Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 509282, 8 pages
http://dx.doi.org/10.1155/2014/509282

Research Article

Link Prediction in Directed Network and Its Application in Microblog

Yan Yu[1,2] and Xinxin Wang[2]

[1] *College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*
[2] *Computer Science Department, Southeast University Chenxian College, Nanjing 210088, China*

Correspondence should be addressed to Yan Yu; yuyanyuyan2004@126.com

Received 11 September 2013; Revised 1 December 2013; Accepted 24 December 2013; Published 16 January 2014

Academic Editor: Marek Lefik

Link prediction tries to infer the likelihood of the existence of a link between two nodes in a network. It has important theoretical and practical value. To date, many link prediction algorithms have been proposed. However, most of these studies assumed that links of network are undirected. In this paper, we focus on link prediction in directed networks. We provide an efficient and effective link prediction method, which consists of three steps as follows: (1) we locate the similar nodes of a target node; (2) we identify candidates that the similar nodes link to; and (3) we rank candidates using weighing schemes. We conduct experiments to evaluate the accuracy of our proposed method using real microblog data. The experimental results show that the proposed method is promising.

1. Introduction

Many complex systems, such as social, information, and biological systems, can be modeled as networks, where nodes correspond to individuals or agents, and links represent the relations or interactions between two nodes. Network is a useful tool in analyzing a wide range of complex systems. Many efforts have been made to understand the structure, evolution, and function of networks. Recently, the study of link prediction in network has attracted increasing attention. Link prediction tries to infer the likelihood of the existence of a link between two nodes, which has important theoretical and practical value. In theory, research on link prediction can help us understand the mechanism of evolution of complex network. In practical application, link prediction can be applied to many practical fields. For example, link prediction can be used in biological network to infer existence of a link so as to save experimental cost and time. It also can be utilized to online social networks to recommend friends for users, so as to improve the users' experience.

To date, many link prediction algorithms have been proposed. Most of them are designed based on node similarity. The higher the similarity score between two nodes, the higher the possibility of them being connected. In order to measure the node similarity, many link prediction algorithms exploit network structure [1]. One reason is that links in a network indicate certain similarity between the nodes they connect. According to the domain of required structure of network, there are two main kinds of approaches in the domain of link prediction. The first one is based on local features of a network, detecting mainly the local nodes' structure; the second one is based on global features of a network, focusing on the overall structure of a network [1].

However, most studies of link prediction assumed that links of network are undirected. In fact, examples of directed networks are numerous in real world: the web is made up of directed hyperlinks, the food webs consist of directed links from predators to preys, and in the microblog, followers form links to their opinion leaders. One unique aspect of directed network is the asymmetric nature of links. Modeling links as directed networks introduce complexity but offer significant analytical benefits [2]. To the best of our knowledge, quantitative approaches in directed networks are few.

In this paper, we focus on link prediction in directed networks. We provide an efficient and effective link prediction method in directed network. We conduct experiment to evaluate the accuracy of proposed method using real microblog data.

The rest of this paper is organized as follows. Section 2 introduces some related work. Section 3 describes a new link prediction method in directed network. Section 4 presents the experimental setup and we present the results of evaluation in Section 5. Section 6 concludes this paper.

2. Related Work

Link prediction focuses on inferring the likelihood of the existence of a link between two nodes in a network in terms of observed links and attributes of nodes in a network. Link prediction can predict missing links or the links that may exist in the near future in a network. To date, many link prediction algorithms have been proposed, most of which are based on the node similarity [1]. Rationale behind them is the principle of homophily, that is, the "similarity breeds connection" [2]. The higher the similarity score between two nodes, the higher the possibility of them being connected. In order to measure the node similarity, many link prediction algorithms exploit network structure [1], because the topology of network can indicate certain similarity between the nodes [2].

According to the domain of required network structure, there are two main kinds of approaches in the domain of link prediction. The first one is based on local features of a network, detecting mainly the local nodes' structure; the second one is based on global features of a network, focusing on the overall structure of a network [1]. For example, Common Neighbour [3], Adamic-Adar [4], Resource Allocation [5], FriendLink [6], and PropFlow [7] are local ones, which consider the local neighborhood information; Rooted PageRank [8], SimRank [9], and Random Walk with Restart [8] are global ones, which consider the whole structure of a network.

In this paper, we mainly focus on the local-based algorithms. There are two reasons. First, global-based algorithms require more time and space than local-based ones. Some networks, such as microblog, contain hundreds of millions of nodes. This implies that algorithm needs to be applicable to network with millions of nodes. Second, Papadimitriou et al. found that some local-based algorithms outperform some global-based algorithms because global-based methods traverse the network globally, missing to capture adequately the local characteristics of the network [6]. Therefore, we introduce some typical local-based algorithms in the following.

Common Neighbor (CN) [3] measures the similarity of two nodes in the network. Intuitively, two nodes are more likely to have a link if they have many common neighbors. Because of its simplicity, many online social network such as Facebook use CN to recommend people to connect with. Adamic-Adar (AA) [4] refines the simple counting of common neighbors by assigning the lower connected neighbors more weights. Resource Allocation (RA) [5] refines the CN index, which is closely related to the resource allocation process. It weighs common neighbor by inverse of its degree. Considering a pair of nodes, u and v, which are not directly connected, the node u can send some resource to v, with their common neighbors playing the role of transmitters. Each transmitter has a unit a resource, and averagely distribute to all its neighbors. As a result the amount of resource v receives is defined as the similarity between u and v. FriendLink [6] defines a node similarity of two nodes by traversing all paths of a limited length based on the algorithmic small world hypothesis. By traversing all possible paths between a person and all other nodes in network, a node can be connected to another by many possible paths. Nodes in network can use all the pathways connecting them, proportionally to the pathway length. Thus, two nodes which are connected with many unique pathways have a high possibility to know each other, proportionally to the length of the pathways they are connected with. PropFlow [7] corresponds to the probability that a restricted random walk starting at node u ends at v in l steps. The restrictions are that the walk terminates upon reaching v or upon revisiting any node including u. This produces a score that can serve as an estimation of the similarity of two nodes. PropFlow is somewhat similar to Rooted PageRank, but it is a more localized measure of propagation, and is insensitive to topological noise far from the source node. Unlike Rooted PageRank, the computation of PropFlow does not require walk restarts or convergence but simply employs a modified breadth-first search restricted to height l. It is thus much faster to compute.

Most existing methods of link prediction assume that these links in network are undirected. However, examples of directed networks are numerous: the web is made up of directed hyperlinks, the food webs consist of directed links from predators to preys, and users form links to their opinion leaders in microblog. Modeling links as directed networks introduce complexity but offer significant analytical benefits [2]. When a link is symmetric, there are only two states: the link is present or absent. When links are asymmetric, there are four states between two nodes: node u links to node v, v links to u, u and v are mutually connected, or the absence of a link between u and v. If there exists a directed link from u to v, we might say that v has a power or status advantage over u, since v is more important to u than u is to v [2]. The directed link is an indicator of the direction in which attention flows. To the best of our knowledge, quantitative approaches in directed networks are few.

To fill this gap, we focus on link prediction in directed networks in this paper. We propose link prediction method, which can provide efficient and effective link prediction in directed network. We conduct experiment to evaluate the accuracy of proposed method using real-world microblog data.

3. The Proposed Method

In this section, we propose a link prediction method in directed network. The idea of the proposed method is that a node tends to link to the nodes which its similar nodes link to. So, for a given node, the method we present consists of three steps: (1) we locate similar nodes of a target node; (2) we identify candidates that the similar nodes link to; and (3) we rank candidates using weighing schemes.

To describe the proposed method, we construct a directed graph $G(V, E)$, where V represents a set of nodes in directed network and E represents a set of links among these nodes.

A directed link $\langle u, v\rangle \in E$ exists between nodes u and v if u links to v. The set of out neighbors of node u is $\Gamma_{\text{out}}(u) = \{v \in V \mid (u, v) \in E\}$, and the out-degree of u is $|\Gamma_{\text{out}}(u)|$, where $|\cdot|$ denotes the size of the set. Similarly, $\Gamma_{\text{in}}(u)$, $\Gamma_{\text{in}}(u) = \{v \in V \mid (v, u) \in E\}$, represents the set of in neighbors of u and in-degree of u is $|\Gamma_{\text{in}}(u)|$. The input to our problem is the directed network G and a target node u. Our task is to predict the likelihood of the existence of the link from u to other unlinked nodes in terms of observed topology of the directed network. In the remaining subsections, we, respectively, provide detailed descriptions of these three steps that essentially constitute the proposed method.

3.1. Finding Three Categories of Similar Nodes. Nodes that have certain common interests are simply called similar nodes. In this subsection, we explore three categories of similar nodes with a target node. Assuming that a target user is u, three categories of similar nodes with u are termed below as $S_1(u)$, $S_2(u)$, and $S_3(u)$.

Finding $S_1(u)$ is based on a fact that u has already identified some similar nodes, which are its current successors. For example, in Figure 1(a), target node u_1 has a link to u_2; we can presume that u_2 is a similar node with u_1. Thus, we define $S_1(u)$ as a set of the first category of similar nodes of a target node u. Mathematically, we have the following equation to define the set:

$$S_1(u) = \Gamma_{\text{out}}(u). \tag{1}$$

According to (1), we have $S_1(u) = \{u_2\}$, for example, in Figure 1(a).

Finding $S_2(u)$ can be done by extending the scope of u's out neighbors from 1-hop out neighborhood to 2-hop out neighborhood. For example, in Figure 1(a), as u_1 follows u_2, u_2 follows u_3, and we can presume that u_3 is also a similar node with u_1. In general, when taking into account the contribution of longer paths, more nodes that are similar to a target node can be included. By doing so, we can overcome the limitation of overlocalization of the first category of similar nodes. Studies show that 2-hop neighborhood based method outperforms many other methods including longer path or global network based approaches [6]. Therefore, we define $S_2(u)$ as a set of the second category of similar nodes of a target user u. Mathematically, we have the following equation to define the set:

$$S_2(u) = \bigcup_{v \in \Gamma_{\text{out}}(u)} \Gamma_{\text{out}}(v) - \{u\}. \tag{2}$$

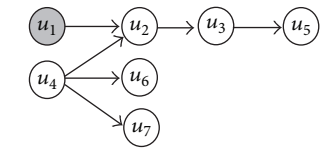
According to (2), we have $S_2(u_1) = \{u_3\}$, for example, in Figure 1(a).

The third category of similar nodes we explore is based on homophily principle of shared interests [2]. In network, directly shared interests can be represented by $u \rightarrow k \leftarrow v$, where node u and node v each links to node k [2]. u and v sharing interests is surely one kind of similarity. For example, in Figure 1(a), u_1 and u_4 each links to u_2. We can presume that u_4 is similar to u_1. Like $S_1(u)$ and $S_2(u)$, therefore, we define $S_3(u)$ as a set of the third category of similar users of a target user u. Mathematically, we have the following equation to define the set:

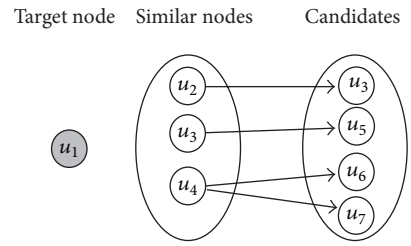
$$S_3(u) = \bigcup_{v \in \Gamma_{\text{out}}(u)} \Gamma_{\text{in}}(v) - \{u\}. \tag{3}$$

In Figure 1(a), we can easily find $S_3(u_1) = \{u_4\}$.

Once we find all three categories of similar nodes, we aggregate all of them as the similar nodes of a target node; that is, $S(u) = S_1(u) \cup S_2(u) \cup S_3(u)$. In Figure 1(a), $S(u_1) = \{u_2\} \cup \{u_3\} \cup \{u_4\} = \{u_2, u_3, u_4\}$.



(a) An example of directed network



(b) Similar nodes and candidates of u_1 in (a)

Figure 1: Example of proposed method.

3.2. Identifying Candidates. After we find the list of similar nodes $S(u)$ for a target node u, we can identify a list of candidates $C(u)$. The rationale behind this step is that a target node u may like to link to nodes that its similar nodes link to. Of course, we should exclude the users that u has already followed. Mathematically, we have the following equation to define the candidates:

$$C(u) = \bigcup_{v \in S(u)} \Gamma_{\text{out}}(v) - \Gamma_{\text{out}}(u). \tag{4}$$

For example, we found $S(u_1) = \{u_2, u_3, u_4\}$ in Figure 1(a). u_2 links to u_3, u_3 links to u_5, and u_4 link to u_2, u_6, and u_7. We can then identify all the candidates for u_1; that is, $C(u_1) = \{u_3\} \cup \{u_5\} \cup \{u_2, u_6, u_7\} - \{u_2\} = \{u_3, u_5, u_6, u_7\}$ as shown in Figure 1(b). Note that u_3 is both similar node and candidate of u_1.

3.3. Ranking Candidates. After we identify a list of candidates of a target user, we rank the candidates using scores in a descending order. We take a unified weighting approach to ranking identified candidates for a target node. Specifically, we evaluate each candidate through a voting process. Each similar node $s \in S(u)$ essentially casts a vote; each vote is weighted by applying $w(u, c, s)$ to each candidate $c \in C(u)$. The total score of a candidate c for the target node u is the sum

of $w(u,c,s)$ for all $s \in S(u)$. We define our unified ranking algorithm as follows:

$$\begin{aligned}\text{score}(u,c) = \alpha \times \frac{\sum_{s\in S_1(u)} w(u,c,s)}{|S_1(u)|} \\ + \beta \times \frac{\sum_{s\in S_2(u)} w(u,c,s)}{|S_2(u)|} \\ + \gamma \times \frac{\sum_{s\in S_3(u)} w(u,c,s)}{|S_3(u)|},\end{aligned} \tag{5}$$

where α, β, and γ are topological structural weights, $\alpha+\beta+\gamma = 1$. If we choose $\alpha = 1$, $\beta = 0$, and $\gamma = 0$, we only consider the votes from the first category of similar nodes. If we choose $\alpha = 0$, $\beta = 1$, and $\gamma = 0$, we only consider the votes from the second category of similar users. Then if we choose $\alpha = 0$, $\beta = 0$, and $\gamma = 1$, we only consider the votes from the third category of similar users. For a practically optimal outcome, α, β, and γ should be determined in real time as they vary with settings and objectives. In addition, each vote carries a weight or value that varies with adopted voting schemes or strategies.

In this paper, we explore three different voting schemes or strategies, termed as V_1 strategy, V_{ra} strategy, and V_{sim} strategy, to compute $w(u,c,s)$. These three strategies are explained in details below.

As shown in (6), V_1 strategy computes a score of a similar nodes $s \in S(u)$ for each candidate $c \in C(u)$, if s follows c:

$$w_{V_1}(u,c,s) = \begin{cases} 1 & c \in \left(C(u) \cap \Gamma_{\text{out}}(s)\right) \wedge s \in S(u) \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

For the link predictionof undirected network, researchers provided many metrics. Studies show that the metrics weighting the contribution of common neighbors by inverse of its degree [5] better predict new links. Therefore, as shown in (7), V_{ra} strategy weights the similar node by applying the inverse of its out-degree:

$$w_{V_{\text{ra}}}(u,c,s) = \begin{cases} \dfrac{1}{|\Gamma_{\text{out}}(s)|} & c \in \left(C(u) \cap \Gamma_{\text{out}}(s)\right) \wedge s \in S(u) \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

V_{sim} strategy is then based on the idea of shared interests. If two nodes both link to the same node, then two nodes may have more shared interests. Therefore, V_{sim} strategy weights a candidate by calculating Pearson's Correlation Coefficients between a target node and a similar node according to overlap of their out neighbors. Consider

$$w_{V_{\text{sim}}}(u,c,s) = \begin{cases} \dfrac{|\Gamma_{\text{out}}(u) \cap \Gamma_{\text{out}}(s)|}{|\Gamma_{\text{out}}(u)| \cdot |\Gamma_{\text{out}}(s)|} & c \in \left(C(u) \cap \Gamma_{\text{out}}(s)\right) \wedge s \in S(u) \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

In summary, our proposed method can form three different approaches: V_1, V_{ra}, and V_{sim} by applying a different voting scheme. In the following two sections, we apply our proposed method to the problem of user recommendation in microblog and evaluate the accuracy of the proposed method.

4. Experimental Setup

To evaluate the accuracy of the proposed method, we apply it to the problem of user recommendation in microblog. In this section, we describe the experimental setup and provide the optimal parameters. Section 5 presents the results of experimental evaluation.

4.1. Dataset and Experiment Setup. Microblog, such as Twitter and Google+, has become tremendously popular in recent years, which attracts hundreds of millions of users. This scale benefits the microblog users but it can also flood users with huge volumes of information and hence puts them at risk of information overload. User recommendation in microblog can reduce the risk of information overload and improve the user experience. User recommendation task involves predicting whether or not a user will follow another user. Microblog is essentially an information platform on which users form an explicit social network by following other users [10]. Thus, user, that is, follower, automatically receives the messages posted by the users he/she follows, known as followees. In microblog, users and follower/followee relationships constitute a directed network, which we call follower/followee network. Therefore, we apply our proposed method for user recommendation in microblog to evaluate the accuracy of the method.

In this paper, we use a real-world dataset from Tencent Weibo. Tencent Weibo, one of the largest microblog websites in China, has become a major platform for building friendship and sharing interests online. Since its launch in April 2010, Currently, there are more than 200 million registered users on Tencent Weibo, generating over 40 million messages each day. The dataset we use for experiment in this paper is the KDD Cup 2012 dataset from the follower prediction task. The dataset contains 2,320,895 users with 50,655,143 following relations and provides rich information in multiple domains such as user profiles and item category.

In this paper, we focus on exploiting following information of users. We make the snapshot of users' following information on October 11, 2011 as a training set S. We take records of following history from 10/11/2011 to 11/11/2011 as the validation set V. We then use records of following history from 11/11/2011 to 30/11/2011 as the test set T. In the experiment, we first use our method on the training set S, use the validation set V to get the optimal parameters α, β, and γ, and then apply our proposed method with optimal parameters to the whole data set $S+V$ and get the predictions on the test set T.

It is noteworthy to mention that there are hundreds of millions of users and tens of billions of follower/followee relationships in microblog. For instance, some celebrities have millions of followers. Computing all followers of such

celebrity could be computationally expensive. In this study, we use a random sampling approach to selecting followers of each followee of a target user for a practical implementation.

4.2. Evaluation Metrics. Researchers have used precision and average precision to evaluate the accuracy of recommendation algorithms for years. Precision measures the average percentage of the overlap between a given recommendation list and the list of followees that are actually followed. Precision can be evaluated at different points in a ranked list of recommended users. Mathematically, precision at rank k ($P@k$) is defined as the proportion of relevant users and recommended users:

$$P@k = \frac{\text{number of relevant users with rank } k}{k}. \quad (9)$$

Average precision (AP), which the KDD cup 2012's organizers adopted, emphasizes the ranking relevant users higher. That is, it is better to have a correct guess in the first place of the recommendation list. It is the average of precisions computed at the point of each of the relevant users in the ranked list:

$$\text{AP}@k = \frac{\sum_{i=1}^{k} (P@i \times \text{rel}(i))}{\text{number of relevant users with } k}, \quad (10)$$

where rel(i) is the change in the recall from $i-1$ to i. MAP@k is the mean value of AP@k.

However, we think it makes more sense to consider the number and the ranking of relevant users, simultaneously. In other words, we simply replace "the number of relevant users with k" with "k" and call it as AP$'$@k.

Let us use an example to illustrate the difference of applying different evaluation metrics. Assume that there are three algorithms of recommending top 3 followees for a target user u_1. Table 2 shows the recommended followees and ones that were actually followed. Algorithm 1 and Algorithm 2 have the same $P@3$, because the number of relevant users is the same. However, we intuitively think Algorithm 2 has relatively better accuracy performance than Algorithm 1, because Algorithm 2 has a correct guess in the first ranking and the second ranking. Meanwhile, Algorithm 2 and Algorithm 3 have the same AP@3. Intuitively, we know that Algorithm 3 should be better than Algorithm 2 as Algorithm 3 recommended more relevant users.

Table 1 indicates that our proposed evaluation metrics can be more accurate than others. Thus, we adopt this new evaluation metrics, AP$'$@k, which is mathematically defined as follows:

$$\text{AP}'@k = \frac{\sum_{i=1}^{k} (P@i \times \text{rel}(i))}{k}. \quad (11)$$

Likewise, MAP$'$@k is the mean value of AP$'$@k of all target users. In the reported experiments, we evaluate MAP$'$@k for values of k equal to 1, 3, 5, and 10.

4.3. Parameters Setting. As discussed earlier, there are three parameters, that is, α, β, and γ, that should be determined in the proposed method. We carry out a parameter-sweep approach to maximize the accuracy in terms of MAP$'$@k. Our experiments show when $\alpha = 0.15$, $\beta = 0.1$, and $\gamma = 0.75$, the performance of proposed approaches is optimal. Table 2 presents the performance of three recommendation approaches with the optimal parameters.

In the remaining part of this paper, we basically apply the above-determined parameters to evaluate the performance of the proposed method. We explicitly state other parameter settings when needed.

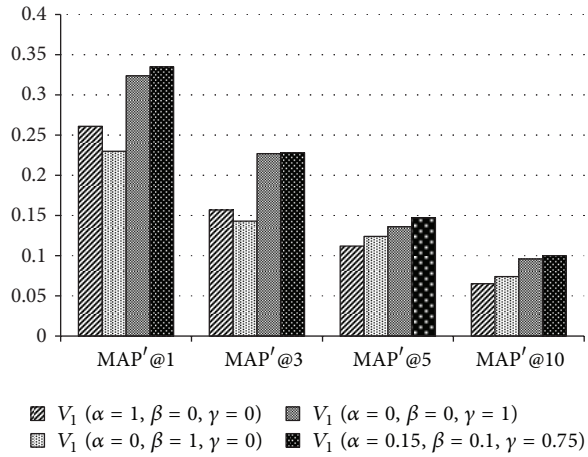


Figure 2: Evaluation of aggregating three categories of similar nodes on V_1 algorithm.

5. Experimental Results

In this section, we present the results of our experimental evaluation. More specially, in Section 5.1, we show how different aggregating approaches to finding similar users differ. In Section 5.2, we examine the performance of three different voting strategies in ranking candidates. Finally in Section 5.3, we report the results by comparing our proposed method with some existing methods.

5.1. Aggregation of Three Categories of Similar Nodes. In this section, we compare the performances of different aggregating approaches that might be adopted in the process of ranking candidates. As discussed earlier, the values of α, β, and γ define how the voices of the similar users of a target user could be aggregated in the voting process. Figures 2, 3, and 4, respectively, show the results for different aggregating approaches to identifying candidates by different groups of similar users while different voting strategies are applied. When $\alpha = 1$, $\beta = 0$, and $\gamma = 0$, the aggregating approach essentially considers the votes by similar users defined by S_1. When $\alpha = 0$, $\beta = 1$, and $\gamma = 0$, it only considers the votes by the similar users defined by S_2. When $\alpha = 0$, $\beta = 0$, and $\gamma = 1$, it then simply considers the votes by the similar users defined by S_3. Note that when $\alpha = 0.15$, $\beta = 0.1$, and $\gamma = 0.75$, it becomes a true aggregation of all the votes by similar users under consideration.

Table 1: Differences of applying different evaluation metrics.

Algorithm	Target user	Recommended user	Accepted user	P@3	AP@3	AP′@3
Algorithm 1	u_1	u_2 u_4 u_3	u_2 u_3	2/3 = 0.667	(1/1 + 2/3)/2 = 0.8333	(1/1 + 2/3)/3 = 0.556
Algorithm 2	u_1	u_2 u_3 u_4	u_2 u_3 	2/3 = 0.667	(1/1 + 2/2)/2 = 1.000	(1/1 + 2/2)/3 = 0.667
Algorithm 3	u_1	u_2 u_3 u_5	u_2 u_3 u_5	3/3 = 1.000	(1/1 + 2/2 + 3/3)/3 = 1.000	(1/1 + 2/2 + 3/3)/3 = 1.000

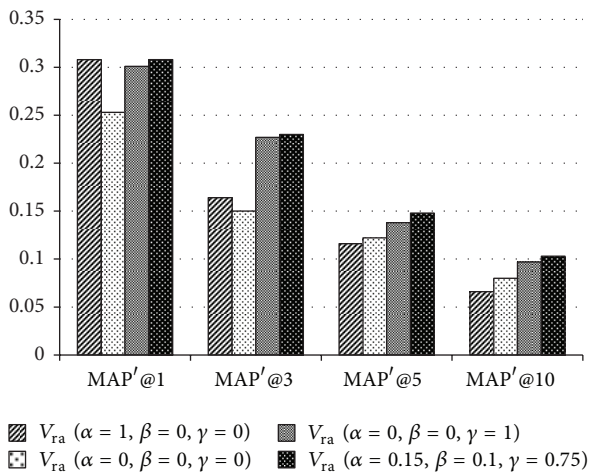


Figure 3: Evaluation of aggregating three categories of similar nodes on V_{ra} algorithm.

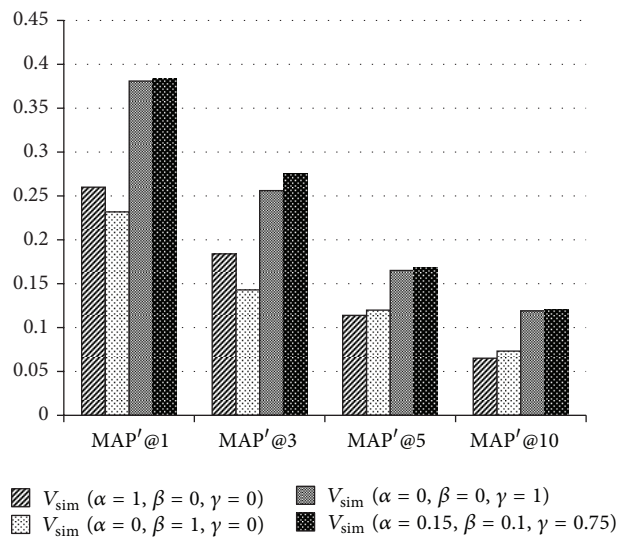


Figure 4: Evaluation of aggregating three categories of similar nodes on V_{sim} algorithm.

Table 2: Performance of strategies with optimal parameters setting.

	MAP′@1	MAP′@3	MAP′@5	MAP′@10
V_1	0.303	0.209	0.145	0.073
V_{ra}	0.317	0.221	0.149	0.078
V_{sim}	0.325	0.218	0.150	0.081

Table 3: Result of comparison of methods.

Method	MAP′@1	MAP′@3	MAP′@5	MAP′@10
CN	0.261	0.157	0.112	0.065
AA	0.282	0.164	0.109	0.066
RA	0.308	0.164	0.116	0.066
FriendLink	0.253	0.179	0.117	0.065
PropFlow	0.279	0.172	0.124	0.074
V_{sim}	**0.384**	**0.276**	**0.168**	**0.121**

The bold numbers represent the result of our proposed method.

We compare the performances of these three extreme scenarios to the approach of simultaneously aggregating the votes from three kinds of similar users based on the determined optimal parameters. Regardless of adopted voting strategies, the results show that the proposed aggregating approach generally outperforms approaches of considering only votes from one kind of similar users.

5.2. Voting Strategies. In this section, we evaluate the performance of the proposed three voting strategies. By comparing the performance of V_1, V_{ra}, and V_{sim}, we have Figure 5. The results in Figure 5 show that V_{sim} outperforms other two strategies in all evaluation ranking metrics. This indicates that when weighing the candidate scores of a target user, it is beneficial by considering the out-degree similarity between the target and intermediate users. The intersection of followees of two users indicates their interests' similarity to some extend. In other words, recommendation from more similar users with more common interests thus can improve the effectiveness of recommendation.

5.3. Comparison with Other Methods. Finally, we compare our method with some existing local-based link prediction

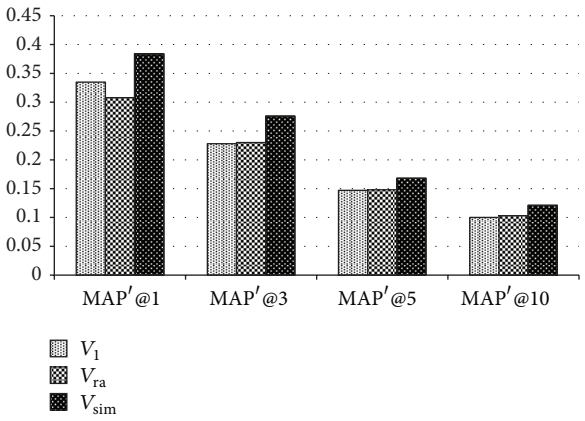


FIGURE 5: Evaluation of three voting strategies.

methods. We first present basic information of the methods that will be compared with our method. Then we report the results of comparison.

CN [3]. This algorithm is based on the intuition that two nodes are more likely to have a link if they have many common neighbors. In our work, we define the CN index of node u and node v as

$$S_{uv}^{\mathrm{CN}} = \left|\Gamma_{\mathrm{out}}(u) \cap \Gamma_{\mathrm{in}}(v)\right|. \tag{12}$$

AA [4]. This algorithm refines the simple counting of common neighbors by assigning the lower connected neighbors more weights. In the directed network, we define it as

$$S_{uv}^{\mathrm{AA}} = \sum_{z \in \Gamma_{\mathrm{out}}(u) \cap \Gamma_{\mathrm{in}}(v)} \frac{1}{\log\left(\left|\Gamma_{\mathrm{out}}(z)\right| + \varepsilon\right)}, \tag{13}$$

where ε is a very small number to avoid denominator to be zero.

RA [5]. RA refines the CN index, which weighs common neighbor by inverse of its degree. In the directed network, we define it as

$$S_{uv}^{\mathrm{RA}} = \sum_{z \in \Gamma_{\mathrm{out}}(u) \cap \Gamma_{\mathrm{in}}(v)} \frac{1}{\left|\Gamma_{\mathrm{out}}(z)\right|}. \tag{14}$$

FriendLink [6]. FriendLink defines node similarity of two nodes by traversing all paths of a limited length, which is defined as

$$S_{uv}^{\mathrm{FriendLink}} = \sum_{t=2}^{l} \frac{1}{t-1} \cdot \frac{\left|\mathrm{path}_{uv}^{t}\right|}{\prod_{k=2}^{t}(n-k)}, \tag{15}$$

where n is the number of nodes in a network, l is the maximum length of a path taken into consideration between the nodes u and v, $1/(t-1)$ is an attenuation factor that weights paths according to their length l, $|\mathrm{paths}_{uv}^{t}|$ is number of all length-l paths from u to v, and $\prod_{k=2}^{t}(n-k)$ is the number of all possible length-l paths from u to v if each node in network is linked with all other nodes.

PropFlow [7]. PropFlow corresponds to the probability that a restricted random walk starting at node u ends at v in l steps. The restrictions are that the walk terminates upon reaching v or upon revisiting any node including u. This produces a score $S_{uv}^{\mathrm{PropFlow}}$ that can serve as an estimation of the similarity of two nodes.

We use the training set S and validate set T to compute the similarity of two nodes according to above-mentioned methods, and then use the test data T to assess the accuracy of these methods. The results are then compared with the performance of applying our proposed method using the V_{sim} voting strategy. The comparisons are stated in Table 3. As shown in Table 3, our proposed V_{sim} clearly provides a more accurate recommendation than other methods, which indicates that our proposed method is effective in user recommendation in microblog. First, aggregating three categories of similar nodes with different weights is effective because they contain more useful information to recommend followees that a target may be interested in. Second, considering similarity of similar users and target user can improve the accuracy performance.

6. Conclusion

Link prediction has important theoretical and practical value. Recently, many link prediction algorithms have been proposed. However, most studies of link prediction assumed that links of network are undirected. In this paper, we focus on link prediction in directed networks, which provide efficient and effective link prediction in directed network. The method we present consists of three steps as follows: (1) we locate the similar nodes of a target node; (2) we identify candidates that the similar nodes link to; and (3) we rank candidates using weighing schemes. We conduct experiment in microblog to evaluate the accuracy of proposed algorithm by using real microblog data. The experimental results show that the proposed approach is promising, which indicates that our proposed method is effective in user recommendation in microblog. First, aggregating three categories of similar nodes with different weights is effective because they contain more useful information to recommend followees that a target may be interested in. Second, considering similarity of similar users and target user can improve the accuracy performance. In light of our future study, we would like to explore an efficient and effective method to determine the required parameters, and we are planning to include other directed networks to carry out more experiment.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research is supported by the Research Foundation of Jiangsu Institute of Modern Educational Technology (no. 2012-R-22749) and the Education philosophy and Social Science Fund Project of Jiangsu Province (no. 2013SJD880063) and is sponsored by Jiangsu's Qing Lan Project.

References

[1] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[2] S. A. Golder and S. Yardi, "Structural predictors of tie formation in twitter: transitivity and mutuality," in *Proceedings of the 2nd IEEE International Conference on Social Computing (SocialCom '10)*, pp. 88–95, Minneapolis, Minn, USA, August 2010.

[3] F. Lorrain and H. C. White, "Structural equivalence of individuals in social networks," *The Journal of Mathematical Sociology*, vol. 1, no. 1, pp. 49–80, 1971.

[4] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.

[5] T. Zhou, J. Ren, M. Medo, and Y. C. Zhang, "Bipartite network projection and personal recommendation," *Physical Review E*, vol. 76, no. 4, Article ID 046115, 7 pages, 2007.

[6] A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos, "Fast and accurate link prediction in social networking systems," *Journal of Systems and Software*, vol. 85, no. 9, pp. 2119–2132, 2012.

[7] 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)*, pp. 243–252, July 2010.

[8] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks*, vol. 30, no. 1–7, pp. 107–117, 1998.

[9] G. Jeh and J. Widom, "SimRank: a measure of structural-context similarity," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*, pp. 538–543, July 2002.

[10] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang, and R. Zadeh, "Wtf: the who to follow service at twitter," in *Proceedings of the 22nd International Conference on World Wide Web*, pp. 505–514.