol. 311, no. 3-4, pp. 590–614, 2002.

[24] M. E. J. Newman, "Scientific collaboration networks. i. network construction and fundamental results," *Physical Review E*, vol. 64, p. 016131, Jun 2001.

[25] X. Liu, J. Bollen, M. L. Nelson, and H. Van de Sompel, "Co-authorship networks in the digital library research community," *Information processing & management*, vol. 41, no. 6, pp. 1462–1480, 2005.

[26] W. Liu and L. Lv, "Link prediction based on local random walk," *Europhysics Letters (EPL)*, vol. 89, no. 5, p. 58007, 2010.

[27] D. Liben-Nowell and J. Jon 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, 2007.

[28] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'10)*, (Washington, DC, USA), pp. 243–252, ACM, 2010.

[29] M. A. Brandão, M. M. Moro, G. R. Lopes, and J. P. Oliveira, "Using link semantics to recommend collaborations in academic social networks," in *Proceedings of the 22nd international conference on World Wide Web companion (WWW'13 Companion)*, pp. 833–840, 2013.

[30] M. Jamali and M. Ester, "Trustwalker: A random walk model for combining trust-based and item-based recommendation," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'09)*, (Paris, France), pp. 397–406, 2009.

[31] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 355–369, 2007.

[32] H. Tong, C. Faloutsos, and J.-Y. Pan, "Random walk with restart: Fast solutions and applications," *Knowledge and Information Systems*, vol. 14, no. 3, pp. 327–346, 2008.

[33] I. Konstas, V. Stathopoulos, and J. M. Jose, "On social networks and collaborative recommendation," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 195–202, ACM, 2009.

[34] H. Tong and C. Faloutsos, "Center-piece subgraphs: Problem definition and fast solutions," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'06)*, (Philadelphia, PA, USA), pp. 404–413, 2006.

[35] A. Töscher, M. Jahrer, and R. M. Bell, "The bigchaos solution to the netflix grand prize," *Netflix prize documentation*, 2009.

[36] M. Ley, "Dblp: some lessons learned," *Proceedings of the VLDB Endowment*, vol. 2, no. 2, pp. 1493–1500, 2009.

[37] M. Ley, "The dblp computer science bibliography: Evolution, research issues, perspectives," in *String Processing and Information Retrieval* (A. H. F. Laender and A. L. Oliveira, eds.), vol. 2476 of *Lecture Notes in Computer Science*, pp. 1–10, Springer Berlin Heidelberg, 2002.

[38] F. Fouss and M. Saerens, "Evaluating performance of recommender systems: An experimental comparison," in *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'08)*, vol. 1, (Sydney, Australia), pp. 735–738, 2008.

[39] G. Shani and A. Gunawardana, "Evaluating recommendation systems," in *Recommender Systems Handbook*, pp. 257–297, Springer US, 2011.

[40] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison Wesley Professional, 2 ed., 2011.