Information Sciences 376 (2017) 158–171

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Projection-based link prediction in a bipartite network

Man Gao[a], Ling Chen[a,b,*], Bin Li[a,b], Yun Li[a], Wei Liu[a], Yong-cheng Xu[a]

[a] Department of Computer Science, Yangzhou University, Yangzhou, 225009, China
[b] State Key Lab of Novel Software Tech, Nanjing University, Nanjing, 210093, China

ARTICLE INFO

Article history:
Received 6 December 2015
Revised 14 June 2016
Accepted 6 October 2016
Available online 11 October 2016

Keywords:
Bipartite network
Link prediction
Projection
Candidate node pairs

ABSTRACT

An algorithm for link prediction in a bipartite network is presented. In the algorithm, we first map the bipartite network onto a unipartite network called a projected graph. Based on the projected graph, we define the concept of a candidate node pair (CNP). We perform link prediction only within the CNPs to reduce the computation time. We also define the patterns covered by the CNPs and weights of the patterns. By calculating the weights of the patterns that a CNP covers, the connectivity of the CNP can be obtained, which can be used as the final score of link prediction. For a bipartite network with n and m nodes in the two parts, the time complexity of the proposed algorithm is $O(m)$, whereas those of other algorithms are $O(mn)$ or $O((m+n)^3)$. The experimental results show that our algorithm can achieve higher speed and superior quality link prediction results in bipartite networks compared with other methods.

1. Introduction

Networks can naturally describe various social structures. In such networks, vertices denote entities, and links represent communications or relations between the entities. A social network reflects persons or social organizations and their relations such as partnership or friendship. The analysis of social networking has drawn increasing attention in the field of sociology. It analyzes and explores the potential relations between social objects. In recent years, social network analysis has also attracted high interest in many business fields, such as e-business analysis and market modeling.

Link prediction is an important area in research on network analysis [41]. The objective of link prediction is to detect unobserved links from existing topologic features of the network or forecast future links from the current topologic structure of the network. In a social security network, link prediction is employed to discover underground groups of terrorists or criminals [28], whereas in networks of human behavior, link prediction is used to identify and categorize the activities and movement of people [1]. Link prediction also has many applications in networks reflecting social relations such as communication networks, email networks and sensor networks. In sensor networks, link prediction is used to discover dynamic temporal properties [44], to ensure information transfer secrecy [25], and to realize the most favorable routing [17].

Because relations among social members change continuously over time, links in real-world social networks are constantly varying and evolving. New links may appear, and existing links may vanish from the network. For example, email communications between friends, transactions between businesses, partnership between scientific researchers are changing over time. Therefore, link prediction algorithms should be capable of detecting dynamic relationships between members in

* Corresponding author. Fax: +8651487887937.
E-mail address: yzulchen@163.com (L. Chen).

http://dx.doi.org/10.1016/j.ins.2016.10.015
0020-0255/

a temporal social network. Recently, various approaches have been proposed to detect potential or future links in temporal social networks.

The similarity-based method is the most commonly used method for link prediction. In this method, each node pair is associated with an index to indicate the similarity between the corresponding nodes. This similarity quantifies the likelihood of link existence in the graph. Some essential attributes of the nodes can be used to define their similarity, such as the existence of many common features or topological structures between the nodes. Many studies in social networks show the existence of a relative similarity between individuals who are close to each other [2,11]. The structural similarity indices are often used in popular similarity-based methods. There are three types of similarity indices: local, quasi-local and global indices. Local indices are based on the neighbor information of the nodes, such as Common Neighbors, Jaccard, Salton, Sorensen, Preferential Attachment, Hub Depressed, Hub Promoted, Adamic-Adar, Resource Allocation, and Leicht–Holme–Newman (LHN1) indices [28]. Global indices require comprehensive information for link prediction tasks. They use global topological information of networks; such indices include Katz, Matrix Forest Index (MFI), and Leicht–Holme–Newman (LHN2) [28]. Quasi-local indices require less structural information than global indices but more information than local indices. Such indices include Superposed Random Walk (SRW) [27], Local Random Walk (LRW), and Local Path Index [48]. Compared with the local indices, global indices can achieve a more accurate solution of link prediction and require more computation time. Another class of similarity-based methods is random walk methods. Those methods include SimRank, Random Walk with Restart, Cos+, and Average Commute Time [28].

Some of these methods are based on the analysis of the topological features of the network. Purnamrita et al. [33] presented a nonparametric algorithm to predict potential links in dynamic networks. This algorithm partitioned the time domain into subsequences, which are represented by graph snapshots. Their method predicts connections between the nodes based on their topological features and local neighbors. For link prediction in social networks, they advanced a weighted-proximity-based method. Kim et al. [18] presented a method to predict future network topology using node centrality, which can identify important nodes in the future.

Machine learning strategies have also been exploited in network link prediction methods. Pujari et al. [32] applied a supervised rank aggregation method for link prediction in complex networks. Vu et al. [40] introduced a continuous-time regression model for temporal network link prediction. This model can be incorporated with both time-varying regression coefficients and time-dependent network statistics. Zeng et al. [45] presented a method using semi-supervised learning in link prediction utilize the latent information in the unlinked node pairs in networks. He et al. [15] proposed an algorithm for a link prediction ensemble based on an ordered weighted averaging operator. The algorithm assigns weights for nine local information-based link prediction algorithms and then aggregates their results to obtain the final prediction scores. Bao et al. [4] advanced a network link predictor using principal component analysis to identify features that are important to link prediction. Bringmann et al. [8] presented an algorithm for link prediction in networks based on techniques of association-rule mining and frequent-pattern detection. Using techniques for data mining and machine learning, the method can predict future co-participation of individuals in social events. To avoid high computational cost of optimization in the machine learning methods, some heuristic methods are employed in link prediction. Sherkat et al. [38] introduced an ant-colony-optimization-based algorithm for link prediction. Catherine et al. [7] proposed an approach to predict future links by applying the covariance matrix adaptation evolution strategy.

Some methods for link prediction on a network are based on probabilistic models. Hanneke et al. [14] proposed a family of statistical models for dynamic social network link prediction by extending the exponential random graph model. Liu et al. [26] presented a method for predicting potential links in a user-object bipartite network. Their method considers both the time attenuation and diversion delay. Gao et al. [12] advanced a model that exploits multiple information sources in the dynamic network to obtain link occurrence probabilities. Barbieri et al. [5] presented a stochastic link prediction model on directed graphs with node attribute features. In addition to predicting links, the model also provides explanations for the links detected. F. Hu et al. [16] presented a probabilistic model to detect human motion in social networks and advanced a method for labeling human motion using a constraint-based genetic algorithm to optimize the model. However, such a probabilistic model requires a predefined distribution of link appearance, which is difficult to know in advance for a given network.

The bipartite network is an important type of complex networks in real world applications, in which the nodes are partitioned into two parts such that no two nodes of the same part are adjacent. In a bipartite network, all edges connect only nodes in different parts, and no nodes of the same part are linked. Many real social networks are logically bipartite networks, such as the scientists–papers cooperation network [23,34], disease–gene network [10,47,20,19], RNA–protein interaction network [13], club members–activities network [36], term–document network [21], and investors–company network [39]. In recent years, link prediction in bipartite networks is applied in areas such as recommendation [9,31,42], social network analysis [6], and drug side effect prediction [29].

In recent years, many methods have been proposed to predict potential links in bipartite networks [6,9,22,24,29,43,37,46]. Supervised learning strategies have been exploited in bipartite network link prediction methods. Chang [9] et al. applied a supervised learning approach to predict potential links in the bipartite network representing Wikipedia. Benchettara et al. [6] introduce new topological features to quantify the probability of the link connecting the node pair in a bipartite network. Treating link prediction as a problem of binary classification, their approach uses a supervised machine learning approach to learn prediction models. However, those methods require a predefined distribution of link appearance, which is difficult to know in advance for a given network. By studying the structural holes in bipartite social networks, Xia et al. [43] presented

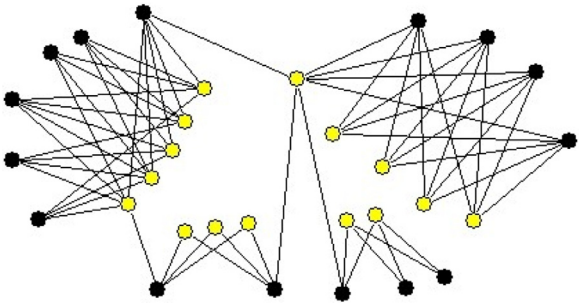

Fig. 1. The actor–film bipartite network.

a method to predict links between different nodes. Some kernel-based methods are also proposed for bipartite network link prediction. Li et al. [24] designed a graph kernel in a bipartite network that inspects two types of nodes to predict whether there may be a link. To apply the link prediction methods for unipartite networks to bipartite ones, Kunegis et al. [22] generalized a class of spectral transformation graph kernels to bipartite graphs using the odd component of the spectral transformation. However, a large amount of time to is required to build and process such spectral transformation graph kernels. Allali et al. [3] defined the concept of internal links in a bipartite graph and presented an approach for link prediction based on internal links. In their approach, all non-existing links are tested to find the internal links and their weights. The internal links with weights greater than a threshold are considered to be the potential links. Because the approach must enumerate all non-existing links, it requires a large amount of computation time. In addition, it must predefine a proper threshold to obtain accurate results. However, it is difficult to set such a proper threshold in advance for a given dataset.

In this paper, we present an algorithm for link prediction in a bipartite network. In the algorithm, we first map the bipartite network onto a unipartite one called a projected graph. We define the concept of candidate node pair (CNP) on the projected graph and perform the link prediction only within the CNPs. The connectivity of the CNP is calculated based on the weights of the patterns it covers. Such connectivity of the CNP is used as the final score of link prediction.

The rest of this paper is organized as follows: Section 2 defines the projected graph for a bipartite network. Section 3 presents the concepts of candidate node pairs and their reliabilities in the projected graph, and Section 4 gives the frameworks of our algorithm for link prediction in a bipartite network. In Section 5, we present and analyze the experimental results of our link prediction methods on real datasets. Finally, Section 6 gives our conclusions.

2. Bipartite graphs and weighted projection

A bipartite network can be represented by a bipartite graph $G=(U, V, E)$, where U and V are two parts of the nodes in G. In this paper, we use uppercase letters, such as B, C, to denote the U-part nodes and lowercase letters, such as x, y, to denote the V-part notes. E is the set of edges in G. There is no edge between the nodes in the same set of U or V; namely, every edge $(B, x) \in E$ satisfies $B \in U$, and $x \in V$. Assuming that U has n nodes and V has m nodes, the adjacency matrix of G takes the following block off-diagonal form:

$$A = \begin{bmatrix} 0_{n\times n} & A_{n\times m} \\ A^T_{m\times n} & 0_{m\times m} \end{bmatrix}.$$

where $0_{n\times n}$ and $0_{m\times m}$ are $n\times n$ and $m\times m$ all-zero matrixes, respectively, and $A_{n\times m}$ is an $n\times m$ nonzero matrix. Because the adjacency matrix is symmetric, we can simply use matrix $A_{n\times m}$ to represent bipartite graph G, where each row represents a node in the set of U, and each column represents a node in the set of V.

Fig. 1 shows the actor–film bipartite network. In the network, there are two types of nodes, which are colored in black and yellow. The black nodes represent the actors, and the yellow nodes represent the films. We use $N(B)=\{x|x \in V, (B, x) \in E\}$ to denote the set of neighbors of node B in G.

To analyze the potential links in the bipartite network, we first map it onto a unipartite one called a projected graph.

Definition 1. (projected graph). Given a bipartite graph $G=(U, V, E)$, its U-projected graph is defined as a unipartite graph $G_u=(U, E_u)$, where the set of edges is

$$E_u = \{(B, C)|B, C \in U, \exists x \in V, x \in N(B) \cap N(C)\} \quad (1))$$

From the definition, we can see that if U-part nodes B and C in the bipartite network G have at least one common neighbor in the V-part, then there exists a link (B, C) in the U-projected graph G_u. Similarly, we can define the V-projected graph $G_v=(U, E_v)$ of G.

For instance, we denote the set of upper and lower part nodes of the bipartite graph G in Fig. 2(a) as U and V, respectively. The U-projected and V-projected graphs of G are then as shown in Fig. 2(b) and (c), respectively.

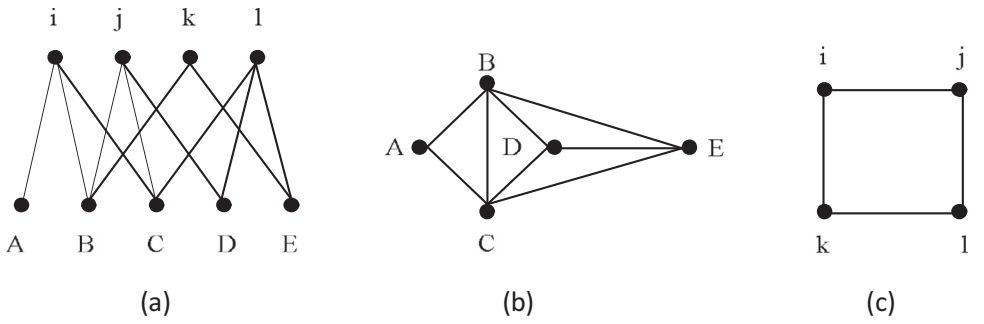

Fig. 2. Bipartite graph and its projected graphs.

3. Candidate node pairs of the patterns

To predict the possible links in the bipartite network G, we first define the candidate node pairs based on its projected graph.

Definition 2. (Candidate Node Pair by U-projection). Let $G=(U, V, E)$ be a bipartite graph, $B \in U$ and $x \in V$ be two nodes in G, and $(B, x) \notin E$. Denote the U-projected graph of G as $G_u=(U, E_u)$. By adding a new link $(B, x) \in U \times V$ to G, we construct a bipartite graph $G'=(U, V, E')$, where $E'=E \cup \{(B, x)\}$. Let $G'_n=(U, E'_u)$ be the U-projected graph of G'. If $G_u=G'_u$, then (B, x) is a candidate node pair (CNP) in graph G by U-projection.

In other words, the candidate node pair is a pair of unlinked nodes $(B, x) \in U \times V$ in G such that adding an edge (B, x) to G makes no change to the U-projected graph of G.

Similarly, we can define a candidate node pair by V-projection.

For instance, the unlinked node pair (B, l) in Fig. 2(a) is a CNP. In fact, all neighbors of l—namely, nodes C, D, and E—are already linked with node B in U-projected graphs G_u shown in Fig. 2(b). Those neighbors of l are also linked with node B in V-projected graph G_v shown in Fig. 2(c). Therefore, adding link (B, l) to G makes no change to its U-projected graph.

Let (B, x) be a pair of nodes in bipartite graph $G=(U, V, E)$, where $B \in U$, $x \in V$ and $(B, x) \notin E$. Denote the U-projected graph of G as $G_u=(U, E_u)$. We define the set of neighbors of node $B \in U$ in G_u as $N_u(B)=\{C | C \in U, (B, C) \in E_u\}$ and the neighbor of node $x \in V$ in G as $N(x)=\{B | B \in U, (B, x) \in E\}$. We first give the following Theorem.

Theorem 1. *A node pair (B, x) in bipartite graph $G=(U, V, E)$ is a CNP by U-projection if and only if it satisfies*

$$N_u(B) \cap N(x) \neq \Phi \text{ and } B \notin N(x) \tag{2}$$

Here, Φ is the empty set.

Proof. (1) Suppose that node pair (B, x) in G satisfies $N_u(B) \cap N(x) \neq \Phi$ and $B \notin N(x)$; let $N_u(B) \cap N(x)=\{C_1, C_2, \ldots, C_k\}$. Because $C_i \in N(x), i=1, \ldots, k$, there exist edges (C_i, x), $i=1, \ldots, k$, in bipartite graph G. Because $C_i \in N_u(B), i=1, \ldots, k$, there also exist edges (B, C_i), $i=1, \ldots, k$ in U-projected graph G_u. If we add an edge (B, x) to G to form a new bipartite graph $G'=(U, V, E')$, where $E'=E \cup \{(B, x)\}$, the U-projected graph of G' can be obtained by adding edges (B, C_i), $i=1, \ldots, k$ on the U-projected graph of G. However, because $(C_i, x) \in E$ and $(B, x) \in E$, edge (B, C_i) has already been in the U-projected graph of G. Therefore, adding an edge (B, x) to G makes no change to the U-projected graph. By Definition 2, we know that (B, x) is a CNP of G by U-projection.

(2) Suppose that (B, x) is a CNP of G by U-projection; by Definition 2, we know that $B \notin N(x)$. Assume that $N_u(B) \cap N(x)=\Phi$. Let the set of nodes in G connected with x be $N(x)=\{C_1, C_2, \ldots, C_k\}$—namely $(C_i, x) \in E$, $i=1, .., k$. Because $N_u(B) \cap N(x)=\Phi$, we know that $C_i \notin N_u(B)$, and there is no edge connecting C_i and B in U-projected graph G_u. If we add a new edge (B, x) to graph G to form a new graph $G'=(U, V, E')$, where $E'=E \cup \{(B, x)\}$, because $(B, x) \in E'$ and $(C_i, x) \in E \subset E'$ in G', there must exist an edge connecting C_i and B in the U-projected graph of G'. However, there is no such edge in the U-projected graph of G, which means that adding an edge (B, x) to G makes changes to the U-projected graph of G. This is contradictory with the fact that (B, x) is a CNP of G by U-projection. Therefore, we have $N_u(B) \cap N(x) \neq \Phi$.

From Theorem 1, we can see that for a CNP (B, x) of G by U-projection, the set of U-part nodes $N_u(u) \cap N(v)$ is not empty. Namely, there must be at least one node in U that is linked with node x in bipartite graph G and is also linked with B in projected graph G_u.

Theorem 2. *Let $G=(U, V, E)$ be a bipartite graph, $B \in U$ and $x \in V$ be two nodes in G, and $(B, x) \notin E$. Node pair (B, x) is a CNP in graph G by V-projection, if and only if it is also a CNP by U-projection.*

Proof. If (B, x) is a CNP in graph G by U-projection, by Theorem 1, we know that $N_u(B) \cap N(x) \neq \Phi$. Suppose that node $C \in N_u(B) \cap N(x)$. Because $C \in N_u(B)$, there is an edge (C, B) in the U-projected graph G_u; namely, there must exist at least one

node y in G such that $(B, y) \in E$ and $(C, y) \in E$. Because $C \in N(x)$, there is an edge $(C, x) \in E$. From the fact that $(C, y) \in E$ and $(C, x) \in E$, we can see that there is an edge (x, y) in the V-projected graph G_v, which means that $y \in N_v(x)$. Because $(B, y) \in E$—namely, $y \in N(B)$—we have $y \in N(B) \cap N_v(x)$, which means that $N_v(x) \cap N(B) \neq \Phi$. Because $(B, x) \notin E$, we have $x \notin N(B)$. Because $N_v(x) \cap N(B) \neq \Phi$ and $x \notin N(B)$, by Theorem 1, we know that (B, x) is a CNP in graph G by V-projection.$f(B, x)$ is a CNP in graph G by V-projection, we can similarly prove that it is also a CNP in graph G by U-projection.

Theorem 2 shows that we must consider the CNP of only one type of projection for link prediction. Therefore, we use the term CNP to represent the candidate node pair by both V-projection and U-projection. To reduce the computation time, our method detects CNPs by U-projection when $|U| < |V|$ and by V-projection when $|U| \geq |V|$. In the following, we suppose that $|U| < |V|$ and illustrate our methods by using CNPs by U-projection.

If (B, x) is a CNP in graph G, for each node C in $N_u(B) \cap N(x)$, there is a link (B, C) in projected graph G_u. Because (B, x) is a CNP in G, we know that $B \notin N(x)$, and there is no edge (B, x) in graph G. If we add an edge (B, x) to bipartite graph G, for every node $C \in N_u(B) \cap N(x)$, a new edge (B, C) should be added to the projected graph G_u because nodes B and C are both linked with node x in G. Noticing that edge (B, C) has already been included in G_u, we call edge (B, C) in projected graph G_u a pattern covered by candidate node pair (B, x).

Definition 3. (pattern). Let B and C be two nodes in the U part of bipartite graph $G=(U, V, E)$. If there exists a node $x \in V$ such that $(B, x) \in E$ and $(C, x) \in E$, then node pair $\{B, C\}$ is called a pattern in G.

From the definition, we can see that if $\{B, C\}$ is a pattern in G, then nodes B and C in the U part have at least one common neighbor in the V part. Therefore, there must exist an edge (B, C) in the projected graph. Conversely, each edge (B, C) in the projected graph corresponds to a pattern $\{B, C\}$ in bipartite graph G. Let (B, x) be a CNP in bipartite graph G and G_u be the projected graph of G. For each node $C \in N_u(B) \cap N(x)$, there must exist an edge (B, C) in the projected graph G_u such that the node pair $\{B, C\}$ is a pattern in bipartite graph G.

Definition 4. (Pattern covered by a CNP). Let (B, x) be a CNP in bipartite graph G and G_u be the projected graph of G. For each node $C \in N_u(B) \cap N(x)$, we call $\{B, C\}$ a pattern covered by CNP (B, x).

A CNP may cover one or more patterns in the projected graph. Such a pattern covered by the CNP indicates that a similar link has already existed in bipartite graph G. The more patterns a CNP covers, the more likely the CNP will be linked in the future. Therefore, the number of patterns covered by a CNP can be used to measure the likelihood of its link existence.

For instance, the bipartite network in Fig. 2(a) represents a customer–item network in a store, in which the bottom nodes represent the items, and the top nodes represent the customers. Link (A, i) in Fig. 2(a) indicates that customer i buys item A. In the figure, customer l buys items C and D simultaneously; this shopping strategy then forms a pattern $\{C, D\}$—namely, an edge (C,D) in the projected graph G_u. In the figure, we can see that item B is not bought by customer l. If we add a new edge (B, l) in the bipartite graph—namely, we assume that customer l buys a new item B—then all patterns generated by (B, l), such as $\{B, C\}$, $\{B, D\}$ and $\{B, E\}$, are identical to those that already exist. This means that this shopping strategy is very similar to those existing ones. Therefore, it is highly probable that customer l buys item B; namely, link (B, l) has a high probability to appear.

Conversely, if a pair of unlinked nodes (B, l) is not a CNP, then $N_u(B) \cap N(l) = \Phi$, and we cannot find any pattern it covers. In other words, there exists no such shopping strategy similar to (B, l), and it is unlikely that nodes B and l are linked. Therefore, we can ignore all non-CNP node pairs and perform the link prediction only in the set of CNPs to narrow the search scope and reduce the computation time for link prediction.

In our link prediction method, each CNP is assigned an index called connectivity, which is a measurement of the appearance likelihood of the link between each pair of nodes. To compute the connectivity of a CNP, we consider the number of patterns it covers.

By Definition 3, we know that the number of patterns covered by a CNP (B, x) is equal to the size of set $N_u(B) \cap N(x)$. The larger the size of $N_u(B) \cap N(x)$, the more patterns are covered by CNP (B, x) and hence the more likely nodes B and x are connected by a link.

For instance, in Fig. 3, node pairs (A, i) and (A, j) are all CNPs, but the number of patterns they cover are different. CNP (A, i) covers only pattern $\{A, B\}$, whereas CNP (A, j) covers patterns $\{A, B\}$, $\{A, C\}$ and $\{A, D\}$. Obviously, the link corresponding to CNP (A, j) has a higher probability to appear than that corresponding to (A, i). Therefore, we use the number of patterns covered by a CNP to calculate its connectivity measuring the likelihood of its corresponding link.

4. Weights of the patterns

When we use the number of patterns covered by a CNP to calculate its connectivity, we should also consider the importance of the patterns. We measure such importance of a pattern by assigning a weight to its corresponding edge in the projected graph. In computing the weight of a pattern $\{A, B\}$, three factors should be considered:

(1) Number of common neighbors of nodes A and B.

A pattern covered by a CNP is represented by an edge in the projected graph. Each edge (A, B) in projected graph G_u indicates that the nodes in pattern $\{A, B\}$ have common neighbors in the bipartite graph G. However, the edge (A, B) in projected graph G_u cannot specify the number of their neighbors in the bipartite graph. For instance, two bipartite graphs in

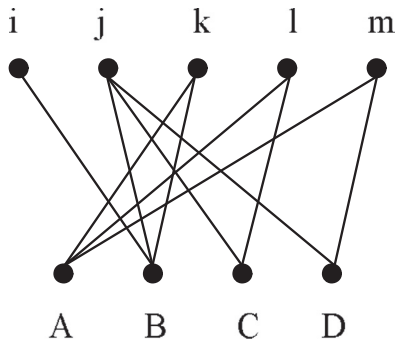

Fig. 3. Candidate node pairs (A, i) and (A, j) in Bipartite graph G.

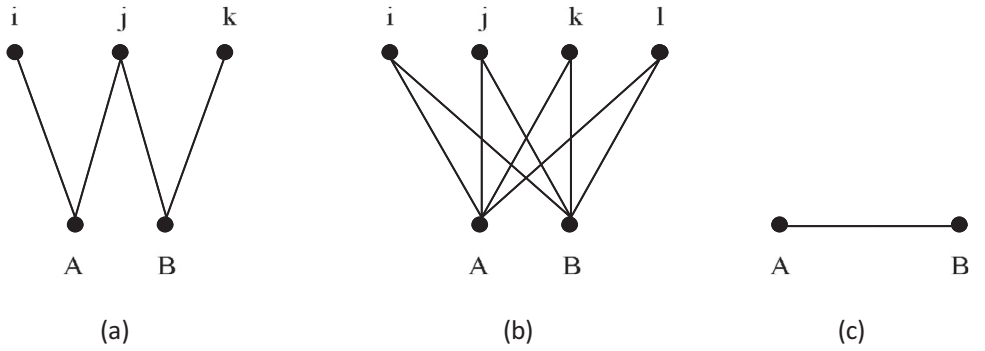

Fig. 4. Two different bipartite graphs and their projected graph.

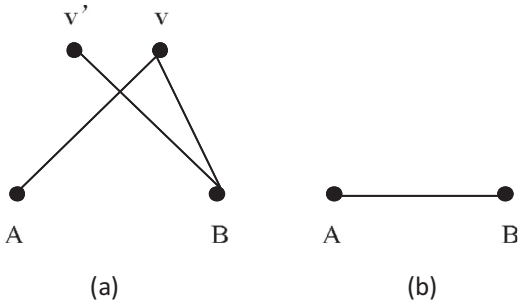

Fig. 5. Candidate node pair (A, v') and its projection.

Fig. 4(a) and (b) are different, but their projected graphs are identical as shown in Fig. 4(c). Therefore, the simple projected graph lost the topological information of the original bipartite graph. To preserve such information, we use a weighted projected graph in which each link (A, B) is assigned a weight according to the number of common neighbors of nodes A and B in the bipartite graph to reflect their similarity. For instance, nodes A and B in bipartite graph 4(a) have only one common neighbor, so its edge in the projected graph should be assigned lower weight. However, nodes A and B in bipartite graph 4(b) have 4 common neighbors, so its edge in the projected graph should be assigned higher weight.

(2) Degree of common neighbors of nodes A and B

For each pattern $\{A, B\}$, the degree of the common neighbors of nodes A and B should also be considered. If the common neighbors of A and B have smaller degrees, then we assign the pattern $\{A, B\}$ higher weight. For instance, the projected graph of the bipartite graphs in Figs. 5(a) and 6(a) are as shown in Figs. 5(b) and 6(b), respectively. (A, v') in the two bipartite graphs are both CNPs, and they both cover the pattern $\{A, B\}$. However, in graph 5(a), the degree of A and B's common neighbor v is 2, whereas that in graph 6(a) is 4. Suppose that A and B are two customers and node v is an item in a department store; then, in bipartite graph 5(a), item v is purchased only buy A and B. *However,* in bipartite graph 6(a), item v is purchased by all 6 customers, and A and B have less shopping similarity because v is a common item. It is obvious that edge (A, B) in 5(b) should have higher weight.

(3) Degrees of nodes A and B in bipartite graph

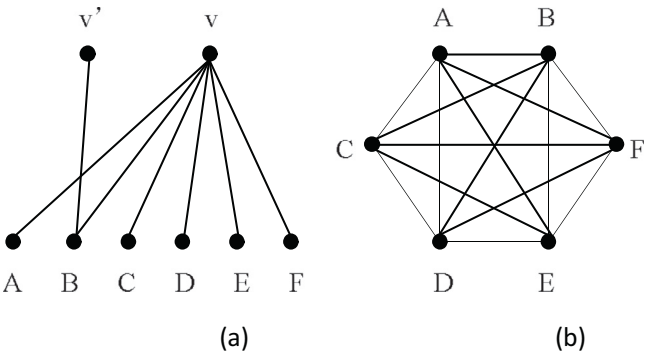

Fig. 6. Candidate node pair (A, v') and its projection.

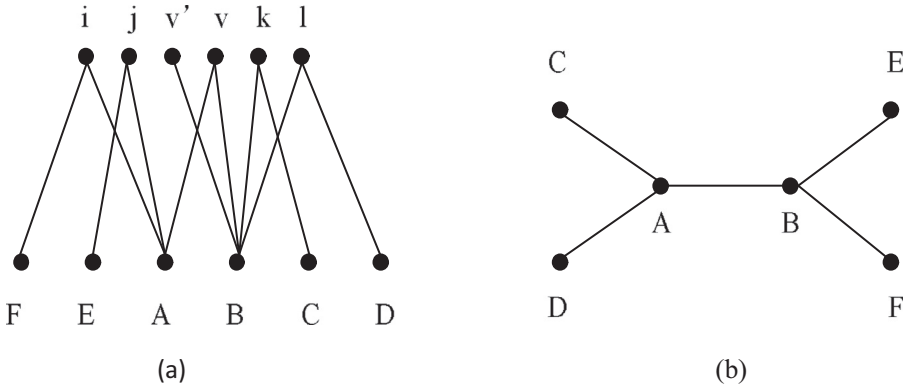

Fig. 7. Candidate node pair (A, v') and its projection.

The degrees of nodes A and B in bipartite graph G also influences the weight of pattern $\{A, B\}$. If the nodes of A and B have smaller degrees in bipartite graph G, then we assign the pattern $\{A, B\}$ higher weight. For example, bipartite graph 5(a) and 7(a) have the same CNP (A, v'). In graph 5(a), the degree of nodes A and B are 1 and 2, respectively. In graph 7(a), the degrees of nodes A and B are 3 and 4, respectively. Suppose that A and B are two customers, and node v represents an item. In bipartite graph 5(a), customer A buys only item v, and customer B buys only items v and v'. However, in bipartite graph 7(a), customers A and B also buy some other items, and item v is the only item they purchased in common. Therefore, customers A and B in graph 7(a) have less shopping similarity, and the weight of pattern $\{A, B\}$ in Fig. 7(b) should be lower than that in Fig. 5(b).

Based on the analysis above, we can define the weight of a pattern as follows:

Definition 5. (Weight of a pattern). Suppose that $G_u=(U, E_u)$ is the projected graph of bipartite graph $G=(U, V, E)$. Let $(A, B) \in E_u$ be an edge in G_u, then the weight of pattern $\{A, B\}$ is defined as

$$w(A,B) = \frac{2}{D(A)+D(B)} \sum_{v \in N(A) \cap N(B)} \frac{1}{D(v)} \tag{3}$$

where $D(A)$, $D(B)$ and $D(v)$ are the degrees of nodes A,B and v, respectively, in bipartite graph G. $N(A)$ and $N(B)$ are the sets of neighbors of A and B, respectively, in bipartite graph G.

From the definition, we can see that edge (A, B) in the projected graph will be assigned higher weight if nodes A and B in the bipartite graph have more common neighbors with lower degrees. We use the weight of edge (A, B) in the projected graph as the weight of pattern $\{A, B\}$. It is clear that a CNP that covers the patterns with higher weight will have higher probability to be connected by a link and should have higher connectivity. Therefore, we define the connectivity of a CNP as follows.

Definition 6. (Connectivity of candidate node pairs). The connectivity of a CNP (B, x) in bipartite graph $G=(U, V, E)$ is defined as

$$S(B,x) = \sum_{\{B,C\} \in \Gamma(B,x)} w(B,C). \tag{4}$$

Here, $w(B, C)$ is the weight of pattern $\{B, C\}$, and $\Gamma(B, x)$ is the set of the patterns covered by CNP (B, x):

$$\Gamma(B, x) = \{\{B, C\} | C \in N_u(B) \cap N(x)\} \tag{5}$$

From the definition, we can see that the connectivity of the CNP (B, x) is simply the summation of the weights corresponding to the patterns covered by (B, x). From formula (4), we know that if a CNP covers more patterns with higher weights, then it will have higher connectivity. Because the weight of a pattern reflects the similarity of nodes in the pattern, the connectivity of the CNP defined by (4) reflects the probability of the CNP to be connected. Therefore, we use the connectivity of a CNP as the final score of link prediction.

5. Framework of the algorithm and complexity analysis

Based on CNPs and their connectivities, we present a PLP (potential link prediction) algorithm for link prediction in bipartite networks. The algorithm detects potential links only within the set of CNPs. It first maps the given bipartite graph into a projected graph and then calculates the weight of each link in the projected graph according to formula (3). Finally, the connectivity of each CNP (B, x) is computed according to formula (4) and output as the final score indicating the likelihood of link (B, x). The framework of the PLP algorithm is described as follows.

PLP Algorithm (Potential Link Prediction).

Input: Bipartite graph $G=(U, V, E)$;
Output: Set of CNPs and connectivity matrix S;
Begin
(1) /* Construct the set of all patterns */
$E_u = \Phi$;
For each node B in U **do**
For each node x in $N(B)$ **do**
For each node C in $N(x)$ **do**
$E_u = E_u \cup \{(B, C)\}$;
Endfor
Endfor
Endfor
(2) /* Calculate the weight of each pattern */
For each edge (B, C) in E_u **do**
Calculate the weight of pattern (B, C) according to formula (3);
Endfor;
(3) /* Calculate the connectivity of CNPs */;
For each node B in projected graph G_u **do**
For each neighbor C of B in projected graph G_u **do**
For each node x linked with C in bipartite graph G **do**
If $(B, x) \notin E$ then
$S(B, x) = S(B, x) + w(B, C)$;
Endif
Endfor
Endfor
Endfor
(4) Output the connectivity of the CNPs in matrix S.
End

The first step of the algorithm constructs the set of all patterns by enumerating all edges in the projected graph. According to (3), the second step calculates the weight of each pattern—namely, the weights of links in the projected graph. The third step calculates the connectivities for all CNPs. Instead of enumerating all CNPs, we itemize all patterns and add their weights to all CNPs covering the pattern.

Suppose that $|U| = m$, and d is the maximal degree of nodes in set $U \cup V$. In the first step of the algorithm, because the maximum iterations of the three nested loops are m, d and d, the time complexity for this step is $O(m \times d^2)$. The second step of the algorithm calculates the weights of the patterns corresponding to the edges in the projected graph. Let B be a node in the projected graph; each edge in the projected graph connecting with B corresponds to a 2-hop path in the bipartite graph starting from B. Because number of such paths is less than d^2, the maximum number of edges connected with B in the projected graph is d^2. Because there are no more than $|U| = m$ nodes in the projected graph, there exist no more than $m \times d^2$ edges in the projected graph. Therefore, the complexity of the second step is $O(m \times d^2)$. In the third step of the algorithm, because the maximum iterations of the three nested loops are m, d and d, respectively, the time complexity for this step is also $O(m \times d^2)$. Therefore, the complexity of the PLP algorithm is $O(m \times d^2)$. Generally, the largest degree d in a network can be treated as a constant. The time complexity of the PLP algorithm is actually $O(m)$, which is linear to the number of nodes in part U. The reason for the low time complexity of the PLP algorithm is that it searches for the potential links only within the set of CNPs and ignores many node pairs that are impossible to be linked. In this way, the algorithm can reduce the computation time tremendously.

The algorithm described above is based on U-projection, which is efficient when $|U| < |V|$. If $|V| \leq |U|$, V-projection can be used to reduce the computation time. In general, let $m = \min(|V|, |U|)$; the time complexity of the PLP algorithm is $O(m)$,

which is less than that of other methods for link prediction in bipartite networks. For instance, the time complexity of the method based on the common neighbor (CN) index is $O(|V| \cdot |U|) = O(m^2)$. For the method based on the Katz index, its time complexity is $O((|V|+|U|)^3) = O((m+n)^3)$.

Allali et al. [3] defined the concept of internal links in a bipartite graph and presented an approach for link prediction based on internal links. Although the concept of candidate node pairs we presented is somewhat similar to the term of the internal link in [3], there are some important differences between our work and [3]:

1. In [3], the probability of an internal link to occur is related only to the number of edges it covers. However, in our method, the probability of each CNP to be a real link is determined according to the number of patterns it covers and the weights of those patterns. The weight of each pattern can be calculated according to three factors (Definition 5): (1) Number of common neighbors of the nodes in the pattern. (2) Degree of common neighbors of the nodes in the pattern. (3) Degrees of nodes in the pattern in the bipartite graph. By Definition 6, we define the measurement of connectivity, which can be used as the similarity index of a CNP. Because more topological information is considered in connectivity, our method can obtain more accurate prediction results.
2. In [3], after the concept of internal linkage is defined, no further analysis of the concept was provided. In our work, we present the necessary and sufficient condition for a node pair to be a CNP (Theorem 1). That condition is very useful for reducing the computation time to detect the CNP.
3. In [3], the detection of the internal links must enumerate all non-existing links; this requires large amount of computation time, which could be as large as $O(mn)$. In our work, by the necessary and sufficient condition for CNP we presented in Theorem 1, a CNP can be identified efficiently without enumerating all non-existing links, and the time complexity of our proposed algorithm is $O(m)$.
4. In [3], the projection can be performed on either part of the bipartite graph. It did not explain whether different projections could produce different results or the part on which the projection should be performed to obtain a better result. In this work, we have proved in Theorem 2 that a node pair is a CNP by projection on one part, if and only if it is also a CNP by projection on the other part. This means that projecting to either part will achieve the same result. Therefore, we must project the graph only to the part with fewer nodes to reduce the computation time.
5. In [3], the internal links with weights greater than a threshold are considered to be the potential links. Therefore, a proper threshold must be predefined to obtain accurate results. However, it is difficult to set a proper threshold in advance for a given dataset. In our method, we define connectivity as the similarity measure of a CNP, which integrates more topological information and can obtain more accurate prediction results. Like all similarity-based methods, such a similarity index indicates the probability of link occurrence.

6. Experimental results and analysis

To evaluate the proposed PLP algorithm for link prediction in a bipartite network, we test it by a set of experiments on some real-world bipartite networks: Southern Women, Divorce and Scotland. All experiments were performed on an Intel Core i3 computer running Windows 7, with 4GB of memory. The algorithm was coded using Matlab, and the results are visualized by Pajek and Visio.

We use AUC (area under curve) scores to measure the quality of the prediction results by the algorithms. After calculating the algorithms and ranking the similarities of all node pairs, which represent all existent and nonexistent links, the AUC value is the probability that an existing link has a higher similarity score than a non-existing link. We can randomly pick an existing link and a non-existing link to compare their similarity scores. Suppose that the existing link has a higher similarity than the non-existing link n' times and that they have an identical score n'' times. Let n be the total number of comparisons; the AUC value is defined as

$$AUC = (n' + 0.5n'')/n \tag{6}$$

In general, a larger AUC value indicates higher performance; hence, the AUC value of the perfect result is 1.0, whereas the AUC of the result by a random predictor is 0.5.

To evaluate the accuracy of the results, we use a random 10-fold cross-validation (CV). In 10-fold CV, nodes in the network are randomly divided into 10 subsets. From the 10 subsets, a single subset is selected as the test data, and the other 9 subsets are used as training data. This 10-fold CV procedure is performed 10 times on different training datasets. The average AUC score of the 10 tests is the final AUC score of the prediction results. We calculated the standard deviation of the AUC scores and found that all standard deviations are less than 0.023. On each dataset, we compare the AUC value of *PLP* with that of the other similarity-based link prediction algorithms such as common neighbor (*CN*), *Katz* and *LIP* [3]. The *CN* index used here for the bipartite graph is based on the number of quadrilaterals involving each node pair [33]. To use the *Katz* index for bipartite network, we keep only the odd components in the *Katz* polynomial [22]. The method of *LIP* [3] is based on the concept of internal links.

6.1. Test on Southern Women network dataset

We use the Southern Women network [30] to verify the accuracy of our algorithm. This dataset was collected by Davis et al. around Mississippi during the 1930s in a wide-ranging study of class and race in the Deep South. Because its community

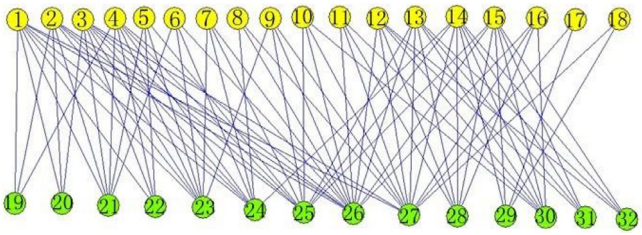

Fig. 8. Southern Women bipartite network.

Table 1
Main topological features of the Southern Women bipartite network.

Number of women nodes	Number of event nodes	Number of edges	Number of edges in testing set	Number of edges in training set	Number of CNPs
18	14	89	10	83	68

Table 2
Comparison of AUCs by *PLP* and other methods on the Southern Women dataset.

Test	*PLP*	*CN*	*Katz*	*ILP*
1	**0.9293**	0.8743	0.9107	0.9151
2	**0.9268**	0.8927	0.8997	0.9068
3	**0.9172**	0.8859	0.9043	0.9102
4	**0.9452**	0.9031	0.9155	0.9096
5	**0.9465**	0.9325	0.9239	0.9358
6	0.9217	0.8791	**0.9392**	0.9001
7	**0.9484**	0.9033	0.9179	0.9338
8	**0.9389**	0.8976	0.9007	0.9182
9	**0.9637**	0.9077	0.9351	0.9355
10	**0.9274**	0.8939	0.9114	0.9192
Average	**0.9365**	0.8970	0.9158	0.9184

structure is known, this dataset has been widely used as a benchmark for social network analyzers. The Southern Women network depicts the participation of 18 women in 14 social events. There are 89 edges linking women nodes and event nodes. If a woman participated in an event, the corresponding nodes are connected by an edge in the bipartite network. There is no edge between any two nodes of women or events. Fig. 8 shows the structure of the Southern Women network, in which each yellow node represents a woman, and each green node represents an event. In the network, each edge indicates that the woman attended the corresponding event. Table 1 shows the main topological features of Southern Women network.

Table 2 shows the AUC scores of 10 tests achieved by the algorithms. In the table, the largest AUC scores in the tests by 3 algorithms are stressed in boldface. From the table, we can find that *PLP* achieves higher AUC scores than *CN* and *ILP* in all 10 tests. For example, on the 1st test, the *PLP* algorithm obtains the AUC score 0.9293, whereas *CN* and *ILP* achieve 0.8743 and 0.9151, respectively. We can also see that *PLP* has higher AUC scores than *Katz* on 9 of the 10 tests. For example, on the 2nd test, the *PLP* algorithm obtains the AUC score 0.9268, whereas *Katz* achieves 0.8997. Although in the 6th test, *PLP* has a lower AUC score than *Katz*, the difference is insignificant. This shows that the *PLP* algorithm can obtain higher-quality results than other methods.

6.2. Test on Divorce dataset

As the second example, we test the Divorce dataset, a bipartite network of divorce status in the fifty states of the United States [49]. The dataset consists of 50 states and 9 statuses of divorce. A characteristic of this bipartite network is that the numbers of nodes in the two parts are seriously imbalanced. Fig. 9 shows the structure of the Divorce network, in which each yellow node represents a state, and each green node represents a divorce status. There are 255 edges in the network, each of which indicates that the state has the status of divorce. Table 3 shows the maim topological features of the Divorce network.

Table 4 presents the AUC scores of 10 tests achieved by different algorithms. In the table, the highest AUC scores in the tests by the 3 algorithms are emphasized in boldface. From the table, we can see that *PLP* obtains the highest AUC scores among the 4 algorithms in all 10 tests. This shows that the *PLP* algorithm can achieve higher-quality results than the other methods.

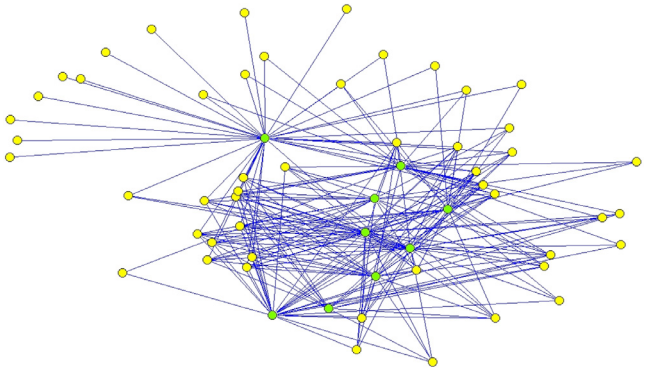

Fig. 9. Divorce bipartite network.

Table 3
Main topological features of the Divorce bipartite network.

Number of states	Number of status of divorce	Number of edges	Number of edges in testing set	Number of edges in training set	Number of CNPs
50	9	225	23	202	130

Table 4
Comparison of AUCs by *PLP* and other methods on Divorce dataset.

Test	*PLP*	*CN*	*Katz*	*ILP*
1	**0.9413**	0.8920	0.9239	0.9331
2	**0.9493**	0.9021	0.9085	0.9053
3	**0.9321**	0.8874	0.8974	0.9145
4	**0.9452**	0.9210	0.9230	0.9221
5	**0.9571**	0.9170	0.9184	0.9377
6	**0.9248**	0.8970	0.8678	0.8824
7	**0.9770**	0.9207	0.9387	0.9575
8	**0.9447**	0.8799	0.9149	0.9289
9	**0.9609**	0.9309	0.9407	0.9385
10	**0.9545**	0.9024	0.9257	0.9410
average	**0.9487**	0.9050	0.9159	0.9261

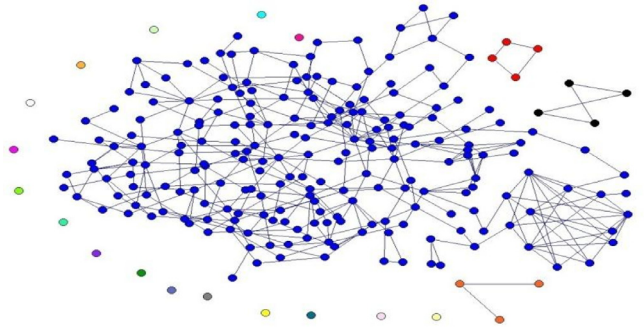

Fig. 10. Structure of Scotland bipartite network.

6.3. Test on Scotland dataset

As the third example, we test the Scotland dataset, a network of corporate interlocks in Scotland in the early twentieth century [35]. The dataset characterizes 108 Scottish firms during the years of 1904–1905, detailing the corporate sector, capital, and board of directors for each firm. The dataset includes only those board members who held multiple directorships, totaling 136 individuals as shown in Fig. 10.

Here, we focus on the bipartite network of firms and directors, with edges existing between each firm and its board members. Unlike the Southern Women and Divorce networks, the Scotland corporate interlock network is not connected.

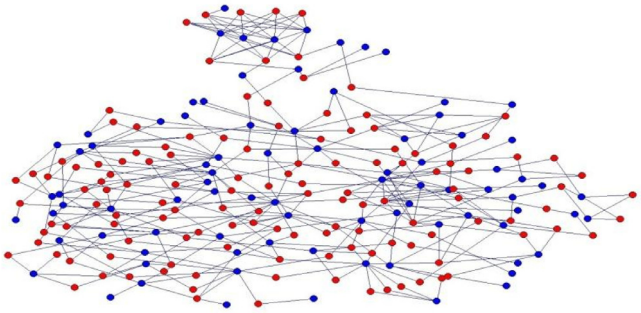

Fig. 11. The largest component of the Scotland bipartite network.

Table 5
Main topological features of the largest component of the Scotland bipartite network.

Number of directors	Number of firms	Number of edges	Number of edges in testing set	Number of edges in training set	Number of CNPs
131	86	348	35	313	123

Table 6
Comparison of AUCs by *PLP* and other methods on the Scotland dataset.

Test	*PLP*	*CN*	*Katz*	*ILP*
1	**0.9179**	0.8901	0.9011	0.9059
2	0.8871	0.8953	**0.9108**	0.8911
3	**0.9235**	0.8989	0.9019	0.9004
4	0.8813	0.8947	**0.8997**	0.8806
5	**0.8951**	0.8759	0.8796	0.8875
6	**0.8880**	0.8645	0.8849	0.8863
7	**0.9136**	0.8830	0.8904	0.9021
8	0.8818	0.8794	**0.8997**	0.8816
9	**0.9144**	0.8793	0.9023	0.8987
10	**0.9545**	0.9307	0.9124	0.9426
Average	**0.9057**	0.8892	0.8983	0.8977

Table 7
W-values of AUC scores by *PLP* with those by other methods.

Dataset*W*-values	*PLP-CN*	*PLP-Katz*	*PLP-ILP*
Southern Women	0	3	0
Divorce	0	0	0
Scotland	5	20	2

We conduct the experiments on the largest component of the network containing 131 directors and 86 firms as shown in Fig. 11. Table 5 shows the maim topological features of the largest component of the Scotland network.

Table 6 presents the AUC scores in 10 tests achieved by different algorithms. From the table, we can see that *PLP* obtains higher AUC scores than *CN* in 8 tests, higher AUC scores than ILP in 9 tests, and higher AUC scores than *Katz* on 7 tests. Although in a few tests, *PLP* has lower AUC scores than *CN, ILP* or *Katz*, their differences are insignificant. From Fig. 11, we can see that the largest component of the Scotland network is sparsely connected. Because of the sparseness of the Scotland network, the AUC scores of link prediction results by all algorithms decrease to a certain extent. However, from Table 6, we can see that *PLP* still obtains the highest-quality results in most of the tests. This shows that the *PLP* algorithm can achieve higher-quality results than other methods even in sparse networks.

Based on the AUC scores of the results on the three datasets shown in Tables 2, 4 and 6, we use the Wilcoxon signed-rank test to show that the AUC score of the result by *PLP* is significantly different from those by the other methods. We compare the *W*-values of AUC scores by *PLP* with those by the other methods, and the results are shown in Table 7. We set the confidence level α=0.01 and the number of samples $n = 10$. From the table of the criterion *W* value, we know that $W(0.01, 10) = 7$. From Table 7, we can see that the *W* values of AUC scores by *PLP* compared with those by *CN* are all less than $W(0.01, 10)$. This means that there are significant differences between the AUC scores by *PLP* and *CN*. Therefore, the quality of the results by the *PLP* algorithm is obviously higher than that of *CN* in all datasets. We also can see that the *W* values of AUC scores by *PLP* compared with those by *ILP* are all less than $W(0.01, 10)$. This means that the quality of the

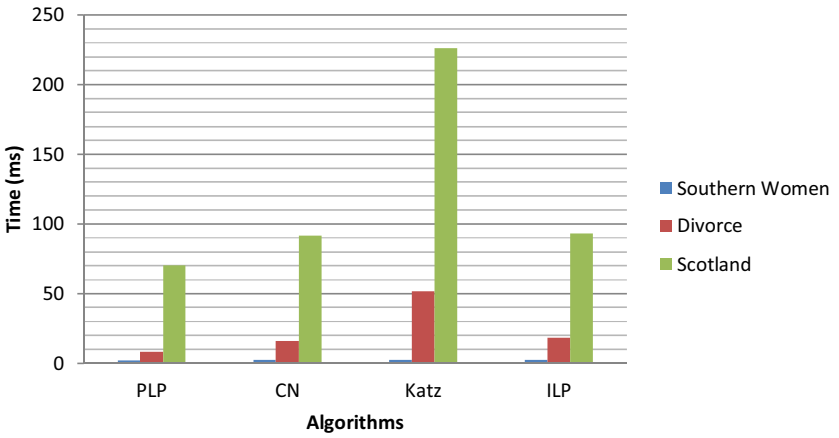

Fig. 12. Computation time of the algorithms.

results by the *PLP* algorithm is obviously higher than that of *ILP* in all datasets. From Table 7, we can see that the *W* values of AUC scores by *PLP* compared with those by *Katz* on the datasets Southern Women and Divorce are all less than *W*(0.01, 10). This means that the quality of the results by the *PLP* algorithm is obviously higher than that of *Katz* in those datasets. For the dataset Scotland, the *W* values of the AUC scores by *PLP* compared with those by *Katz* is greater than *W*(0.01, 10), which means that there is no significant difference between the AUC scores by *PLP* and *Katz* on Scotland. Because *Katz* is a global index, it can obtain high-quality prediction results. However, *PLP* can obtain results with high quality similar to those by *Katz* in less computation time. Therefore, *PLP* is more efficient.

6.4. Test on the time requirement by the algorithms

Computational complexity is another important concern in designing a link prediction algorithm. In the experiments, we compared the time required by the *PLP* algorithm with those by algorithms based on indexes *CN*, *Katz* and *ILP*. Fig. 12 shows the comparison of the average computation times required by different algorithms. From the figure, we can see that the *PLP* algorithm consumes much less computation time than the other algorithms.

Let n and m be the numbers of nodes of the two parts in the bipartite network, and $m < n$. The time complexity for link prediction in the bipartite network by the *PLP* algorithm is $O(m)$. However, the time complexity for computing the *CN* index is $O(m.n)$. Because *CN* is a local similarity index that uses only the second-order neighbors of the nodes, the quality of its prediction results is much lower than that of *PLP*. The *ILP* algorithm detects the internal links by enumerating all non-existing links; it requires a large amount of computation time, which could be as large as $O(m.n)$. *Katz* is a global similarity index, and its time complexity is $O((m+n)^3)$. Compared with all similarity-based methods, the *PLP* algorithm can achieve high-quality prediction results in less computation time. The reason why *PLP* consumes less computation time is that it detects the potential links only within the set of CNPs and ignores the node pairs that cannot be linked. Therefore, *PLP* can obtain a better prediction result in less time than the other methods.

7. Conclusions

An algorithm named *PLP* for link prediction in a bipartite network is presented. In the algorithm, we first map the bipartite network onto a unipartite one called a projected graph. Based on the projected graph, we define the concept of a candidate node pair (CNP). We perform the link prediction only within the CNPs to reduce the computation time. We also define the pattern covered by the CNPs and the weights of the patterns. By calculating the weights of the patterns covered by a CNP, the connectivity of the CNP can be obtained, which can be used as the final score of link prediction. Let n and m be the numbers of nodes on the two parts in the bipartite network, and $m < n$. The time complexity for link prediction in the bipartite network by the *PLP* algorithm is $O(m)$, whereas those of other algorithms are $O(mn)$ or $O((m+n)^3)$. Our experimental results show that our algorithm can achieve higher speed and superior quality link prediction results.

Acknowledgments

This research was supported in part by the Chinese National Natural Science Foundation under Grant Nos. 61379066, 61070047, 61379064, 61472344, 61402395; the Natural Science Foundation of Jiangsu Province under contracts BK20130452, BK2012672, BK2012128, BK20140492; the Natural Science Foundation of Education Department of Jiangsu Province under contracts 12KJB520019, 13KJB520026, 09KJB20013; and the Six Talent Peaks Project in Jiangsu Province (Grant No. 2011-DZXX-032).

References

[1] H.S. Ahmed, B.M. Faouzi, J. Caelen, Detection and classification of the behavior of people in an intelligent building by camera, Int. J. Smart Sens. Intell. Syst. 6 (4) (2013) 1317–1342 September.
[2] L.M. Aiello, A. Barrat, R. Schifanella, et al., Friendship prediction and homophily in social media, ACM Trans. Web (TWEB) 6 (2) (2012) 9.
[3] O. Allali, C. Mangier, M. Latapy, Link prediction in bipartite graphs using internal links and weighted projection, in: Proceedings of 2011 IEEE International Conference on Computer Communications Workshops (workshop in INFORM), Shanghi, 2011, pp. 936–941.
[4] Z.F. Bao, Y. Zeng, Y.C. Tay, sonLP: social network link prediction by principal component regression, in: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, August, 2013.
[5] N. Barbieri, F. Bonchi, G. Manco, Who to follow and why link prediction with explanations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 1266–1275.
[6] N. Benchettara, R. Kanawati, C. Rouveirol, supervised machine learning applied to link prediction in bipartite social networks, in: Proceedings of 2010 International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2010, pp. 326–330.
[7] C.A. Bliss, M.R. Frank, C.M. Danforth, P.S. Dodds, An evolutionary algorithm approach to link prediction in dynamic social networks, J. Comput. Sci. 5 (5, September) (2014) 750–764.
[8] B. Bringmann, M. Berlingerio, F. Bonchi and A. Gionis, learning and predicting the evolution of social networks, IEEE Intell. Syst., (2010), 26–34.
[9] Y.J. Chang, H.Y. Kao, Link prediction in a bipartite network using wikipedia revision information, in: Proceedings of the 2012 Conference on Technologies and Applications of Artificial Intelligence (TAAI), 2012, pp. 50–55.
[10] L. Ferreri, E. Venturino, M. Giacobini, Do diseases spreading on bipartite networks have some evolutionary advantages , evolutionary computation, machine learning and data mining in bioinformatics, Lect. Notes Comput. Sci. 6623 (2011) 141–146.
[11] S. Gao, L. Denoyer, P. Gallinari, et al., Probabilistic latent tensor factorization model for link pattern prediction in multi-relational networks, J. China Univ. Posts Telecommun. 19 (2012) 172–181.
[12] S. Gao, L. Denoyer, P. Gallinari, Temporal ink prediction by integrating content and structure information, in: Proceedings of CIKM'11, Glasgow, Scotland, UK, October 2011, pp. 1169–1174.
[13] M.Q. Ge, A. Li, M.H. Wang, A bipartite network-based method for prediction of long non-coding RNA–protein interactions, Genomics Proteomics Bioinformatics 14 (1) (2016) 62–71.
[14] S. Hanneke, W.J. Fu, E.P. Xing, Discrete temporal models of social networks, Electron. J. Stat. 4 (2010) 585–605.
[15] Y.L. He, J.N.K. Liu, Y.X. Hu, X.Z. Wang, OWA operator based link prediction ensemble for social network, Expert Syst. Appl. 42 (1) (2015) 21–50.
[16] F.Y. Hu, H.S. Wong, Labelling of human motion based on CBGA and probabilistic model, Int. J. Smart Sens. Intell. Syst. 6 (2) (2013) 583–609.
[17] X. Jia, F. Xin, W.R. Chuan, Adaptive spray routing for opportunistic networks, Int. J. Smart Sens. Intell. Syst. 6 (1) (2013) 95–119.
[18] H. Kim, J. Tang, R. Anderson, C. Mascolo, Centrality prediction in dynamic human contact networks, Comput. Netw. 56 (2012) 983–996.
[19] B. Kaya, M. Poyraz, Supervised link prediction in symptom networks with evolving case, Measurement 56 (2014) 231–238.
[20] B. Kaya, M. Poyraz, Unsupervised link prediction in evolving abnormal medical parameter networks, Int. J. Mach. Learn. Cybern. 7 (1) (2016) 145–155.
[21] P. Klimek, A.S. Jovanovic, R. Egloff, R. Schneider, Successful fish go with the flow: citation impact prediction based on centrality measures for term—document networks, Scientometrics 107 (3) (2016) 1265–1282.
[22] J. Kunegis, E.W. De Luca, S. Albayrak, The Link prediction problem in bipartite networks, in: E. Hullermeier, R. Kruse, F. Hoffmann (Eds.), Proceedings of IPMU, 2010, pp. 380–389.
[23] Y.H. Li, A.M. Wen, Q. Lin, R.X. Li, Zhengding Lu, Name disambiguation in scientific cooperation network by exploiting user feedback, Artif. Intell. Rev. 41 (4) (2014) 563–578.
[24] X. Li, H.C. Chen, Recommendation as link prediction in bipartite graphs: a graph kernel-based machine learning approach, Decis. Support Syst. 54 (2) (2013) 880–890.
[25] Z.H. Liu, J.F. Ma, Y. Zeng, Secrecy transfer for sensor networks: from random graphs to secure random geometric graphs, Int. J. Smart Sens. Intell. Syst. 6 (1) (2013) 77–94.
[26] J. Liu, G. Denga, Link prediction in a user_object network based on time-weighted resource allocation, Physica A 388 (2009) 3643–3650.
[27] W. Liu, L. Lü, Link prediction based on local random walk, Europhys. Lett. 89 (5) (2010) 58007.
[28] L.Y. Lü, T. Zhou, Link prediction in complex networks: a survey, Physica A, 390(2011) 1150–1170.
[29] Y.F. Luo, Q.J. Liu, W.J. Wu, F. Li, X.C. Bo, predicting drug side effects based on link prediction in bipartite network, in: Proceedings of the 2014 7th International Conference on BioMedical Engineering and Informatics, (BMEI 2014), pp.729–733.
[30] M. Newman, Detecting community structure in networks, Eur. Phys. J. B 38 (2) (2004) 321–330.
[31] A. Nigam, N.V. Chawla, Link prediction in a semi-bipartite network for recommendation, Lect. Notes Comput. Sci. 9622 (2016) 127–135 March.
[32] M. Pujari, R. Kanawati, Supervised rank aggregation approach for link prediction in complex networks, in: Proceedings of WWW 2012 Companion, Lyon, France, April 16–20, 2012, pp. 1189–1196.
[33] S. Purnamrita, D. Chakrabarti, M. Jordan. Nonparametric link prediction in dynamic networks, 2012, arXiv preprint arXiv:1206-6394.
[34] M. Savić, M. Ivanović, M. Radovanović, Z. Ognjanović, A. Pejović, T.J. Krüger, The structure and evolution of scientific collaboration in Serbian mathematical journals, Scientometrics 101 (3) (2014) 1805–1830.
[35] J. Scott and M. Hughes, The anatomy of Scottish capital: Scottish companies and Scottish capital, 1900–1979, CroomHelm, London, 1980.
[36] S. Shahinpour, S. Butenko, Distance-based clique relaxations in networks: s-clique and s-club, models, algorithms, and technologies for network analysis, Springer Proc. Math. Stat. 59 (2013) 149–174.
[37] B. Shams, S. Haratizadeh, SibRank: Signed bipartite network analysis for neighbor-based collaborative ranking, Physica A: Statistical Mechanics and its Applications 458 (15) (2016) 364–377.
[38] E. Sherkat, M. Rahgozar, M. Asadpour, Structural link prediction based on ant colony approach in social networks, Physica A 41 (1) (2015) 80–94.
[39] C. Villiers, The role of investor networks in transnational corporate governance, Networked Governance, Transnational Business and the Law, (2014) 285–313
[40] D.Q. Vu, A.U. Asuncion, D.R. Hunter, P. Smyth, Continuous-time regression models for longitudinal networks, in: Proceedings of the 25th Annual Conference on Neural Information Processing Systems, 2011, pp. 1–9.
[41] P. Wang, B.W. Xu, Y.R. Wu, X.Y. Zhou, Link prediction in social networks: the state-of-the-art, Sci. China Info. Sci. 58 (1) (2015) 1–38.
[42] X.M. Wang, Y. Liu, F. Xiong, Improved personalized recommendation based on a similarity network, Physica A 456 (15) (2016) 271–280.
[43] S. Xia, B.T. Dai, E.P. Lim, Y. Zhang, C.X. Xing, Link prediction for bipartite social networks: the role of structural holes, in: Proceedings of 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2012, pp. 153–157.
[44] L.T. Yang, S. Wang, H. Jiang, Cyclic temporal network density and its impact on information diffusion for delay tolerant networks, Int. J. Smart Sensing Intell. Syst. 4 (1) (2011) 35–52.
[45] Z.Z. Zeng, K.J. Chen, S.B. Zhang, H.J. Zhang, A link prediction approach using semi-supervised learning in dynamic networks, in: Proceedings of 2013 Sixth International Conference on Advanced Computational Intelligence (ICACI), 2013, pp. 276–280.
[46] P. Zhang, A. Zeng, Y. Fan, Identifying missing and spurious connections via the bi-directional diffusion on bipartite networks, Phys. Lett. A 378 (32-33) (2014) 2350–2354.
[47] Y.J. Zhang, C. Tao, G.Q. Jiang, A.A Nair, J. Su, C.G. Chute, H.F. Liu, Network-based analysis reveals distinct association patterns in a semantic MEDLINE-based drug-disease-gene network, J. Biomed. Semant. 5 (2014) 33.
[48] T. Zhou, L. Lü, Y.C. Zhang, Predicting missing links via local information, Eur. Phys J B 71 (4) (2009) 623–630.
[49] http://www.stat.lsa.umich.edu/.