

Received 16 May 2014; revised 31 July 2014; accepted 13 August 2014. Date of publication 00 xxxx 0000;
date of current version 00 xxxx 0000.

Digital Object Identifier 10.1109/TETC.2014.2356505

MVCWalker: Random Walk-Based Most Valuable Collaborators Recommendation Exploiting Academic Factors

FENG XIA¹, (Senior Member, IEEE), ZHEN CHEN¹, WEI WANG¹,
JING LI¹, AND LAURENCE T. YANG^{2,3}

¹School of Software, Dalian University of Technology, Dalian 116620, China

²School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

³Department of Computer Science, St. Francis Xavier University, Antigonish, NS B2G 2W5, Canada

CORRESPONDING AUTHOR: F. XIA (f.xia@ieee.org)

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 paper, we present MVCWalker, an innovative method that stands on the shoulders of random walk with restart (RWR) for recommending collaborators to scholars. 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 friend of friends in various aspects, including, e.g., the impact of critical parameters and academic factors. Our experimental results show that incorporating the above factors into random walk model can improve the precision, recall rate, and coverage rate of academic collaboration recommendations.

INDEX TERMS Most valuable collaborator, academic recommendation, big scholarly data, random walk, link prediction.

I. 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

51 how to get connected with him/her. In order to satisfy scholars' special requirements, it becomes vital to develop special
 52 collaboration recommendation methods based on ASN.
 53

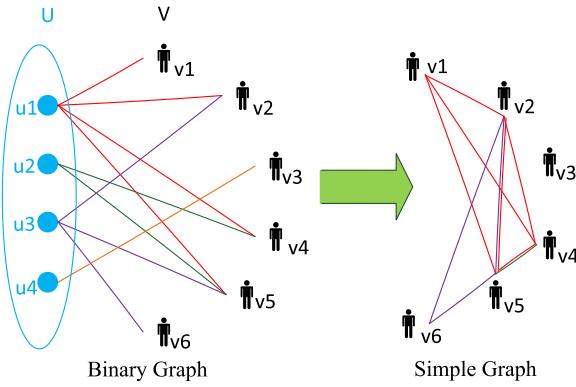


FIGURE 1. Extraction from a Binary Graph to a Simple Graph. *U* is a list of papers, *V* is the list of authors. The figure shows three cases: (1) If a scholar has no collaborator, he/she is an isolated node, just like v_3 ; (2) If two authors coauthor a paper, there is a link between them, such as (v_1,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

54 A co-author network is an extraordinary social network
 55 due to its academic property of co-authorship, which can be
 56 modeled by a simple graph evolving from the author-paper
 57 binary graph as shown in Fig. 1. The links represent the
 58 relationships between researchers, and it should be noticed
 59 that the importance of the links are different. There are many
 60 factors which influence the measurement of relationships
 61 between researchers, e.g. latest collaboration time, times of
 62 collaboration and coauthor order. Consequently, as pointed
 63 out in [6], when recommending new co-authors in academic
 64 social networks, social interactions and its relational aspects
 65 should be taken into consideration.

66 In this paper, we propose a novel random walk model
 67 named MVCWalker for recommending the most valuable
 68 collaborators to scholars, which explores the use of academic
 69 factors. This paper substantially extends our previous pre-
 70 liminary work in this line [10]. MVCWalker is a kind of
 71 Random Walk (RW) model in which the rich information
 72 of both nodes and links are taken into account when being
 73 used for recommendation technology [11]. Random Walk
 74 with Restart (RWR) is a classic instance of RW model, which
 75 can provide a good relevance score between two nodes in a
 76 weighted graph, it has been successfully used in numerous
 77 settings, such as personalized PageRank and social recom-
 78 mendation [12]. Hence, we take advantage of RW model,
 79 by utilizing RWR and optimizing it through guidance for
 80 the random walker on the coauthor networks. Additionally,
 81 we define link importance based on some academic factors
 82 (i.e. the latest collaboration time point, the times of collab-
 83 oration and the coauthor order). We further explore these
 84 three factors to measure co-authors' link importance. This
 85 can provide the random walk with more possibility to visit

the MVCs on a weighted network and help improve the
 86 recommendation quality and accuracy.

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

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

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

II. RELATED WORK

A. ACADEMIC SOCIAL RECOMMENDER SYSTEMS

114 Social networks have been studied for decades in an effort to
 115 comprehend the relationships between people and detect pat-
 116 terns in such interactions [13]. Recently much research work
 117 has been done on how to utilize social network information
 118 to improve recommender systems [14], [15]. For instance,
 119 Ma *et al.* [16] elaborated on how social networks information
 120 can benefit recommender systems and provided a general
 121 method for improving recommender systems by incor-
 122 porating social network information. Perugini *et al.* [17] suggested
 123 that recommendation has an intrinsic social element and is
 124 essentially intended to connect people.

125 In this paper, we specifically aim to recommend MVCs
 126 in academic social networks (based on e.g. big scholarly
 127 data), which is different from recommending normal friends
 128 or items in academic context. Chin *et al.* [18] learned
 129 how Offline to Online interactions in a conference can
 130 help link people together, improve friend recommendations.
 131 Lee *et al.* [19] studied how well content-based, social and
 132 hybrid recommendation algorithms predicted coauthor rela-
 133 tionship, and the result show that a hybrid algorithm combin-
 134 ing expertise and social network information outperformed
 135 better. Pavlov and Ichise [20] proposed a method that extracts
 136 structural attributes from graph of past collaborations and
 137 uses them to train a set of predictors using supervised learning
 138 algorithms, these predictors can then be used to predict future

links between existing nodes in the graph. Lopes *et al.* [6] 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.

B. COAUTHOR NETWORKS

Scientific collaboration is a complex social phenomenon in research that has been systematically studied since the 1960s. Furthermore, co-authorship is one of the most tangible and well documented forms of scientific collaboration [21]. 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.* [22] 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 [23] 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 [23] 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.* [24] 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.

C. LINK PREDICTION

We can formalize academic collaboration recommendation as a link-prediction problem. Many approaches have been proposed for various link predictions [25]. For instance, David *et al.* [26] 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.* [27] 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 [28], 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 [28] in that we consider the details of coauthor relationship.

D. ACADEMIC RANDOM WALK MODEL

Random walk model is often used in daily recommendation scenarios, etc. Mohsen *et al.* [29] 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.* [30] 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 [31]. It has been successfully used in numerous areas including e.g. collaborative recommendation and link prediction. Konstas *et al.* [32] 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.* [11] 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 [10]. 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. In this paper, we extend ACRec, inject academic factors and build MVCWaler to guide random walk. We conduct much more extensive experiments in this paper than those on ACRec. We examine the influence of two more parameters, iteration times in random walking process and partition of training and testing data sets. We perform further optimizations over the settings of these two parameters. In addition, to further verify the superiority of our method,

250 we compare MVCWalker to both the basic RWR and the
251 popular FOF model in this paper.

III. DESIGN OF MVCWALKER

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

TABLE 1. List of symbols.

Symbol	Definition
MR	$N * 1$ ranking score vector
p_i	Node p_i
$L(p_i)$	Number of all the neighbors of node p_i
$M(p_i)$	Set of nodes incident to node p_i
α	Damping coefficient: the probability of a walker walking to the next neighbor
N	Total number of nodes in a graph
S	Transfer matrix
q	$N * 1$ initial vector of MR
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

A. PROBLEM STATEMENT

258 In this paper, our goal is to find and recommend collaborators
259 to researchers. According to Section 2, the aforementioned
260 research efforts exploit collaboration recommendation
261 systems by considering various information in big scholarly
262 data. Some factors will be helpful when making collabora-
263 tion recommendation based on academic social networks.
264 In addition, the random walk model can be adopted for its
265 remarkable characteristic of integrating the rich information
266 of both nodes and links. In particular, we are targeting the
267 following problems.

268 *Problem 1:* Academic collaboration recommendation is
269 different from the traditional social recommendation. Schol-
270 ars need collaborators who have common research interests,
271 valuable and connectable. To recommend MVCs to a scholar,
272 what factors should be considered? How can a recommenda-
273 tion algorithm (or model) be designed to achieve this goal?

274 *Problem 2:* As mentioned in Section 1, social interactions
275 and its relational aspects can help recommend collaborators.
276 Various academic social network features could be consid-
277 ered. However, what relations are featured in academic social
278 networks? What features are available?

B. OVERVIEW OF MVCWALKER

280 The MVCWalker collaborator recommendation model is
281 inspired by the truth that scholars usually desire to co-operate

282 with people who have high academic value. Such people nor-
283 mally have fruitful high-quality papers, which can generally
284 be used to represent their academic achievements. Besides,
285 as the RWR model has been proved to be competent for
286 calculating the similarity of nodes in network, we use it as
287 a basic model for the coauthor social network. Furthermore,
288 we introduce edge attributes information into the network
289 structure to bias the random walk such that it will more easily
290 traverse to the positive nodes.

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

$$297 MR(p_i) = \frac{1 - \alpha}{N} + \alpha \sum_{p_j \in M(p_i)} \frac{MR(p_j)}{L(p_j)} \quad (1)$$

298 where MR represents the rank score vector, and $MR(p_i)$ is the
299 rank score of node p_i , which is the quantized importance of
300 node p_i to the target node. $M(p_i)$ is the set of nodes incident
301 to node p_i , with $L(p_j)$ being the number of all the neighbors
302 of node p_j . α denotes the probability that the walker will con-
303 tinue to walk to the next neighbor. Above all, in MVCWalker
304 model, the walker has the probability to randomly skip to
305 any other nodes. Equation (1) represents only the step to get
306 the rank score of a node. With respect to each node in the
307 whole graph, the personalized random walk process is defined
308 by (2), which is an iterative process.

$$309 MR^{(t+1)} = \alpha S MR^{(t)} + (1 - \alpha)q \quad (2)$$

310 where S is the transfer matrix, representing the probability for
311 each node to skip to other nodes. $MR^{(t)}$ represents the rank
312 score vector at step t , and q is the row vector, and its form is
313 $(0, \dots, 1, \dots, 0)$. In fact, at the beginning, $MR^{(0)} = q$, and
314 the rank score of target node is 1, while others' are 0.

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

320 The relevance score defined by MVCWalker has many
321 good properties: as compared with those common neighbor
322 models, it can capture the global structure of a graph; while
323 compared with those traditional short distance models, it can
324 capture the multi-facet relationship between two nodes [33].

325 Basic random walk models usually assume that the weights
326 of edges are the same [12], and define the cells of matrix S
327 as $S_{i,j} = \frac{1}{L(p_j)}$. In contrast, here we define matrix S by link
328 importance based on academic factors. We will introduce the
329 link importance in Section 3.3.

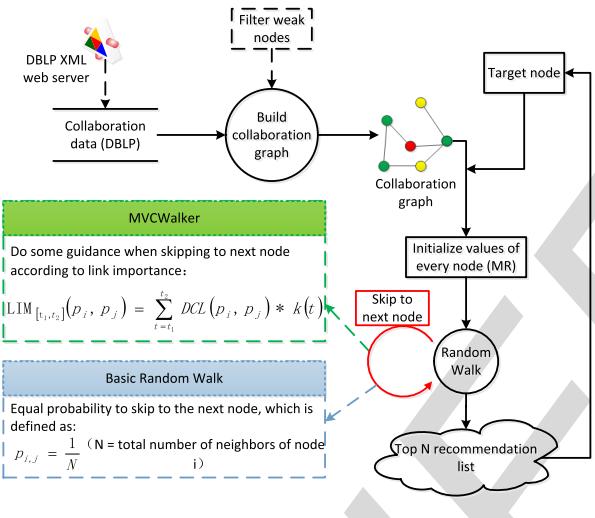
330 The detailed process of MVCWalker is described below
331 and the corresponding pseudocode is shown in Algorithm 1.
332 The structure of MVCWalker is illustrated in Fig. 2.

- 333 • The initial input data of MVCWalker is a set of several
334 years' papers published by many scholars. To extract the

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

```

1:  $S \leftarrow \text{ComputeTransferMatrix}()$ 
2:  $MR_0 \leftarrow R$ 
3:  $Q \leftarrow R$ 
4: for  $k \leftarrow 0$  to  $\text{MaxIteration} - 1$  do
5:    $diff \leftarrow 0$ 
6:   for  $i \leftarrow 0$  to  $\text{len}(Q) - 1$  do
7:      $MR_{k_i} = \alpha \sum_{j=0}^{\text{len}(Q)} S_{i,j} MR_j + (1 - \alpha) Q_i$ 
8:      $diff \leftarrow diff + (MR_k - MR_{k-1})$ 
9:   end for
10:  if  $diff < \text{MinDelta}$  then
11:    break
12:  end if
13: end for
14: Predictions  $\leftarrow \text{predictions}(MR)$ 
15: return Predictions
  
```

**FIGURE 2. Structure of MVCWalker.**

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 $LIM(P_i, p_j)$.
- 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. S is the set of probabilities for each P_i in G skipping to next node P_j . This can be described

as $S_{i,j} = \frac{LIM(P_i, p_j)}{\sum_{P_k \in M(P_i)} LIM(P_i, p_k)}$, while $M(P_i)$ 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 $(0, \dots, 1, \dots, 0)$. 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 $MR(p_k)$. 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.

We present below the details of how the link importance is computed by taking into account the three academic factors.

C. 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 .

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).

408 Consider two nodes p_i, p_j in a coauthor list. Assume that
 409 $j > i$, and there are more than one author of a paper. For the
 410 sake of simplicity, we calculate $DCL(p_i, p_j)$ as follows.

$$411 DCL(p_i, p_j) = \begin{cases} \frac{1}{i} + \frac{1}{j} & j \leq 3 \\ \frac{1}{i} + \frac{2}{j} & j > 3, \quad i \leq 3 \\ \frac{2}{i} + \frac{2}{j} & i > 3 \end{cases} \quad (3)$$

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

417 2) LATEST COLLABORATION TIME POINT

418 In recent years, more and more social recommendation
 419 systems introduced time dependence models (see [34]). Especially,
 420 academic social networks are obvious time-varying, where the links among scholars change over time. For
 421 instance, scholars might be more willing to collaborate with
 422 whom coauthored a paper last year, as compared to the
 423 co-authors of ten years ago. Hence, we measure the link
 424 dynamics using $LIM_t(p_i, p_j)$ (i.e. Link Importance):
 425

$$426 LIM_t(p_i, p_j) = DCL(p_i, p_j) * k(t) \quad (4)$$

427 To reflect the dynamic feature of coauthor-ship, we introduce
 428 the equation $k(t)$, a monotonically increasing function over
 429 time. We can measure the impact of different coauthoring
 430 time points by adjusting the parameter $k(t)$. Here, we define
 431 $k(t)$ as:

$$432 k(t) = \frac{t_i - t_0}{t_c - t_0} \quad (5)$$

433 where t_i is the published time of the paper i two researchers
 434 coauthored (in year here), t_c is the current time (i.e. 2014 in
 435 this paper) and $t_0 = \text{Published year of the first coauthored}$
 436 $\text{paper} - 1$. In fact, this is a process of normalization.
 437 The experimental results in section 4 demonstrated its
 438 effectiveness.

439 3) TIMES OF COLLABORATION

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

$$446 LIM_{[t_1, t_2]}(p_i, p_j) = \sum_{t=t_1}^{t_2} LIM_t(p_i, p_j) \\ = \sum_{t=t_1}^{t_2} DCL(p_i, p_j) * k(t) \quad (6)$$

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

TABLE 2. Seven papers by five researchers.

Paper ID	Authors	Published year
1	Alice, Bob, Charlie, David	2009
2	Bob, Eden	2010
3	Alice, Charlie, David	2013
4	Eden, David, Bob	2012
5	Bob, David, Alice, Charlie	2013
6	Charlie, Bob, Alice	2014
7	Bob, Eden, David	2011

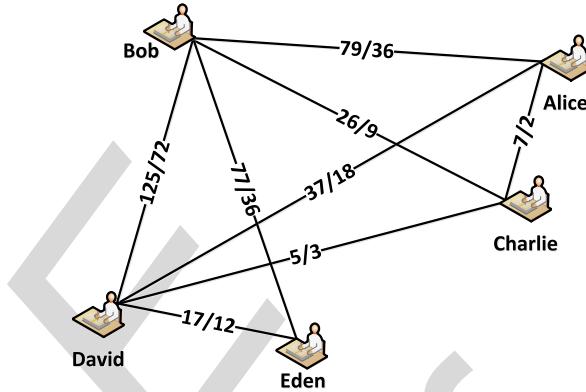


FIGURE 3. Coauthor network with link importance over these five researchers.

451 D. AN EXAMPLE

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

463 Assume that we make some academic cooperation
 464 recommendations for Eden. We know that she has two
 465 co-authors now, Bob and David. Regarding Eden as a random
 466 walker with initial MR value 1 (others 0), when deciding
 467 next jump, she has two equiprobable choices in basic RWR
 468 model, i.e. 0.5 probability to Bob and 0.5 probability to
 469 David. In MVCWalker model, we utilize a guidance pro-
 470 cedure for this decision making process. She has probability of
 471 0.602 (i.e. $\frac{77/36}{17/12+77/36}$) jumping to the node Bob, and 0.398
 472 (i.e. $\frac{17/12}{17/12+77/36}$) to node David. The result of our experiment
 473 has proved the effectiveness of the guidance work. After
 474 limited times of iteration (19 times in this case), each of the
 475 nodes have their own MR values which represent the
 476 importance to Eden. In this case, The MR values of Bob,
 477 Charlie, David, and Alice are 0.500, 0.391, 0.384, 0.376
 478 respectively. Hence we know that Bob is the most suitable

partners in this circle, and Charlie can be recommended to Eden as a new cooperator (even though Eden has never worked with Charlie).

IV. EVALUATION AND ANALYSIS

We conducted extensive experiments using data from DBLP [35]. 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, except for 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 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.

A. DATA SET

DBLP indexes more than 2.3 million articles on computer science and contains many links to home pages of computer scientists. Generally, each DBLP record contains these attributes, authors, title, pages, years, crossref, (in)proceedings or journals, etc. According to the methods provided by [35], 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 [28] 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. The data sets contain 59659 nodes (authors) and 90282 edges

TABLE 3. Statistics of data set from DBLP.

Statistics	Nodes	Edges	Average Degree
Number	59659	90282	1.513

(coauthor relations). In additionally, the average degree is 1.513, which indicate that these data sets are very sparse.

TABLE 4. Possible results of recommendation.

	Recommended	Unrecommended
Collaborated	A	B
No collaborated	C	D

B. METRICS

We chose three popular metrics, precision, recall rate and coverage rate, to evaluate the performance of MVCWalker [36], [37]. Usually, the output of a recommendation system including MVCWalker model is a recommendation list. After some time, there will be a new list of co-authors 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.

The metric precision is defined as:

$$P = \frac{A}{(A + C)} \quad (7)$$

The metric recall rate is defined as:

$$R = \frac{A}{(A + B)} \quad (8)$$

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.

$$c = \frac{\sum d}{n} \quad (9)$$

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.

C. IMPACT OF VARIOUS PARAMETERS

In this section, we examine the impact of different experimental parameter settings, including range of target nodes'

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

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.

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.

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 divided into two subsets (Training Set and Test Set) by the year of publication, based on the concept of split [38]. 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.

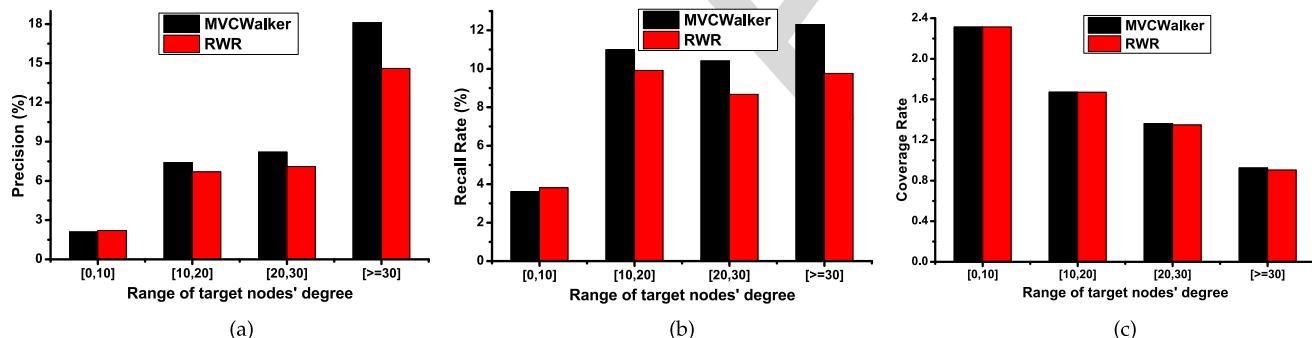


FIGURE 4. Performance of MVCWalker and Basic RWR over Target Nodes' Degree. (a) Precision. (b) Recall Rate. (c) Coverage Rate.

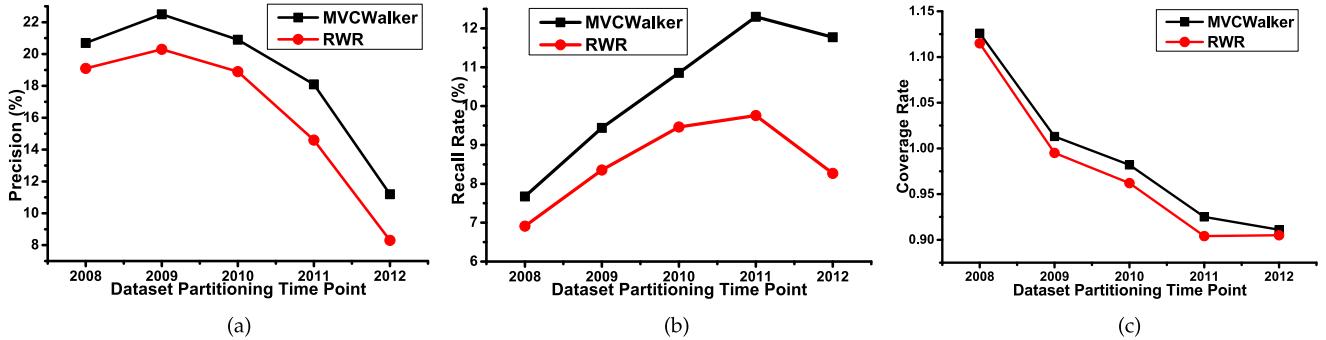


FIGURE 5. Performance of MVCWalker and Basic RWR over Dataset Partitioning Time Point. (a) Precision. (b) Recall Rate. (c) Coverage Rate.

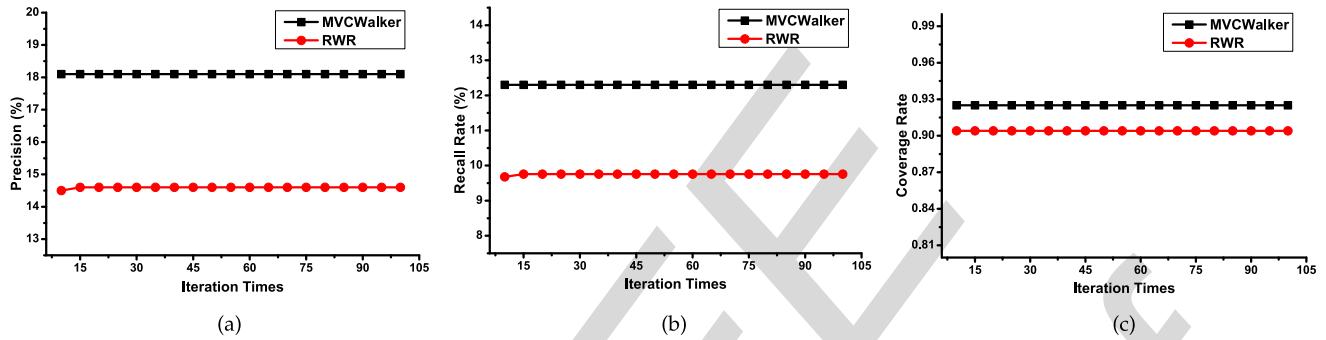


FIGURE 6. Performance of MVCWalker and Basic RWR over Iteration Times. (a) Precision. (b) Recall Rate. (c) Coverage Rate.

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 this is that the academic social network extracted from the data set enlarges in scale and tightens its topology, making it faster to find a collaborator when the partitioning time point increases. Fig. 5(c) validates our thought that there is a trade off between precision and coverage.

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) 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 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

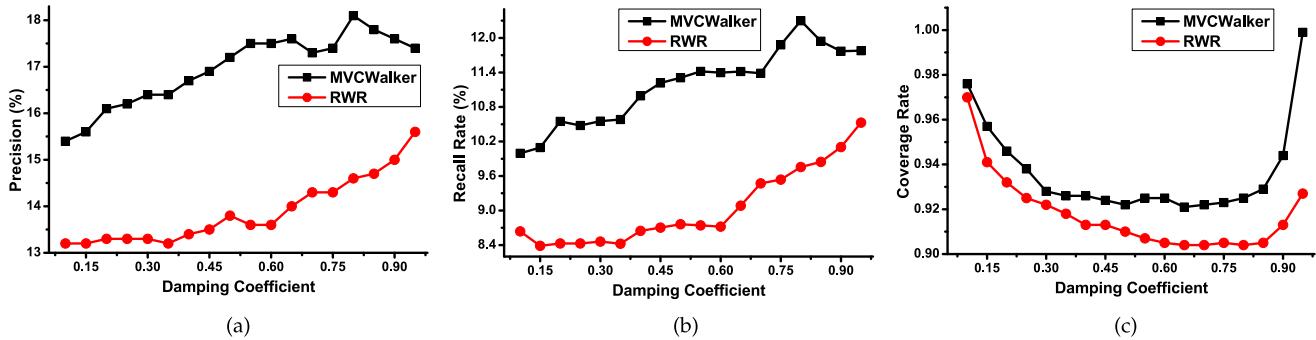


FIGURE 7. Performance of MVCWalker and Basic RWR over Damping Coefficient. (a) Precision. (b) Recall Rate. (c) Coverage Rate.

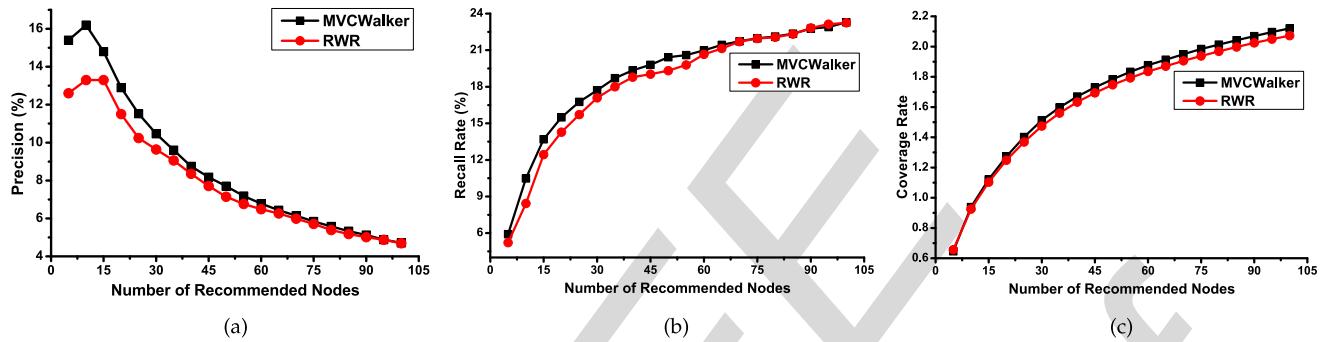


FIGURE 8. Performance of MVCWalker and Basic RWR over Number of Recommended Nodes. (a) Precision. (b) Recall Rate. (c) Coverage Rate.

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.

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 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.

D. COMPARISON WITH OTHER METHODS

As can be seen from 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

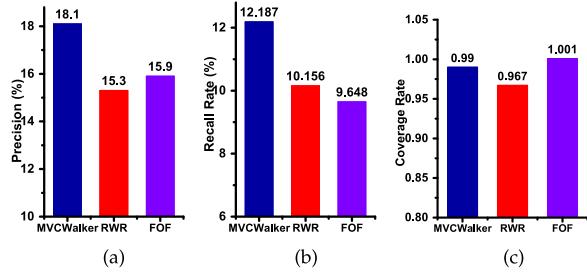


FIGURE 9. Comparison of MVCWalker, RWR and FOF.
(a) Precision. (b) Recall Rate. (c) Coverage Rate.

account affect the performance in diverse manners and we have found their best values for MVCWalker.

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 as compared to FOF. But for most recommendation system, this disadvantage in time complexity is acceptable and does not much affect the recommendation 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.

V. 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 have 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 could be conducted.

REFERENCES

- [1] J. S. Katz and B. R. Martin, "What is research collaboration?" *Res. Policy*, vol. 26, no. 1, pp. 1–18, 1997.
- [2] S. Lee and B. Bozeman, "The impact of research collaboration on scientific productivity," *Soc. Stud. Sci.*, 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 *Proc. 11th Annu. Int. ACM/IEEE Joint Conf. Digit. Libraries (JCDL)*, Ottawa, ON, Canada, Jun. 2011, pp. 231–240.
- [4] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "ArnetMiner: Extraction and mining of academic social networks," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Las Vegas, NV, USA, 2008, pp. 990–998.
- [5] J. Tang, S. Wu, J. Sun, and H. Su, "Cross-domain collaboration recommendation," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Beijing, China, 2012, pp. 1285–1293.
- [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*. Berlin, Germany: Springer-Verlag, 2010, pp. 190–199.
- [7] J. Herlocker, J. A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," *Inf. Retr.*, vol. 5, no. 4, pp. 287–310, 2002.
- [8] A. Tötscher, M. Jahner, and R. Legenstein, "Improved neighborhood-based algorithms for large-scale recommender systems," in *Proc. 2nd KDD Workshop Large-Scale Recommender Syst. Netflix Prize Competition (NETFLIX)*, Las Vegas, NV, USA, 2008, pp. 4:1–4:6.
- [9] D. M. Boyd and N. B. Ellison, "Social network sites: Definition, history, and scholarship," *J. Comput.-Mediated Commun.*, vol. 13, no. 1, pp. 210–230, 2007.
- [10] 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 *Proc. 23rd Int. Conf. World Wide Web Companion (WWW Companion)*, Seoul, Korea, 2014, Art. ID 4.
- [11] L. Backstrom and J. Leskovec, "Supervised random walks: Predicting and recommending links in social networks," in *Proc. 4th ACM Int. Conf. Web Search Data Mining (WSDM)*, Hong Kong, 2011, pp. 635–644.
- [12] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *Proc. 6th Int. Conf. Data Mining*, 2006, pp. 613–622.
- [13] A.-L. Barabási and J. Frangos, *Linked: The New Science of Networks*. New York, NY, USA: Basic Books, 2002.

- [14] J. Freyne, S. Berkovsky, E. M. Daly, and W. Geyer, "Social networking feeds: Recommending items of interest," in *Proc. 4th ACM Conf. Recommender Syst. (RecSys)*, 2010, pp. 277–280.
- [15] J. He and W. W. Chu, "A social network-based recommender system (SNRS)," in *Data Mining for Social Network Data* (Annals of Information Systems), vol. 12. New York, NY, USA: Springer-Verlag, 2010, pp. 47–74.
- [16] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. 4th ACM Int. Conf. Web Search Data Mining (WSDM)*, Hong Kong, 2011, pp. 287–296.
- [17] S. Perugini, M. A. Gonçalves, and E. A. Fox, "Recommender systems research: A connection-centric survey," *J. Intell. Inf. Syst.*, vol. 23, no. 2, pp. 107–143, 2004.
- [18] A. Chin, B. Xu, H. Wang, and X. Wang, "Linking people through physical proximity in a conference," in *Proc. 3rd Int. Workshop Model. Soc. Media*, 2012, pp. 13–20.
- [19] D. H. Lee, P. Brusilovsky, and T. Schleyer, "Recommending collaborators using social features and MeSH terms," in *Proc. Amer. Soc. Inf. Sci. Technol.*, 2011, vol. 48, no. 1, pp. 1–10.
- [20] M. Pavlov and R. Ichise, "Finding experts by link prediction in co-authorship networks," in *Proc. Finding Experts Web Semantics Workshop (FEWS)*, vol. 290. 2007, pp. 42–55.
- [21] W. Glänzel and A. Schubert, "Analysing scientific networks through co-authorship," in *Handbook of Quantitative Science and Technology Research*. New York, NY, USA: Springer-Verlag, 2005, pp. 257–276.
- [22] A. L. Barabási, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek, "Evolution of the social network of scientific collaborations," *Phys. A, Statist. Mech. Appl.*, vol. 311, nos. 3–4, pp. 590–614, 2002.
- [23] M. E. J. Newman, "Scientific collaboration networks. I. Network construction and fundamental results," *Phys. Rev. E*, vol. 64, p. 016131, Jun. 2001.
- [24] X. Liu, J. Bollen, M. L. Nelson, and H. Van de Sompel, "Co-authorship networks in the digital library research community," *Inf. Process. Manage.*, vol. 41, no. 6, pp. 1462–1480, 2005.
- [25] W. Liu and L. Lü, "Link prediction based on local random walk," *Europhys. Lett.*, vol. 89, no. 5, p. 58007, 2010.
- [26] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 58, no. 7, pp. 1019–1031, 2007.
- [27] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Washington, DC, USA, 2010, pp. 243–252.
- [28] M. A. Brandão, M. M. Moro, G. R. Lopes, and J. P. M. Oliveira, "Using link semantics to recommend collaborations in academic social networks," in *Proc. 22nd Int. Conf. World Wide Web Companion (WWW Companion)*, 2013, pp. 833–840.
- [29] M. Jamali and M. Ester, "TrustWalker: A random walk model for combining trust-based and item-based recommendation," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Paris, France, 2009, pp. 397–406.
- [30] 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 Trans. Knowl. Data Eng.*, vol. 19, no. 3, pp. 355–369, Mar. 2007.
- [31] H. Tong, C. Faloutsos, and J.-Y. Pan, "Random walk with restart: Fast solutions and applications," *Knowl. Inf. Syst.*, vol. 14, no. 3, pp. 327–346, 2008.
- [32] I. Konstas, V. Stathopoulos, and J. M. Jose, "On social networks and collaborative recommendation," in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2009, pp. 195–202.
- [33] H. Tong and C. Faloutsos, "Center-piece subgraphs: Problem definition and fast solutions," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Philadelphia, PA, USA, 2006, pp. 404–413.
- [34] A. Töscher, M. Jahrer, and R. M. Bell, "The BigChaos solution to the netflix grand prize," *Netflix Prize Document.*, to be published.
- [35] M. Ley, "DBLP: Some lessons learned," *Proc. VLDB Endowment*, vol. 2, no. 2, pp. 1493–1500, 2009.
- [36] F. Fouss and M. Saerens, "Evaluating performance of recommender systems: An experimental comparison," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell., Intell. Agent Technol. (WI-IAT)*, vol. 1. Sydney, Australia, Dec. 2008, pp. 735–738.
- [37] G. Shani and A. Gunawardana, "Evaluating recommendation systems," in *Recommender Systems Handbook*. New York, NY, USA: Springer-Verlag, 2011, pp. 257–297.
- [38] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval: The Concepts and Technology Behind Search*, 2 ed. Reading, MA, USA: Addison-Wesley, 2011.



FENG XIA (M'07–SM'12) received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2001 and 2006, respectively. He is currently an Associate Professor and a Ph.D. Supervisor with the School of Software, Dalian University of Technology, Dalian, China. He is a Guest Editor of several international journals. He serves as the General Chair, PC Chair, Workshop Chair, Publicity Chair, or a PC Member of a number of conferences. He has authored or co-authored one book and over 160 scientific papers in international journals and conferences. His research interests include social computing, mobile computing, and cyber-physical systems. He is a member of the Association for Computing Machinery.



ZHEN CHEN received the B.S. degree from the Dalian University of Technology, Dalian, China, in 2013, where he is currently pursuing the M.S. degree in software engineering. His research interests include recommender systems, big scholarly data, and smart computing systems.



WEI WANG received the B.S. degree in electronic information science and technology from Shenyang University, Shenyang, China, in 2012. He is currently pursuing the Ph.D. degree in software engineering with the Dalian University of Technology, Dalian, China. His research interests include big scholarly data, social networks, and computational social science.



JING LI received the B.E. degree from the Dalian University of Technology, Dalian, China, in 2011, where she is currently pursuing the master's degree with the School of Software. Her research interests include recommender systems, social computing, and smart meeting.



LAURENCE T. YANG received the B.E. degree in computer science and technology from Tsinghua University, Beijing, China, and the Ph.D. degree in computer science from the University of Victoria, Victoria, BC, Canada. He is currently a Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, and the Department of Computer Science, St. Francis Xavier University, Antigonish, NS, Canada. His

research interests include parallel and distributed computing, and embedded and ubiquitous/pervasive computing. His research has been supported by the National Sciences and Engineering Research Council, and the Canada Foundation for Innovation.

AUTHOR QUERIES

AQ:1 = Please note that references [6] and [10] are the same, hence we deleted Ref. [10] and renumbered the other references. This change will also reflect in the citations present in the body text. Please confirm.

AQ:2 = Please provide the volume no., issue no., page range, month, and year for ref. [34].

IEEE Proof