MVCWalker: An Academic Factors Injected Random Walk Model for Most Valuable Collaborators Recommendation

Feng Xia, *Senior Member, IEEE,* Jing Li, Zhen Chen, Wei Wang, and Laurence T. Yang

Abstract—In academia, scientific research achievements would be inconceivable without scientific collaboration and cooperation among researchers. Previous study confirms that productive scholars tend to be more collaborative. However, it is often difficult and time-consuming for researchers to discover most valuable collaborators (MVCs) from the huge volume of big scholarly data. In this work, we present MVCWalker, an innovative model standing on the shoulders of RWR (random walk with restart) to recommend collaborators to scholars based on big scholarly data. We exploit academic factors (i.e. coauthor order, latest collaboration time point and times of collaboration) to define link importance on academic social networks, for the sake of recommendation quality. We conduct experiments on DBLP data set to compare MVCWalker against the basic model of RWR in different aspects (e.g. the influence of various parameters, the effect of academic factors and the performance on the whole DBLP dataset). The results show that incorporating these factors into RWR can improve the precision, recall rate and coverage rate of recommendation.

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

1 Introduction

In academic culture, collaboration among researchers has been increasingly popular and necessary. Previous study confirms 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, researches on big scho[illegible]mic social networks [4], [5] show th[illegible]rative context prefer to find valu[illegible] yet known to them, or contact with faraway researchers, in addition to staying in touch with their close colleagues.

collaboration
recommendation
on social network

Unfortunately, the huge size of big scholarly data makes it a big 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 recommender systems do in social networking sites (SNS). Specifically, a feature in SNS called ”People You May Know” has been proved of merit to recommend people based on a FOF (friend of friends) method [6], [7]. Besides, typical SNS systems such as Facebook usually recommend friends that the users have already known offline [8]. However, due to the nature of users' needs for general social networking sites (i.e. entertainment), this kind of methods recommends already known friends based on social relationship, and has inherent weakness on satisfying scholars' requirement of figuring out valuable collaborators. Recommending researchers in academic social networks (based on big scholarly data) is an increasingly important topic, because it is dissimilar from traditional recommendation of friends in social networks. For instance, before deciding a collaborator, researchers often have to consider e.g. whether someone has common research interests, if he/she is valuable in research from a collaboration perspective, and how to get connected with him or her. Furthermore, academic achievements on paper publications reflect numerous aspects of scholars' value. In order to satisfy scholars' special requirements, it becomes vital to develop collaborator recommendation methods based on coauthor networks.

Coauthor network is an extraordinary social network for its academic property of coauthorship, which is a simple graph evolving from the author-paper binary graph (as shown in Fig. 1). However, current researchers usually directly treat the links with equal importance, neglecting whether the relationship is strong or not. There are many factors that influence the measurement of the relationships between researchers, e.g. latest coauthoring time. When choosing a collaborator, for instance, a scholar might prefer to a researcher with whom he/she coauthored a paper within the past year, rather than coauthors of ten years ago. Furthermore, as pointed out in [9], when recommending new coauthors on academic social networks, social interactions and its relational aspects should be considered. As a consequence, academic social factors can be taken into consideration to help recommend MVCs.

The RWR (Random Walk with Restart) model can

- *F. Xia, J. Li, Z. Chen, and W. Wang 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.*

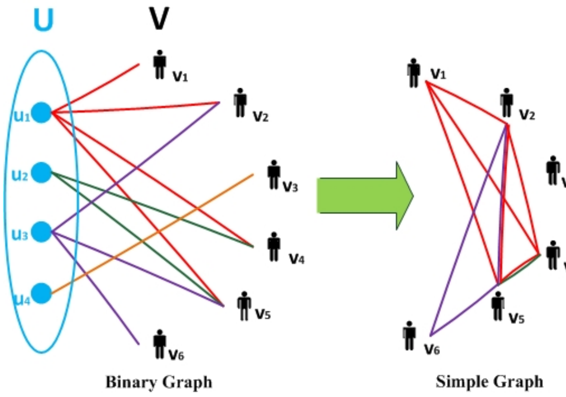


Fig. 1. Extraction from a Binary Graph to a Simple Graph. U is a list of papers, V is the list of authors. The figure shows three cases: (1) If a scholar has no collaborator, he/she is an isolated node, just like v_3; (2) If two authors coauthor a paper, there is a link between them, such as (v_1, v_2)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

easily combine the information from the network structure with node and link attributes [10]. Hence, we take advantage of RWR and optimize it for the guidance of a random walk on a graph extracted from coauthor networks. Besides, we define link importance based on some academic factors (i.e. the latest collaboration time point, the times of collaboration and the coauthor order). To be exact, we explore them to measure coauthors' 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. All the ideas above contribute to the novelty of our proposed model, namely MVCWalker: recommending most valuable collaborators with academic factors injected random walk model.

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 the others.
- In order to improve the recommendation quality and accuracy, we propose to define the link importance by exploiting three specific academic network factors including coauthor order, collaboration time points (i.e. the latest collaboration time) and frequency of collaboration (i.e. collaboration times).
- We conduct extensive experiments on DBLP data set to evaluate the performance of the proposed solution in various scenarios as compared against the basic model of RWR. Promising results are presented and analyzed.

The remainder of the paper is structured as follows. Section 2 briefly surveys the related work in regard of social recommender systems, features of coauthor networks, link prediction and RWR. We discuss the details of our proposed model in Section 3, including e.g. problem statement, workflow and computation of link importance. In Section 4 we introduce the experiment settings and analyze the performance of MVCWalker as compared against RWR. Finally Section 5 concludes the paper.

2 Related Work

2.1 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 [11]. Recently much research work has been done on how to utilize social network information to improve recommender systems [12], [13]. For instance, Ma *et al.* [14] elaborate how social networks information can benefit recommender systems and provide a general method for improving recommender systems by incorporating social network information. Perugini *et al.* [15] suggest that recommendation has an intrinsic social element and it's essentially intended to connect people.

In contrast to previous work in this field, we aim to recommend MVCs on academic social networks (based on e.g. big scholarly data) in this work. The requirements for recommending collaborators, especially academic collaborators, are different from recommending items. In academic social networks, the links between authors represent the coauthoring relationship. For the purpose of collaborator recommendation, we should consider aspects that influence academic collaboration relationships. In [9] Lopes *et al.* consider the researcher's publications area and the vector space model to make collaboration recommendation on academic social networks. A search engine for collaboration discovery named Collabseer has been proposed in [3]. However, our work takes three specific academic factors into consideration.

2.2 Characteristics of Coauthor Networks

The coauthorship network of scientists represents a kind of complex social networks. In these networks two scholars are normally considered connected if they have coauthored one or more papers together. Coauthor network characteristics have been studied comprehensively. By mapping the electronic database containing all relevant journals in mathematics and neuro-science for an 8-year period (1991-1998), Barabasi *et al.* [16] infer the dynamics and the structural mechanisms that govern the evolution and topology of coauthorship networks. They analyze the basic network properties of academic social networks in terms of degree distribution, average separation, clustering coefficient, average degree and so on. Their results indicate 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 [17] studies 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 they find 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. Based on these characteristics, we propose our model and design our experiments.

2.3 Link prediction

To some degree, we can formalize academic friend recommendation as a link-prediction problem. Many approaches have been proposed for various link prediction [19]. For instance, David *et al.* [18] define 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 develop approaches to link prediction based on measures for analyzing the ″proximity″ of nodes in a network. The approaches have been applied to large social networks and the results suggest that fairly subtle measures for detecting node proximity can outperform direct measures.

In [20], Lichtenwalter *et al.* examine important factors for link prediction in networks and provide 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 outperforms unsupervised link prediction methods.

The work more closely related to ours is [21], whose emphasis is also on recommending academic friends and considering the link semantics. The authors propose two new metrics respectively representing the institutional affiliation and the geographic location of the researchers for recommending new collaborators. Our work differs from [21] in that we consider the details of coauthor relationship.

2.4 Random Walk with Restart

Random Walk with Restart (RWR) provides a good way to measure how closely related two nodes are in a graph [22]. It has been successfully used in numerous areas including e.g. collaborative recommendations and link prediction, etc. In [23], Mohsen *et al.* propose a random walk model combining the trust-based and collaborative filtering approaches for recommendation. They take advantage of random walk to define and measure the confidence of a recommendation. In [24], Fouss *et al.* present a new perspective on characterizing the similarity among elements of a graph. It is based on a Markov-chain model of random walk through the database. Although these studies are quite close to our work, we exploit a different model named MVCWalker, in which we inject academic factors to guide random walk.

TABLE 1
List of Symbols

Symbol	Definition
MR	$N * 1$ ranking score vector of MVCs
p_i	Node i
$L(p)$	Count of all the neighbors of node p
$M(p)$	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

The main reason why we adopt RWR is that it allows us to directly predict the closeness among researchers from the coauthor network by taking into account the link importance. More importantly, when a single walker walks, it can calculate the social influence on recommendation by traversing the whole graph.

3 Design of MVCWalker

In this section, we describe the details of our model, i.e. MVCWalker. Following problem statement, we introduce the RWR model. An overview of MVCWalker is then presented. Moreover, 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 and Notations

Problem** 1: **Available Academic Social Network Features
As mentioned in Section 1, social interactions and its relational aspects can help to recommend collaborators. However, what relations do academic social networks are featured? Whether part or all of the features are available?

The common features have been described in Section 2 and we choose coauthor order, latest collaboration time point, collaboration times for MVC recommendation. The detailed principles will be presented in Section 3.4.

Problem** 2: **MVC Recommendation
To recommend MVCs for a scholar, what factors can be

considered? How to design an algorithm (or model) to achieve the goal?

3.2 RWR Model

In RWR 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 $MR(p)$ is the rank score of node p, which is the quantized importance of node p 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 of the walker continuing walking to the next neighbor. Above all, in RWR model, the walker has some probability to randomly skip to any other nodes.

Equation (1) represents only the step to get the rank score of a node. As for each node in the whole graph, random walk with restart is defined by (2), which is an iterative process. $\mathbf{S}$ is the transfer matrix, representing the probability for each node to skip to other nodes. I is a row vector, calculated as $I = (1, \ldots, 1, \ldots, 1)$.

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

For the recommendation in this paper, we use the personalized RWR model, which can be defined as:

$$MR^{(t+1)} = \alpha \mathbf{S} MR^{(t)} + (1-\alpha) q \tag{3}$$

$MR^{(t)}$ 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 RWR has many good properties: as compared with those common neighbors 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 [25].

However, basic random walk models usually assume that the weights of edges are the same, and define the cells 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 detail.

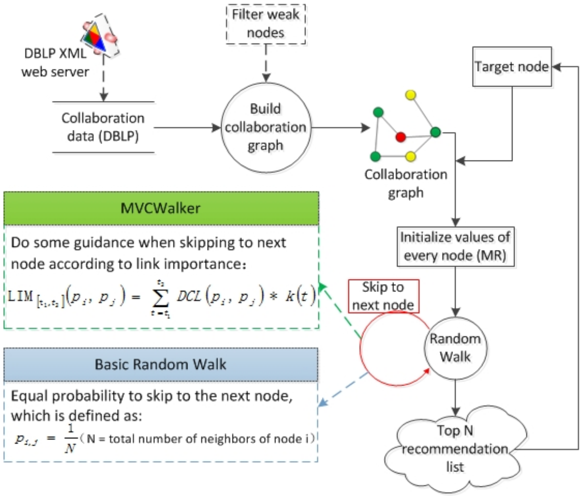

DBLP XML
web server
Filter weak nodes
Target node
Collaboration data (DBLP)
Build collaboration graph
Collaboration graph
MVCWalker
Do some guidance when skipping to next node according to link importance:
$\mathrm{LIM}_{[t_1,t_2]}(p_i, p_j) = \sum_{t=t_1}^{t_2} DCL(p_i, p_j) * k(t)$
Initialize values of every node (MR)
Skip to next node
Random Walk
Basic Random Walk
Equal probability to skip to the next node, which is defined as:
$p_{i,j} = \frac{1}{N}$ (N = total number of neighbors of node i)
Top N recommendation list

Fig. 2. Structure of MVCWalker.

3.3 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 people's 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 features data into the network structure to bias the random walk such that it will more easily traverse to the positive nodes.

The detailed process of MVCWalker is described below and the corresponding pseudocode is given by Algorithm 1. The structure of MVCWalker is given 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 the authors as nodes in the network. Before that, it's necessary to filter out the isolated nodes and others which are too weak (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 can be defined as set E. In this step, we should 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 random walk, we should get the transfer matrix S as described in Algorithm 1. $S = ComputeTransferMatrix()$. To be specific,

give P_i as the current node while P_j as the next node. 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.

- When our MVCWalker starts, initialize the rank score vector $MR^{(0)}$ and the restart probability vector q as (0,. . . ,1,. . . ,0). Set target node P_i as 1 while others 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 by corresponding rank score.
- Recommend nodes in the TOP N of the list MR to target nodes. Of course, we can take out the nodes which have been in its coauthor list before recommending. That is the new coauthors recommendation.

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

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

Given above is the whole process of MVCWalker. Below we will detail how to calculate the link importance by taking into account the three academic factors.

3.4 Link Importance

As mentioned above, when people extract a simple graph from a binary graph, they usually overlook information about link features. Traditional RWR model assumes that links among nodes are of the same importance, which implies that the probability of choosing any node to walk next is equal. Furthermore, RWR-based algorithms rarely consider the impact of link features, which indicate cooperation relationship closeness between nodes at link ends. In reality, however, people might be more willing to choose certain nodes of high feature value for them. Therefore, a critical requirement of the MVC recommendation algorithm is to assign cooperation graph edge with weights 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) and intend to calculate edge weights based on three factors: coauthor order, latest collaboration time point, and collaboration times.

3.4.1 Coauthor Order

There is always a list of (co)authors for one paper. Normally, their contributions to the paper differ from each other. For example, the first and the second authors usually make more contributions than the rest authors. In such cases, the cooperation relationship between the first two authors is competently strong. Moreover, 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 $j > 1$, and there are more than one author of a paper. For the sake of simplicity, we calculate $DCL(p_i, p_j)$ as follows.

$$DCL(p_i, p_j) = \begin{cases} \frac{1}{i} + \frac{1}{j} & j \leq 3 \\ \frac{1}{i} + \frac{2}{j} & j > 3, i \leq 3 \\ \frac{2}{i} + \frac{2}{j} & i > 3 \end{cases} \tag{4}$$

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.

3.4.2 Latest Collaboration Time Point

Previous studies consider the social networks to be static. However, academic social networks are time-varying, where the links among scholars change over time. For instance, scholars might be more willing to collaborate with who coauthored a paper last month, as compared to the coauthors of ten years ago. Hence, we measure the link dynamics using $LIM_t(p_i, p_j)$ (i.e. Link Importance):

$$LIM_t(p_i, p_j) = DCL(p_i, p_j) * k(t) \tag{5}$$

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:

$$k(t) = \frac{t_i - t_0}{t_c - t_0} \tag{6}$$

where t_i is the link formation time (in year here), t_c is the current time (i.e. 2013 in this paper) and t_0 is the first link formation time.

3.4.3 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. However, no previous study has taken into consideration the times of collaboration. Here we measure the impact of different times of coauthoring as follows:

$$\begin{aligned} LIM_{[t_1,t_2]}(p_i,p_j) &= \sum_{t=t_1}^{t_2} LIM_t(p_i,p_j) \\ &= \sum_{t=t_1}^{t_2} DCL(p_i,p_j) * k(t) \qquad (7) \end{aligned}$$

In the above equation, during time period (t_1,t_2), if there are n links between p_i and p_j, we will calculate the sum of each link importance.

4 Evaluation and Analysis

We conducted extensive experiments using data from DBLP [26], a computer science bibliography website hosted at University Trier. In this section, we describe the processing of DBLP data set, the evaluation metrics we use and the experiments we conduct to evaluate the performance of MVCWalker, as well as detailed analysis of the results.

We designed different experiments to compare MVCWalker with the basic model of RWR in terms of multiple metrics (i.e. precision, recall rate and coverage rate). For each set of experiments, we examine several aspects, including the influence of different parameters, the performance on the entire DBLP data set, and the effect of the three factors we consider.

All experiments are 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 choose 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 [26] and is expected to be the most comprehensive bibliography of the computer science field with support from the Anthology project [27]. 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. According to the data provided by the official web site of DBLP, we list the number of authors per publication in Fig. 3. From the figure we can see that when the number of authors are larger than five, the corresponding number of publications drops dramatically and this is why the scale of X-axis is 0-10.

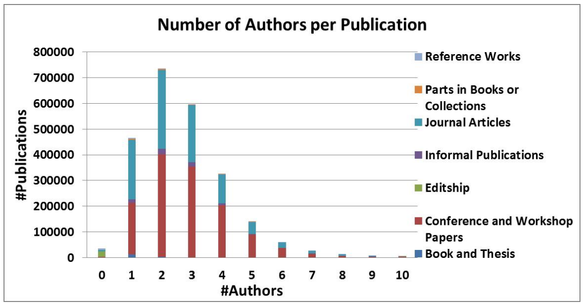


Fig. 3. Number of Authors per Publication. X-axis and Y-axis represent the number of authors and publications, respectively.

TABLE 2
Statistics of Data Set of Data Mining from DBLP

Statistics	Nodes	Edges	Average Degree
Number	59659	90282	1.513

Generally, a piece of DBLP records contains record attributes, author, title, pages, years, authors' information including person records, person IDs, etc. With the methods provided by [26], we can extract a subset of the entire data set with information we need to conduct our experiments. The reasons why we use a subset of the dataset are as follows.

- It is possible that the subset of the researchers' publications be represented by a social network [21] 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 and they are not helpful in our recommendation process as well. 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, we ignore the weak nodes whose degrees are less than ten.

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 2, covering the number of nodes, edges and average degree.

4.2 Evaluation Metrics

We choose three popular metrics, precision, recall rate and coverage rate, to evaluate the performance of MVCWalker [28], [29]. Usually, after recommendation, we can divide the nodes into four groups according to the following four cases (as shown in Table 3):

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

TABLE 3
Possible Results of Recommendation

	Recommended	Unrecommended
Collaboration	A	B
No collaboration	C	D

TABLE 4
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

- D: not collaborating with target nodes and not recommended.

The metric precision is defined as:

$$P = \frac{A}{\langle A + C \rangle} \tag{8}$$

The metric recall rate is defined as:

$$R = \frac{A}{\langle A + B \rangle} \tag{9}$$

From the definitions. We can see that a bigger precision or recall rate means a better performance.

Considering the unique feature, we define the metric coverage rate in this paper as the average of shortest path from recommended nodes to the target node.

$$c = \frac{\sum d}{n} \tag{10}$$

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 Parameters

In this section, we examine the impact of different parameters, including range of target nodes' degree, time segment, iteration times, damping coefficient and the number of recommended nodes. The ranges and default values of them are summarized in Table 4. When the effect of a parameter is under examination, the other parameters are set to the default values. 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 onto the experiments, we defined four ranges of degree according to the features of our data set, and the ranges are shown in Fig. 4.

As shown in Fig. 4, the target node's degree has an obvious influence on the metrics. From a practical perspective, it is different to recommend coauthors to those who have different number of collaborators. As for 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 we can claim that MVCWalker performs better than RWR on recall rate with varying target node's degree.

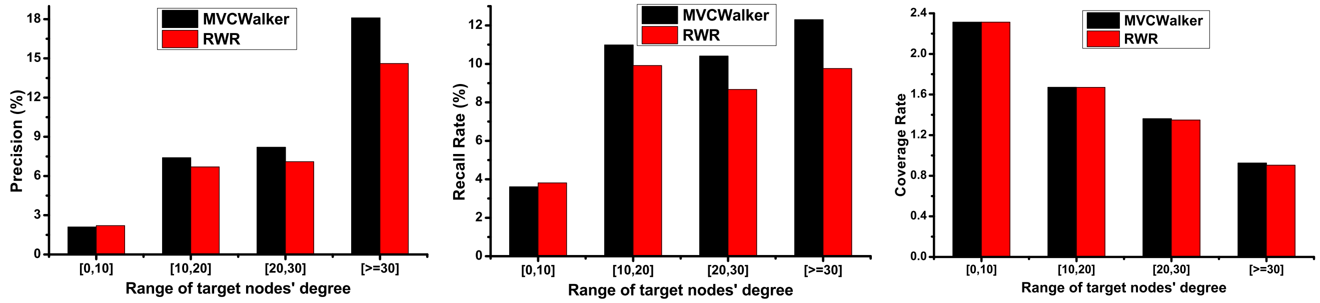
We can see the effect of target nodes' degree on the coverage rate from Fig. 4(c). The overall trend of coverage is distinct from the former metrics. The values of both models decrease respectively from 2.3 to 0.95 and 2.3 to 0.9. The results indicate that, for weak nodes, the neighbouring network becomes 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 Time Point for Dataset Partitioning

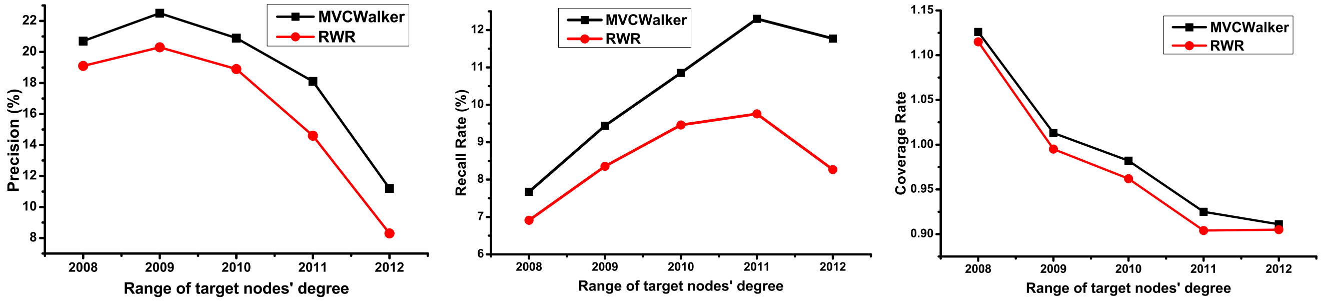
The DBLP data set contains the information ranging from 1970 to 2013. In the experiments, the data set was divided into two subsets (Training Set and Test Set) by the year of publication, based on the concept of split [30]. In this paper, we call the year of a publication as the parameter of partitioning time point. For example, the value of 2010 on X-axis means that the data before 2010 constitute 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 the difference



(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

between MVCWalker and RWR is obvious with the former performing better than the latter.

The results in Fig. 5(a) and 5(b) illustrate that both lines of precision and recall rate seems like parabolas. For 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 regard of coverage rate, it drops with the partitioning time point 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 again, which is that there is a trade off between precision and coverage.

4.3.3 Iteration Times

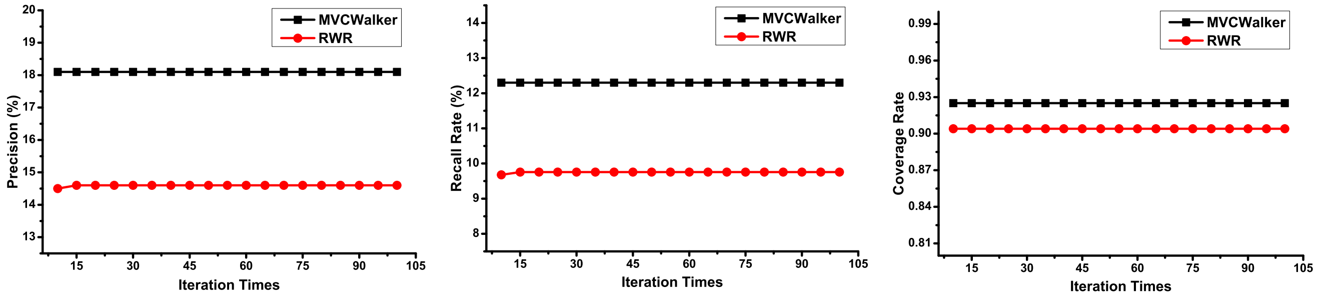
Fig. 6 describes the performance of recommendation under different iteration times, which represent the number of matrix multiplication operations in the relevant equation. A higher number of iterations means that the random walk will conduct more matrix multiplication operations before getting the recommended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

The three sub-figures share one features 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 find some details. For RWR, according to Fig. 6(a) and Fig. 6(b), we can find that both precision and recall rate are lower than common values until 15 iteration times, then the lines get horizontal. The common values of precision and recall rate are respectively 14.6 % and 9.76%, which means that after the random walker conduct 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 in the following experiments.

4.3.4 Damping Coefficient

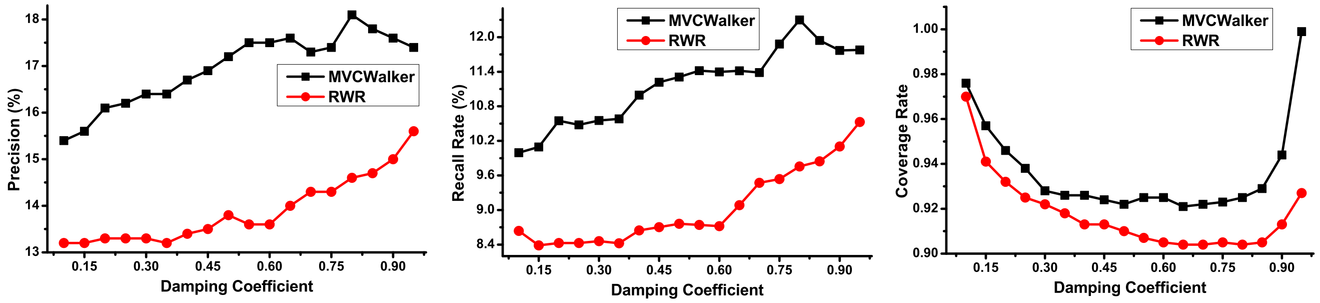
In Random Walk model, there is a damping coefficient, which is usually set to 0.85, such as in PageRank. According to (3), the value of damping coefficient determines the probability for random walker to jump back to the original node when randomly walking. This parameter has a realistic significance as it controls how far the MR value will be dispersed. In this section, we analyze how the 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 basic RWR approach. Thus we will focus on



(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

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. For MVCWalker, it can be as high as 18.1% , corresponding to the damping coefficient of 0.8. As for 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 seen again that there is a trade-off between recommendation precision and coverage. As a result, we regard 0.8 as the best 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.

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 phenomenon is obvious. According to (8), if we recommend more nodes, both the values of A and C rise, but C grows faster than A, resulting that the precision becomes smaller.

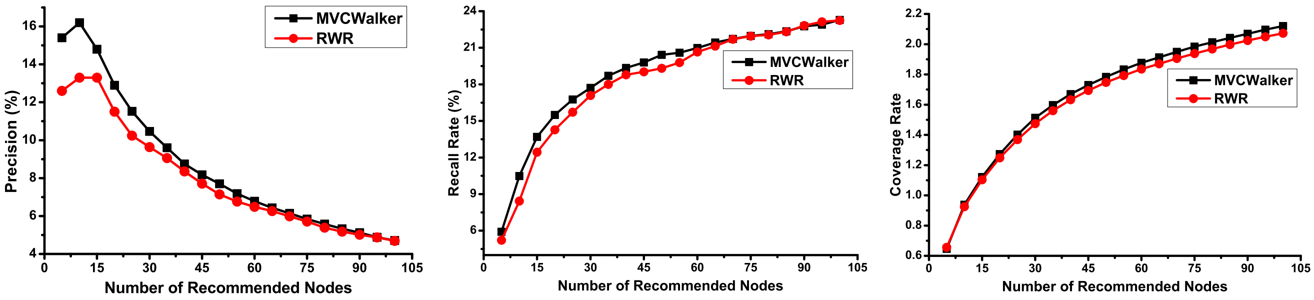
As for the performance of recall rate, Fig. 8(b) shows that the recall rate increases gradually. The result is opposite to that of precision. According to (9), the increase of the number of recommended nodes makes A grow while $(A+B)$ remains the same. Consequently, the recall rate increases.

Fig. 8(c) also depicts clearly that precision is almost inverse to coverage. Additionally, it is shown by the figure that MVCWalker performs a little better than basic RWR, but not so significant.

In summary, 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, at least not worse than the benchmark model. Besides, the parameters we take into account affect the performance in diverse manners and we have found their best values for MVCWalker.

4.4 Academic Social Factors

After determining the five variables above, we carried out experiments to examine the impact of the three factors on MVCWalker. To this end, we use only one of the factors to obtain the weight of coauthors' relationship each time. In this way, we get three separate models,



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

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

TABLE 5
Experiment Results of Various Models

Model	Precision (%)	Recall Rate (%)	Coverage Rate
RWR	15.3	10.156	0.967
MVCWalker	18.1	12.187	0.99
MVC-CO	16.5	10.968	0.976
MVC-CT	21	13.258	0.992
MVC-CTs	18.2	12.113	0.921

including MVC-CO (MVCWalker-only coauthor order considered) and MVC-CT (MVCWalker-only collaboration time considered), and MVC-CTs (MVCWalker-only collaboration times considered). Note that all these three models are special cases of our MVCWalker framework. During the experiments, we run 500 times for each model, and keep the target nodes the same. The parameters are set to the best values based on the 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.

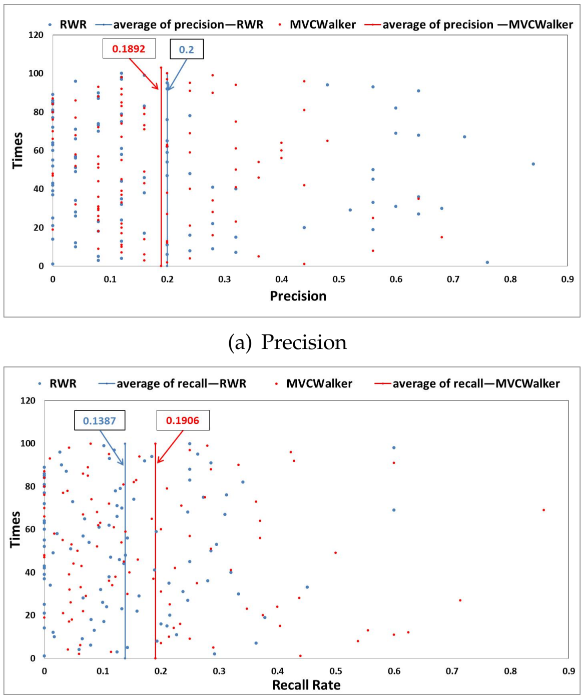
The results are shown in Table 5. It is clear that both precision and recall rate of MVC-CT and MVC-CTs are higher than those of RWR, but the coverage of MVC-CTS is not so good. For MVC-CO, the precision, recall rate and coverage rate are just improved a little bit, as compared with RWR. The results suggest that collaboration times and collaboration time points play more important roles in MVCWalker for recommendation and that coauthor order has a negative effect on MVCWalker's performance. For the weak influence of coauthor order, we may guess that, in a coauthor list, there could be many kinds of relations, e.g. teacher-student, cooperation between institutions, etc. The equation we provide in this paper can not cover all possible situations. This causes that MVCWalker is not able to recommend those potential collaborators of weak relation with target nodes. Hereby, we should conduct more experiments and take into consideration more situations to analyze and confirm the use of the coauthor order factor. In a word, we can still claim that the three factors we explore perform quite well, and MVCWalker is more effective than RWR.

4.5 Entire DBLP Data Set

Next we apply MVCWalker to the whole DBLP dataset. Fig. 9 presents the distribution of nodes with distinct values of precision and recall rate (without coverage rate here). The average values are also shown in the figure with corresponding colored lines.

We can see that MVCWalker has a better performance on recommendation with average precision of 18.92% (see Fig. 9(a)), which is much higher than those tested above with precisions of no more than 8%. Besides, its average recall rate is 19.06% (shown in Fig. 9(b)), almost the highest among the former experiments. This indicates that MVCWalker is an effective method for recommending collaborations. The results also show that MVCWalker outperforms RWR according to the combination of precision and recall. Note that MVCWalker's recall rate is about 5.2 percentages more than that of RWR (13.87%), verifying that injecting academic factors into RWR does increase the performance of the original method. The reason is that these academic factors assign strengths to edges in the network such that a random walker is more likely to visit the potential nodes (i.e. MVCs) in the future.

In the meantime, some drawbacks can be seen from the figure. The first is that there are some bad recommended nodes whose recall rate is 0. This is due to the cold start problem. In the later partitioning time point, previous scholars may collaborate with new scholars who do not exist before, so we can not find them out based on previous information. However, RWR leads to worse results than MVCWalker. It shows that MVCWalker can solve the cold start problem better. The second weakness lies in that the overall precision and recall rate of MVCWalker are lower as compared with other recommendation methods, for example, in [21] the recall rate is as high as 95.18%. Nevertheless, this does not mean that our method performs badly. This is due to the scale of data set. Our data set is much larger than that of [21], where only 629 researchers from 45 Brazilian institutions are considered.



(a) Precision

(b) Recall Rate

Fig. 9. Performance of MVCWalker and Basic RWR Tested on the Entire DBLP Data Set. The red line or dots in each sub-figure refers to data about MVCWalker, and the blue ones correspond to RWR.

5 Conclusion

In this paper, we focus 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 have proposed a new model named MVCWalker, by injecting three academic factors into RWR and the 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 the DBLP data set to examine the performance of MVCWalker with respect to various aspects, including e.g. varying parameters and impact of the factors. Both a subset of the data set and the entire data set are used respectively. The experiment results show that our proposed approach performs better than RWR.

Nonetheless, there is still room for future study in this direction. We only count on three academic factors while many other features exist, such as citation relationship. 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 coauthors of a paper is far more complicated than what we have imagined. More experiments on the entire DBLP data set may 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] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

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

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.

[28] 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.

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

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