Information Sciences 394–395 (2017) 198–216

Contents lists available at ScienceDirect

Information Sciences

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

Link prediction in multi-relational networks based on relational similarity

Caiyan Dai[a], Ling Chen[a,b,*], Bin Li[a], Yun Li[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 30 September 2015
Revised 7 August 2016
Accepted 2 February 2017
Available online 7 February 2017

Keywords:
Multi-relational networks
Link prediction
Similarity

ABSTRACT

Many real-world networks contain multiple types of interactions and relations. Link prediction in such multi-relational networks has become an important area in network analysis. For link prediction in multi-relational networks, we should consider the similarity and influence between different types of relations. In this paper, we propose a link prediction algorithm in multi-relational networks based on relational similarity. In the algorithm, a belief propagation method is presented to calculate the belief of each node and to construct the belief vector for each type of link. We use the similarity between belief vectors to measure the influence between different types of relations. Based on the influence between different relations, we present a nonnegative matrix factorization -based method for link prediction in multi-relational networks. The convergence and correctness of the presented method are proved. Our experimental results show that our method can achieve higher-quality prediction results than other similar algorithms.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

Many social, biological, and information systems in the real world, from the nervous system to the ecosystem, from road traffic networks [7] to the neural networks [53], and from an ant colony structures to human social networks [1,10,35,39], can be naturally described as networks in which vertices represent entities and links denote relationships or interactions between vertices. As a topological approximation of complex systems, it is inevitable that there will be some errors or redundant links in constructing a complex network due to the limitations of time and space or of experimental conditions. At the same time, there will be some undetected potential links. In addition, because of the dynamic evolution of complex network links over time, we must predict missing and potential links according to known network information. This is the goal of the network link-prediction problem [30,47,49].

Link-prediction has a wide range of applications in various real-world fields. For example, in biological networks, such as protein-protein interaction networks, metabolic networks, and diseases-gene networks [23], links between nodes indicate an interaction relationship. To mitigate the high costs of biological experimentation to reveal the hidden interaction relationships in these networks, the results of link prediction can direct biological experiments designed to reduce the cost and improve the success rate of the experiments. Predicting the loss and suspicious links of disease-gene networks can help explore the mechanisms of diseases and predict and evaluate their treatments. It can also be used to find new drug targets and open up new approaches for drug development [18].

[*] Corresponding author.
E-mail addresses: daicaiyan@163.com (C. Dai), yzulchen@163.com (L. Chen), lb@yzu.edu.cn (B. Li), liyun@yzu.edu.cn (Y. Li).

http://dx.doi.org/10.1016/j.ins.2017.02.003
0020-0255/© 2017 Elsevier Inc. All rights reserved.

In complex network analysis, link prediction can also be used as a powerful supplementary tool to accurately analyze the complex network structure. Studies on online social network analysis have been developing rapidly in recent years. In online social networks, the potential friendship of users can be revealed by link prediction, and these potential friends can be recommended to the users [37]. By analyzing social relationships, we can find potential interpersonal links [11,15,22]. Link prediction can also be used in the academic network to predict the type of academic paper and its coordinators [43]. A link-prediction method can also be directly used for recommendations in e-commerce [28,50], such as a commodity recommendation to customers [29,45]. Marketers would like to recommend products or services based on existing preferences or contacts. Social networking websites customize suggestions for new friends and groups using link prediction. Financial corporations must monitor transaction networks to detect fraudulent activity via link prediction. Link prediction is applied in e-mail communication analysis to detect anomalous e-mail. In monitoring networks of criminals, link prediction is used to discover hidden connections between criminals to prevent crime or terrorist activity. In recent years, link prediction has been widely applied in areas such as recommendation [36,50], scientist-paper subject prediction [44], scientific paper impact prediction [26], medical parameter network analysis [25], and protein interaction prediction [17].

Link prediction not only has a broad range of practical applications but also has important theoretical significance. For example, link prediction is helpful to understand the mechanism of the evolution of a complex network [24,34]. Because the magnitude of the internal characteristics of a complex network structure is very large, it is difficult to compare the advantages and disadvantages of different mechanisms. Link prediction can provide a simple and unified platform for a fair comparison of network evolution mechanisms to promote theoretical research on the complex-network evolution model.

Previous works on link prediction mainly focus on single-relational networks, which contain only one type of edge. Many real-world relational systems, however, are naturally described as multi-relational networks containing multiple types of edges [13,42,51]. In such a network, different types of edges represent different relations between the individuals. Such relations can be either explicit or implicit. In the first case, the relations directly reflect the various interactions in reality; in the second case, the relations are defined by the analyst to reflect different interesting qualities of the interactions that can be inferred from the available data. Examples of networks with explicit relations are social networks, where interactions represent information diffusion such as email exchange, telecommunications, and instant messaging services. An example of implicit relations is on-line social networks such as Flickr, where, among other features, two users can be implicitly linked by their favorite photos. In such multi-relational networks, different types of links may reflect different types of relationship, such as friends, relatives, and colleagues [13].

There exists an influence between each pair of relations. This influence may vary for different pairs of relations. For instance, there may be a higher probability that a person will form a new friendship with a friend of his colleague than with a friend of his relative. In this case, the relation “colleague” has a greater influence on the relation “friend” than does “relative”. For link prediction in such multi-relational networks, we should consider the relevance of and influence between different relations integrally. Traditional link prediction methods are not applicable to multi-relational networks because these methods only consider one relation and ignore the whole structure along with the influence between different relations. A simple way to predict the potential links in multi-relational networks is to transform a multi-relational into a single-relational network and then conduct link prediction on the transformed network. However, we may lose too much valuable information in this simplified process and will be unable to obtain high-quality results. In this scenario, we expect that a method that effectively utilizes multi-faceted information in the multi-relational network would enhance the link prediction results.

In this paper, we propose an algorithm named LPMR for link prediction in multi-relational networks based on relational similarity. In the algorithm, we first present a method to calculate the belief of each node by belief propagation and to construct the belief vector for each type of link. We use the similarity between belief vectors as the influence between different relational networks. Based on the influence vectors, a nonnegative matrix factorization-based method is proposed to predict links in multi-relational networks. We theoretically prove the convergence and correctness of the LPMR algorithm. Our experimental results demonstrate that our method can achieve higher-quality prediction results than similar algorithms.

The rest of this paper is structured as follows. Section 2 reviews related works on link prediction in complex networks. Section 3 proposes a belief propagation algorithm for measuring the similarity of graphs and defines the influence between relations based on their belief vectors. In Section 4, a nonnegative matrix factorization method called LPMR is presented to predict links in multi-relational networks. Section 5 shows and analyzes the experimental results obtained by the LPMR algorithm. Finally, we offer a conclusion in Section 6.

2. Related works

Many methods of link prediction have been reported in recent years. These methods can be classified into three categories: similarity-based methods, machine-learning methods, and probabilistic model-based methods.

The similarity-based method is the most commonly used method for link prediction. In the similarity-based method, each node pair is assigned an index, which is defined as the similarity score between the two nodes. All non-observed links are ranked according to their similarity scores, and the non-observed links connecting more similar nodes are supposed to have higher existence likelihoods. A node similarity score can be defined by using the essential topological features of nodes: two nodes are assigned a higher similarity score if they have more common features or correlated topological structures [2,20]. Many studies have found substantial levels of topical similarity among individuals who are close to one another in the social network. For instance, L. M. Aiello et al. [3] studied friendship prediction in social networks based on the presence of homology in three systems that combine tagging social media with online social networks. The structural similarity scores can be classified into three categories: local indices, global indices, and quasi-local indices. Local indices use only the neighbor information of the nodes. Typical local indices include Common Neighbors, Salton Index, Sorensen Index, Jaccard Index, Leicht-Holme-Newman Index, Hub Depressed Index, Hub Promoted Index, Preferential Attachment Index, Adamic-Adar Index, and Resource Allocation Index [33]. Global indices require global topological information. Katz Index, Leicht-Holme-Newman Index, and Matrix Forest Index [33] are typical global indices. Quasi-local indices do not require global topological information but make use of more information than do local indices. Such indices include the Local Path Index, Local Random Walk, and Superposed Random Walk [33]. A similar group of indices is based on the random walk. This group includes indices such as Average Commute Time, Cos+, Random Walk with Restart, and SimRank [33]. Lü et al. [32] proposed two new local indices, Resource Allocation index and Local Path index. Empirical results show that these two indices outperform all other local indices. In particular, the local path index, requiring somewhat more information than the common neighbors index, provides competitively accurate predictions compared with the global indexes. X. J. Wang [48] presented a method for predicting link directions using local directed paths in a directed network. Liu and Lü [31] studied the link prediction problem based on Local Random Walk (LRW) and Superposed Random Walk (SRW). They found that the limited step might obtain a better prediction than the result of global random walk. Because the random walk-based methods such as LRW and SRW require O(n^3) time for a network with n nodes, they are unfeasible in many real applications. Das Sarma et al. presented a distributed random walk algorithm [40] with time complexity $O(\sqrt{\tau.\Phi})$, where$\tau$is the dynamic mixing time and Φ is the dynamic diameter of the network.

Machine-learning strategies are also exploited in network link-prediction methods. M. Pujari et al. [38] presented a supervised rank aggregation method for link prediction in complex networks. D. Q. Vu et al. [46] introduced a continuous-time regression model for network link prediction. The model can incorporate both time-dependent network statistics and time-varying regression coefficients. Z. Z. Zeng et al. [52] presented a method incorporating semi-supervised learning into the link prediction task to use the potential information in a large number of unlinked node pairs in networks. Based on a weighted averaging operator, Y. L. He et al. [19] proposed an ensemble-based link-prediction algorithm. The algorithm assigns weights for nine local information-based link prediction algorithms and then aggregates their results to obtain final prediction scores. Z. F. Bao et al. [4] advanced a network link predictor using principal component analysis to identify features that are important to link prediction. B. Bringmann et al. [9] proposed an approach to link prediction in temporal networks based on the techniques of association rule mining and frequent-pattern detecting. Using data mining and machine learning techniques, the method can predict the future co-participation of individuals in social events. To avoid a high computational cost of optimization in the machine-learning methods, some heuristic methods are employed in link prediction. E. Sherkat et al. [41] proposed an unsupervised structural link prediction algorithm based on ant colony optimization. C. A. Bliss et al. [8] presented an approach to predicting future links by employing the covariance matrix-adaptation evolution strategy. J. Y. Ding et al. [14] advanced a method based on multi-resolution community partitioning to predict potential links in a network. B. Chen et al. [12] proposed an approximate algorithm to predict links related to nodes of interest.

Some methods for link prediction for a network are based on probabilistic models. S. Gao et al. [16] proposed a model that exploits multiple information sources in the network to obtain link-occurrence probabilities. N. Barbieri et al. [5] proposed a stochastic topic model for link prediction over directed and node-attributed graphs. The model not only predicts links but also produces a different type of explanation for each link predicted. F. Y. Hu et al. [21] presented a probabilistic model to detect human motion in a social network 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 appearances, which is difficult to know in advance for a given network in real-world applications.

Some methods of link prediction in multi-relational networks have been proposed in recent years. V. Stroele et al. [42] exploited topological features of network structures by tensor decomposition. They defined the overall relationships between node pairs as a link pattern that consists of an interaction pattern and a connection structure in the network. Y. Yang et al. [51] proposed a probabilistic approach for link prediction in multi-relational heterogeneous networks. They also advanced an unsupervised learning method for link prediction in heterogeneous networks. Darcy Davis et al. [13] proposed a probabilistically weighted extension of the Adamic/Adar measurement and used it as a similarity index for multi-relational networks. To develop a solid repertoire of basic concepts and analytical mechanisms for multi-relational networks, M. Berlingerio et al. [6] modeled a multi-relational network as a multi-graph. They systematically developed topological metrics for the graph to characterize local and global properties of multi-relational networks. V. Stroele et al. [42] presented a method for link prediction in multi-relational scientific social networks using clustering techniques with a maximum flow measure. None of these methods for link prediction in multi-relational networks considers the influence between different relations.

3. Influence between relations

3.1. Multi-relational network and its link prediction

A multi-relational network can be presented by a graph $G=(V,E)$, where V is the set of vertices, $E=E_1\cup E_2\cup_{\dots}E_d$ is the set of different types of edges, and d is the number of different types of edges in the network. We denote the sub-graph of G consisting of the r-th type of edges as G_r=(V, E_r), and we use $A^{(r)}=[a_{ij}^{(r)}]_{n^*n}$ to denote the adjacency matrix of sub-graph G_r, where $a_{ij}^{(r)}=1$ if an r-th type edge exists between nodes v_i and v_j, and $a_{ij}^{(r)}=0$ otherwise. It is worth mentioning that a G_r sub-graph is not necessarily connected, even though the original network G is connected. This is because the nodes in G are connected by all types of edges, while the nodes in G_r are connected only by the r-th type of edges. But different subgraphs may be connected with each other since they have identical node set.

Given an integer t $(1 \le t \le d)$, the goal of link prediction is to predict potential t-th type links in the network using the topological information of all types of links represented by adjacent matrixes $A^{(i)}$ $(i=1,2,\ldots,d)$. We call the t-th type link the target relation or target dimension. The final result of link prediction is represented by a link similarity score matrix $S^{(t)}=[s_{ij}^{(t)}]_{n*n}$, where $s_{ij}^{(t)}$ indicates the probability of the existence of the t-th type link between nodes v_i and v_j. The larger the value of $s_{ij}^{(t)}$, the higher likelihood of the t-th type link between nodes v_i and v_j. Therefore, probability matrix $S^{(t)}$ can intuitively reflect the presence of a t-th type link between the nodes.

3.2. Influence between relations

Suppose the r-th type of link is the target relation. For predicting potential r-th type links, we not only consider the topological information in adjacent matrix $A^{(r)}$ of the target type of relation, but we also utilize the topological information of the other relations. This is because different relations might influence each other. However, the influences on the target relation from other relations are quite different. For instance, a friend of Jack's colleague could also be Jack's friend, but it might be nearly impossible for a relative of Jack's colleague to be his relative. To predict the relation of "colleague", the "friend" relation has a greater influence than "relative". Therefore, it is important to analyze the influence of a relation with respect to another type of relation between the node pair. The relevance of and influence between these relations are helpful for achieving more-accurate link prediction results.

The sub-graph $G_t=(V, E_t)$ of G consisting of the t-th type of edges can be considered a single graph. The influence between these relations can be estimated by the similarity between their graphs. Two relations may have greater influence on each other if the two corresponding graphs are more similar.

We first define the similarity between two graphs. Suppose two graphs consist of the same set of nodes but possibly different sets of edges. We assume that we know the correspondence between the nodes of the two graphs. We define the graph similarity as a number between 0 and 1, indicating the degree of topological likeness between these two graphs. Intuitively, because we know the node correspondences, the same node in both graphs would be similar if their neighbors and links are similar. Again, the neighbors themselves are similar if their neighborhoods are similar, and so on. Therefore, we propose a belief propagation algorithm for measuring graph similarity. Given a network, the algorithm assigns each node a value of belief by message passing. Because the algorithm performs message passing based on the nodes' neighborhood structure, the belief of the nodes reflects the global topological features of the network. Two networks are similar if there is a high correlation between the beliefs of their nodes.

3.3. Belief propagation

In this section, we propose a belief propagation algorithm for measuring graph similarity. In the algorithm, each vertex v in the network is assigned a belief factor $B(v)$. To evaluate belief factor $B(v)$, a message is transmitted from each node and modified in the propagation. The method is iterative and consists of message passing between the connected nodes.

We define a transformation function $\psi(v_i, v_j)$ to be the probability for transforming a message from node v_i to node v_j:

$$\psi(v_i, v_j) = \frac{a_{ij}}{\sum\limits_{k\in N(i)} a_{ik}} \tag{1}$$

Here, $N(i)= \{v|v \in V, (v_i, v) \in E\}$is the set of direct neighbors of node v_i.

In belief propagation, nodes pass messages to their neighbors iteratively until convergence or until reaching a maximum number of iterations specified by the user. Let $m_{ij}(v_j)$ be the message from node v_i to node v_j. For a network with n vertices, we assign the initial value of $m_{ij}(v_j)$ as $\frac{1}{n+1}$. The message is updated recursively by the following rule:

$$m_{ji}(v_i) = \psi(v_j, v_i) \prod_{k\in N(j)\backslash i} m_{kj}(v_j) \tag{2}$$

After convergence of such message passing, the node receiving a higher message value will have large belief. Therefore, the belief of node v_i is proportional to the product of all of the messages it receives. Therefore, we define the belief of node v_i as:

$$B(v_i) = \frac{\sum\limits_{j\in N(i)} m_{ji}(v_i)}{d(v_i)} \tag{3}$$

where $d(v_i)=|\ N(i)|$ is the degree of node v_i.

The belief propagation method manages to capture both the local and global topology of the graphs. Therefore, it is able to spot small differences between the graphs. The method is general and can be applied to both directed and undirected graphs.

3.4. Similarity based on the belief vector

Because the belief value obtained by the propagation method captures both the local and global topology of the graphs, it can be used to estimate the similarity between graphs. Let G_t be the target dimension. After we calculate the belief of every vertex in sub-graphs G_1, G_2, ..., G_d, we use the belief of their nodes to evaluate the similarities between sub-graphs G_t and G_i ($i = 1,2,\ldots,d,\ i \neq t$).

Let $G_i = (V, E_i)$ and $G_j = (V, E_j)$ be two sub-graphs of G consisting of the i-th and j-th types of edges, respectively. We denote the belief of node v_j in the sub-graph G_i as b_{ij}. Thus, the beliefs of all of the nodes in sub-graph G_i form a belief vector $b_i = (b_{i1}, b_{i2}, \ldots, b_{in})$. We define the similarity of sub-graphs G_i and G_j as the similarity of their belief vectors b_i and b_j. We use Pearson's coefficient to estimate the similarity of vectors b_i and b_j, and the similarity between sub-graphs G_i and G_j is defined as

$$r_{ij} = \frac{n \sum_{k=1}^{n} (b_{ik} - \bar{b}_i)(b_{jk} - \bar{b}_j)}{\sum_{k=1}^{n} (b_{ik} - \bar{b}_i)^2 \sum_{k=1}^{N} (b_{jk} - \bar{b}_j)^2} \tag{4}$$

Here, $\bar{b}_i = \frac{1}{n} \sum_{k=1}^{n} b_{ik}$ is the mean value of the belief vector b_i. Similarity, r_{ij} between sub-graphs G_i and G_j can be used to measure the influence between the i-th and j-th relations. When predicting potential links in the target relation G_t, influences from the other relations in sub-graphs G_1, G_2, ..., G_d ($i = 1,2,\ldots,d,\ i \neq t$) should be considered. If G_i has higher influence r_{ti} on G_t, the topological information of the i-th type link should be more important in link prediction in G_t.

4. Link prediction by nonnegative matrix factorization

4.1. Nonnegative matrix factorization for link prediction

Let nonnegative matrix A be the $n \times n$ adjacent matrix of a network. The goal of nonnegative matrix factorization (NMF) is to find two nonnegative matrixes $U \in R_+^{n \times k}$ and $V \in R_+^{n \times k}$, such that their product is very close to matrix A :

$$A \approx UV^T$$

Here, k is the dimension of the latent space ($k < n$). Matrix U consists of the bases of the latent space and is called the base matrix. Matrix V represents the combination coefficients of the bases for reconstructing the matrix A and is called the coefficient matrix. This matrix factorization can be modeled as the following optimization problem:

$$\min_{U,V} ||A - UV^T||_F^2, \text{ s.t.} U \geq 0, V \geq 0 \tag{5}$$

Here, $|| \cdot ||_F$is the Frobenius norm, and constraints $U \geq 0$ and $V \geq 0$ require that all of the elements in matrixes U and V be nonnegative. The similarity between nodes v_i and v_j in the latent space can be represented by the similarity between the i^{th} and j^{th} row vectors in matrix V. Such similarity can be used as the score for estimating the probability of the potential link between nodes v_i and v_j.

Suppose there are d types of links in the network and their adjacency matrixes are $A^{(t)} = [a_{ij}^{(t)}]_{n*n}$, $(1 \leq t \leq d)$, where$A^{(t)} \in \mathbb{R}_+^{n \times n}$ is the adjacent matrix of the t-th type link. Let $A^{(1)}$ be the adjacent matrix of the target sub-graph G_1. Here, our goal is to predict the potential links of the first type. Suppose the adjacent matrix $A^{(t)}$ can be decomposed into $A^{(t)} \approx U^{(t)}(V^{(t)})^T$, $(1 \leq t \leq d)$. The j-th row vector of matrix $V^{(t)}$ represents the coefficient of node v_j in k-dimensional latent space represented by $U^{(t)}$. To take advantage of the topological information of different types of links, we should effectively integrate the adjacent matrixes in the low-dimensional latent space. To integrate various types of link information, we try to find a consensus latent space matrix U^* and a coefficient matrix V^* so that for every consensus latent space matrix $U^{(t)}$ and coefficient matrix $V^{(t)}$, the differences between U^* and $U^{(t)}$, V^* and $V^{(t)}$ $(1 \leq t \leq d)$ are minimized. That is, to minimize the Frobenius norm of the difference between U^* and $^{(t)}$, V^* and $V^{(t)}$ $(1 \leq t \leq d)$:

$$D(U^{(t)}, U^*, V^{(t)}, V^*) = ||V^{(t)} - V^*||_F^2 + ||U^{(t)} - U^*||_F^2 \tag{6}$$

Therefore, our goal is to obtain a proper $U^{(t)}$, $V^{(t)}$ $(1 \leq t \leq d)$, U^* and V^* to minimize the following objective function J:

$$\begin{aligned} J = \sum_{t=1}^{d} r_{1t} ||A^{(t)} - U^{(t)}(V^{(t)})^T||_F^2 + \sum_{t=1}^{d} r_{1t} ||U^{(t)} - U^*||_F^2 + \sum_{t=1}^{d} r_{1t} ||V^{(t)} - V^*||_F^2 \\ \text{s.t.} \qquad U^{(t)}, V^{(t)}, U^*, V^* \geq 0, \qquad (1 \leq t \leq d). \end{aligned} \tag{7}$$

In (7), r_{1t} is the influence of G_t on the target sub-graph G_1; it reflects the importance of the topological information of the t-th type link in predicting links on target sub-graph G_1. The base fact we considered is that if G_t has higher influence r_{1t} on G_1, the topological information of the t-th type relation should have higher significance in link prediction on G_1.

4.2. Iterative method for NMF

Although the function J in (7) is convex in each of $U^{(t)}$, $V^{(t)}$, U^*, V^* only; they are not convex in both variables together. Therefore, it is unrealistic to expect an algorithm to solve optimization problem (7) in the sense of finding the global minimum. We have found that the strategy of updating the variables separately using multiplicative update rules is feasible for solving optimization problem (7).

We present an iterative method to solve optimization problem (7) efficiently. Because $U^{(t)}$, $V^{(t)}$, U^* and V^* are variables in (7), we fix three of the variables at each iteration and obtain the best value of the fourth variable to minimize the objective function J. After setting proper initial values for matrixes V^*, U^*, $U^{(t)}$ and $V^{(t)}$, for $t=1,\ldots,d$, each iteration consists of four steps:

i Fix the matrices $V^{(t)}$, U^* and V^*, update $U^{(t)}$ to minimize J, $t=1,\ldots,d$;
ii Fix the matrices $U^{(t)}$, U^* and V^*, update $V^{(t)}$ to minimize J, $t=1,\ldots,d$;
iii Fix the matrices $U^{(t)}$, V^* and $V^{(t)}$, update U^* to minimize J.
iv Fix the matrices $U^{(t)}$, U^* and $V^{(t)}$, update V^* to minimize J.

The method repeats these four steps until convergence.

(1) Updating $U^{(t)}$

Fixing the matrices $V^{(t)}$,V^* and U^*, we update $U^{(t)}$ to minimize the objective function $J(U^{(t)})$, $t=1,\ldots,d$:

$$J(U^{(t)}) = r_{1t}||A^{(t)} - U^{(t)}(V^{(t)})^T||_F^2 + r_{1t}||U^{(t)} - U^*||_F^2 \tag{8}$$

Because

$$\frac{\partial J\left(U^{(t)}\right)}{\partial U^{(t)}} = r_{1t}[-2A^{(t)}V^{(t)} + 2U^{(t)}{V^{(t)}}^TV^{(t)} + 2U^{(t)} - 2U^*] \tag{9}$$

By the Karush-Kuhn-Tucker (KKT) condition, we know that:

$$2U^{(t)}{V^{(t)}}^TV^{(t)} + 2U^{(t)} - 2A^{(t)}V^{(t)} - 2U^* = 0 \tag{10}$$

By (10), we can obtain the following formula for updating $(U^{(t)})_{ij}$:

$$u_{ij}^{(t)} \leftarrow u_{ij}^{(t)}\frac{(A^{(t)}V^{(t)})_{ij}+u_{ij}^*}{\left(U^{(t)}V^{(t)T}V^{(t)}\right)_{ij}+u_{ij}^{(t)}},\\ t=1,2,..d;\quad i=1,2,\ldots n;\quad j=1,2,\ldots,k \tag{11}$$

(2) Updating $V^{(t)}$

Fixing the matrices $U^{(t)}$, U^* and V^*, we update $V^{(t)}$ to minimize the objective function $J(V^{(t)})$:

$$J(V^{(t)}) = r_{1t}||A^{(t)} - U^{(t)}(V^{(t)})^T||_F^2 + r_{1t}||V^{(t)} - V^*||_F^2 \tag{12}$$

Because

$$\frac{\partial J\left(V^{(t)}\right)}{\partial V^{(t)}} = -r_{1t}[2{U^{(t)}}^TU^{(t)}V^{(t)T} - 2{U^{(t)}}^TA^{(t)} + 2V^{(t)} - 2V^*] \tag{13}$$

By the KKT condition, we know that:

$$2{U^{(t)}}^TU^{(t)}V^{(t)T} - 2{U^{(t)}}^TA^{(t)} + 2V^{(t)} - 2V^* = 0 \tag{14}$$

By (14), we can obtain the following formula for updating $v_{ij}^{(t)}$:

$$v_{ij}^{(t)} \leftarrow v_{ij}^{(t)}\frac{\left(U^{(t)T}A^{(t)}\right)^T{}_{ij}+v_{ij}^*}{\left(U^{(t)T}U^{(t)}V^{(t)T}\right)^T{}_{ij}+v_{ij}^{(t)}}, \quad .\\ t=1,2,..,d; i=1,2,...,k; j=1,2,...,n \tag{15}$$

a) Updating U^*

Fixing the matrices $U^{(t)}$, $V^{(t)}$and V^*, we update U^* to minimize the objective function $J(U^*)$:

$$J(U^*) = \sum_{t=1}^{d} r_{1t}||A^{(t)} - U^{(t)}(V^{(t)})^T||_F^2 + \sum_{t=1}^{d} r_{1t}||U^{(t)} - U^*||_F^2 \tag{16}$$

Because

$$\frac{\partial J(U^*)}{\partial U^*} = \frac{\partial}{\partial U^*}\sum_{t=1}^{d} r_{1t}||U^{(t)} - U^*||_F^2 = \sum_{t=1}^{d} r_{1t}\frac{\partial}{\partial U^*}||U^{(t)} - U^*||_F^2 = 2\sum_{t=1}^{d} r_{1t}\left(U^{(t)} - U^*\right) \tag{17}$$

By the KKT condition, we know that:

$$\sum_{t=1}^{d} r_{1t}\left(U^{(t)} - U^{*}\right) = 0 \tag{18}$$

By (18), we can obtain the following formula for updating U^{*}:

$$U^{*} = \frac{1}{\sum\limits_{t=1}^{T} r_{1t}} \sum_{t=1}^{d} r_{1t} U^{(t)} \tag{19}$$

b) Updating V^{*}

Fix the matrices $U^{(t)}$, $V^{(t)}$ and U^{*}. We update V^{*} to minimize the objective function $J(V^{*})$:

$$J(V^{*}) = \sum_{t=1}^{d} r_{1t} ||A^{(t)} - U^{(t)} (V^{(t)})^{T} ||_F^2 + \sum_{t=1}^{d} r_{1t} ||V^{(t)} - V^{*}||_F^2 \tag{20}$$

Because

$$\frac{\partial J(V^{*})}{\partial V^{*}} = \frac{\partial}{\partial V^{*}} \sum_{t=1}^{d} r_{1t} ||V^{(t)} - V^{*}||_F^2 = \sum_{t=1}^{d} r_{1t} \frac{\partial}{\partial V^{*}} ||V^{(t)} - V^{*}||_F^2 = 2 \sum_{t=1}^{d} r_{1t} \left(V^{(t)} - V^{*}\right)$$

By the KKT condition, we know that:

$$\sum_{t=1}^{d} r_{1t}\left(V^{(t)} - V^{*}\right) = 0 \tag{21}$$

By (21), we can obtain the following formula for updating V^{*}:

$$V^{*} = \frac{1}{\sum\limits_{t=1}^{d} r_{1t}} \sum_{t=1}^{d} r_{1t} V^{(t)} \tag{22}$$

At each iteration of our algorithm, the new values of $U^{(t)}$, $V^{(t)}$, U^{*}or V^{*} can be found by multiplying the current value by some factors depending on the quality of the approximation in Eqs. (11), (15), (19), and (22), respectively. We will prove in Section 4.5 that the quality of the approximation improves monotonically with the application of these multiplicative update rules. In practice, this means that repeated iterations of the update rules are guaranteed to converge to an optimal matrix factorization.

4.3. Framework of the algorithm

Based on the updating rules obtained above, we propose a nonnegative matrix factorization-based algorithm for link prediction in multi-relational networks. The framework of the algorithm is as follows:

Algorithm LPMR (link prediction in multi-relational networks)

```
input:
$G = (V,E) = G_1 \cup G_2 \cup_{...} \cup G_d$: the network, where $G_t$ is the sub-graph of the $t$-th type of edges;
$G_1$ : the target sub-graph;
$k$: dimension of the latent space;
output:
$S$: similarity score matrix of link prediction in $G_1$;
Begin
1. For $i = 1$ to $d$ do
2.     Compute the belief vector $b_i$ for sub-graph $G_i$;
3. End for $i$
4. $r_{11} = 1$;
5. For $i = 2$ to $d$ do
6.     Compute the similarity $r_{1j}$ between $G_1$ and $G_i$;
7. End for;
```

(continued on next page)

```
8. $M = \sum_{t=1}^{d} r_{1t}$;
9. Initialize the matrices $U^*$, $V^*$,$U^{(t)}$ and $V^{(t)}$, for $t=1,\ldots,d$; $N=1$;
10. While (not convergence) and ($N < Nmax$) do
11.   for $t=1$ to $d$ do
12.     compute $A_V^{(t)} = A^{(t)}V^{(t)}$, $A_V^{(t)} = U^{(t)}{V^{(t)}}^T V^{(t)}$;
13.     for each element $u_{ij}^{(t)}$ in $U^{(t)}$ do
14.       $u_{ij}^{(t)} \leftarrow u_{ij}^{(t)} \frac{(A_V^{(t)})_{ij}+u_{ij}^*}{(A_V^{(t)})_{ij}+u_{ij}^{(t)}}$;
15.     End for;
16.     compute $A_U^{(t)} = U^{(t)T}A^{(t)}$, $A_U^{(t)} = U^{(t)T}U^{(t)}{V^{(t)}}^T$;
17.     for each element $v_{ij}^{(t)}$ in $V^{(t)}$ do
18.       $v_{ij}^{(t)} \leftarrow v_{ij}^{(t)} \frac{(A_U^{(t)})_{ij\,ij}^T+v_{ij}^*}{(A_U^{(t)})_{ij}^T+v_{ij}^{(t)}}$;
19.     End for;
20.   End for $t$;
21.   $U^* = \frac{1}{M}\sum_{t=1}^{T} r_{1t}U^{(t)}$;
22.   $V^* = \frac{1}{M}\sum_{t=1}^{T} r_{1t}V^{(t)}$;
23.   $N=N+1$
24. End while;
25. For $i=1$ to $n$ do
26.   For $j=1$ to $n$ do
27.     $S(i,j)$=similarity between the $i^{\text{th}}$ and $j^{\text{th}}$ row vectors in $V^*$;
28.   End for $j$
29. End for $i$;
30. Output($S$)
End
```

In the algorithm, lines 1 to 3 calculate the belief vectors for the sub-graphs according to (3). Based on the belief vectors of the sub-graphs, lines 4 to 7 compute the similarities of the sub-graphs with the target sub-graph G_1 according to (4). Line 9 initializes matrices U^*, V^*,$U^{(t)}$ and $V^{(t)}$, $t=1,\ldots,d$. For each matrix, we generate a random nonnegative matrix and use it as the initial value of the matrix. Lines 12 to 15 fix the matrices $V^{(t)}$, U^*, V^* and update $U^{(t)}$ according to (11). Lines 16 to 19 fix the matrices $U^{(t)}$, U^*, V^* and update $V^{(t)}$ according to (15). Line 21 fixes the matrices $U^{(t)}$, V^*, $V^{(t)}$ and updates U^* according to (19). Line 22 fixes the matrices $U^{(t)}$, U^*, $V^{(t)}$ and updates V^* according to (22). Such iteration is repeated until the values of the matrixes converge or the number of iterations exceeds a predefined maximum number *Nmax.* In lines 25 to 29, the similarities between the row vectors in V^* are computed and stored in the matrix S as the final score of link prediction. Similarity measures such as correlation coefficients and cosine similarity can be used in computing the similarity scores.

4.4. Time complexity analysis

In the algorithm, it takes $O(d.n^2)$ time for lines 1 to 3 to calculate the belief vectors for d sub-graphs. Lines 4 to 7 require $O(d.n)$ time to compute the similarities of the sub-graphs with G_1. In each iteration, lines 12 and 16 require $O(n^2.k)$ time for matrix multiplications. Lines 13 to 15 and lines 17 to 19 require $O(d.n.k)$ time to update the elements in matrixes $U^{(t)}$ and $V^{(t)}$; here, d is the number of different types of links. Because each row in matrix V^* is a k-dimensional vector, it takes $O(k)$ time to compute the similarity between such vectors. Therefore, lines 24 to 28 require $O(n^2.k)$ time to compute the similarities for all pairs of the row vectors in V^*. Because k and d can be treated as constants, the complexity of the algorithm is $O(n^2)$. In the similarity-based link prediction methods, $n(n-1)/2$ similarity scores between the node pairs must be computed. Therefore, $O(n^2)$ is the lower bound of the time complexity of the similarity-based link prediction methods.

4.5. Convergence and correctness analysis

In this section, we first prove the convergence of iterations for updating *U, V, U** and *V** by the LPMR algorithm and then show the correctness of the converged solutions. In our proof, we employ the following lemmas by D. Lee *et al* [27]..

Lemma 1. [27] *Let* $F \in \mathbb{R}_+^{n\times n}$, $G \in \mathbb{R}_+^{k\times k}$ *be symmetric matrices, and let* $S \in \mathbb{R}_+^{n\times k}$, $S' \in \mathbb{R}_+^{n\times k}$ *be two* $n \times k$ *matrices. The following inequality holds.*

$$\sum_{i=1}^{n}\sum_{j=1}^{k}\frac{\left(FS'G\right)_{ij}S_{ij}^2}{S'_{ij}} \geq \mathrm{Tr}\left[SF^TSG\right]. \tag{23}$$

Proof. Let $S_{ij} = S'_{ij}p_{ij}$. Using the explicit index, the difference between the LHS and RHS of (23) is,

$$\Delta = \sum_{i,x=1}^{n}\sum_{j,y=1}^{k} F_{ix}S'_{xy}G_{yj}S'_{ij}\left(p_{ij}^2 - p_{ij}p_{xy}\right)$$

Because F and G are symmetric matrices, this is equal to

$$\begin{aligned}\Delta &= \sum_{i,x=1}^{n}\sum_{j,y=1}^{k} F_{ix}S'_{xy}G_{yj}S'_{ij}\left(\frac{p_{ij}^2+p_{xy}^2}{2}-p_{ij}p_{xy}\right)\\ &= \frac{1}{2}\sum_{i,x=1}^{n}\sum_{j,y=1}^{k} F_{ix}S'_{xy}G_{yj}S'_{ij}\left(p_{ij}-p_{xy}\right)^2\\ &\geq 0.\end{aligned}$$

Q.E.D.

Definition 1. [27] Let$L(H)$ be a function of matrix H. $Z(H,\tilde{H})$ is an auxiliary function for $L(H)$ if it satisfies

$$Z(H,\tilde{H}) \geq L(H), \qquad Z(H,H) = L(H)\,\text{for any}\,H,\ \tilde{H}.$$

Lemma 2. [27] *If Z is an auxiliary function for L, then L is non-increasing under the update rule:*

$$H(\tau+1) = \arg\min_{H} Z(H, H(\tau)) \tag{24}$$

Proof. By Definition 1 we have,

$$L(H(\tau+1)) \leq Z(H(\tau+1), H(\tau)) \leq Z(H(\tau), H(\tau)) = L(H(\tau)) \tag{25}$$

Q.E.D

Based on the abovementioned lemmas, we prove the convergence and correctness of the LPMR algorithm in the following theorems.

Theorem 1. Fixing any three matrices in $U^{(t)}$ $V^{(t)}$,V^* and U^*, and using the updating rules (11), (15), (19) and (22) in each iteration of algorithm MF, the value of objective function J is non-increasing.

Proof. When we fix matrices $V^{(t)}$,V^* and U^* to update $U^{(t)}$to minimize the object function J in (7), the formula for $U^{(t)}$ is updated by minimizing the objective function of $J(U^{(t)})$ given in (8). $J(U^{(t)})$ can be rewritten as

$$J(U^{(t)}) = \mathrm{Tr}\left[r_{1t}(A^{(t)}-U^{(t)}V^{(t)^T})(A^{(t)}-U^{(t)}V^{(t)^T})^T + r_{1t}(U^{(t)}-U^*)(U^{(t)}-U^*)^T\right]$$

It is easy to obtain

$$J(U^{(t)}) = r_{1t}\mathrm{Tr}\left[A^{(t)^T}A^{(t)} - 2U^{(t)^T}A^{(t)}V^{(t)} + V^{(t)^T}V^{(t)}U^{(t)^T}U^{(t)} - 2U^{(t)^T}U^* + U^{*T}U^*\right] \tag{26}$$

Ignoring constant terms with respect to $U^{(t)}$, objective function (26) can be equivalently replaced by

$$L(U^{(t)}) = \mathrm{Tr}\left[-2U^{(t)^T}\left(A^{(t)}V^{(t)}+U^*\right) + \left(V^{(t)^T}V^{(t)}+I\right)U^{(t)^T}U^{(t)}\right] \tag{27}$$

Let

$$Z\left(U^{(t)},\tilde{U}^{(t)}\right) = -2\sum_{ij}\left(\left(A^{(t)}V^{(t)}\right)_{ij}+U^*_{ij}\right)U^{(t)}_{ij} + \sum_{ij}\frac{\left(\tilde{U}^{(t)}\left(V^{(t)^T}V^{(t)}+I\right)\right)_{ij}U^{(t)^2}_{ij}}{\tilde{U}^{(t)}_{ij}} \tag{28}$$

Then, $Z(U^{(t)},\tilde{U}^{(t)})$ is an auxiliary function for $L(U^{(t)})$. First, if $\tilde{U}^{(t)} = U^{(t)}$, it is obvious that $Z(U^{(t)},\tilde{U}^{(t)}) = L(U^{(t)})$. By Lemma 1, the second term in (27) is always less than that in (28), so the inequality $Z(U,\ U') \geq L(U)$ holds. Therefore, $Z(U^{(t)},\tilde{U}^{(t)})$ is an auxiliary function for$L(U^{(t)})$.

By the definition of the auxiliary function and Lemma 2, we know that if we find the $U^{(t)}$value to reach the local minimum of auxiliary function $Z(U^{(t)},\tilde{U}^{(t)})$, then the value of $L(U^{(t)})$ is non-increasing over iterations. Therefore, we find that the $U^{(t)}$value minimizes $Z(U^{(t)},\tilde{U}^{(t)})$ by fixing $\tilde{U}^{(t)}$.

Because

$$\frac{\partial Z}{\partial U^{(t)}_{ij}} = -2\left(\left(A^{(t)}V^{(t)}\right)_{ij}+U^*_{ij}\right) + 2\frac{\left(\tilde{U}^{(t)}\left(V^{(t)^T}V^{(t)}+I\right)\right)_{ij}U^{(t)}_{ij}}{\tilde{U}^{(t)}_{ij}},$$

let $\frac{\partial Z}{\partial U^{(t)}_{ij}} = 0$. We know that the value of $U^{(t)}_{ij}$for minimizing $Z(U^{(t)},\tilde{U}^{(t)})$ is

$$U^{(t)}_{ij} = \tilde{U}^{(t)}_{ij}\frac{\left(A^{(t)}V^{(t)}\right)_{ij}+U^*_{ij}}{\left(\tilde{U}^{(t)}V^{(t)^T}V^{(t)}\right)_{ij}+\tilde{U}^{(t)}_{ij}} \tag{29}$$

Table 1
Number of nodes and edges in tested YouTube data.

YouTube					
Nodes	Edges of different types				
Users	CNU	FR	SBN	SBR	VID
15,088	76,765	1940,806	2239,440	5574,249	3797,635

According to (24), we can obtain updating rule (11) from (29) after replacing $U^{(t)}$in (29) by $U^{(t)}(\tau+1)$ and $\tilde{U}^{(t)}$ by $U^{(t)}(\tau)$. Therefore, using (11) in each iteration of the MF algorithm, the value of objective function J is non-increasing.

In a similar way, we can prove that the value of objective function J is also non-increasing by using rules (15), (19), and (22) in each iteration of the LPMR algorithm to update $V_{ij}^{(t)}$,$U^*$and V^*, respectively.

Q.E.D.

Theorem 2. The values of $U_{ij}^{(t)}$,$V_{ij}^{(t)}$,$U^*$and V^*obtained by using updating rules (11), (15), (19), and (22) can converge to the correct solution for the optimization problem (7).

Proof. From (24), we know that the value of $U_{ij}^{(t)}$by (29) is also non-increasing. Because $U_{ij}^{(t)} > 0$, it converges. The correctness of the converged solution is ensured by the fact that at convergence, from (29), the solution will satisfy

$$2\left(-A^{(t)}V^{(t)} - U^* + U^{(t)}V^{(t)^T}V^{(t)} + U^{(t)}\right)_{ij} U^{(t)}_{ij} = 0 \tag{30}$$

It is the same as the fixed point condition of (10), which is also the fixed point condition of (7). The convergence and correctness of updating the $V^{(t)}$ formula in (15) can be proven in a similar way.

To show the convergence of U^*, using the updating formula for U^* in (19), we can obtain

$$U^*(\tau+1) - U^*(\tau) = \frac{1}{\sum\limits_{t=1}^{T}\lambda^{T-t}} \sum_{t=1}^{T} \lambda^{T-t}\left(U^{(t)}(\tau+1) - U^{(t)}(\tau)\right) \tag{31}$$

Because $U^{(t)}$is non-increasing and converges, namely, $U^{(t)}(\tau+1) - U^{(t)}(\tau) \le 0$, then from (31), $precision = \frac{m}{L}$ is non-increasing and converges as well. Because updating the formula of U^* in (19) satisfies the KKT condition, U^* converged to the optimal value minimizes the objective function given in (18). This shows the correctness of U^*obtained after convergence of the iterations. The convergence and correctness proofs of V^* can be similarly derived.

Q.E.D.

5. Experimental results

5.1. Experimental environment

To evaluate the proposed LPMR algorithm for link prediction in multi-relational networks, we test it by conducting a series of experiments on several data sets of networks. We also compare the results given by LPMR with the other four algorithms: CN (Common Neighbors), JC (Jaccard coefficient), PA (Preferential Attachment), and AA (Adamic/Adar). In the test of these algorithms, the multi-relational networks are integrated as a single-relational network. All of the experiments are performed on a Pentium IV computer running Windows XP, with 1.7 G memory, and using VC++ 6. 0.

5.2. Data sets

We use three real-world heterogeneous information networks to test the presented method. The networks were chosen from different domains and have diversity in structure and relationship types, which will support the generality of our conclusions.

5.2.1. YouTube network [54]

The YouTube network is constructed from data crawled on the popular video sharing site in December 2008. The crawl collected information about contacts, favorite videos, and subscriptions. In total, it reached 848,003 users, with 15,088 users sharing all of the information types. These 15,088 users are the nodes in the social network, connected by a network of five different interaction types. These are the contact network of the user (CNU), shared contact with users outside of the network (FR), shared subscriptions (SBN), shared subscribers (SBR), and shared favorite videos (VID). The basic statistics of the tested YouTube data can be found in Table 1.

Table 2
Number of nodes and edges in the tested Disease-Gene Network.

Disease-Gene					
Nodes		Edges of different types			
Diseases	Genes	G	P	PPI	F
703	1132	10,483	74,523	2450	3279

Table 3
Number of nodes and edges in the Climate Network.

Climate Network							
Nodes	Edges of different types						
Locations	GH	VWS	PW	RH	SST	SLP	HWS
1701	249,322	71,458	50,835	25,375	132,469	175,786	31,615

5.2.2. *Disease-gene network* [55]

The disease-gene (DG) network was constructed from three individual data sets. As the name suggests, this network has two distinct node types—diseases and genes—with four edge types connecting them. The diseases are classified by Disease Ontology (DO) codes, and the gene names are based on the HUGO Gene Nomenclature. Genetic association (G) links exist between diseases and genes in a bipartite fashion and represent known disease-gene associations extracted from the Online Mendelian Inheritance in Man (OMIM) database, SwissProt, and the Human Protein Reference Database (HPRD). Protein-protein interaction (PPI) links connect pairs of genes in accordance with combined physical interaction data collected from HPRD, the Online Predicted Human Interaction Database (OPHID). Additionally, disease pairs are connected by a phenotypic (P) link if they are significantly co-morbid in real patients. The disease-gene network consists of 703 diseases and 1132 genes. The basic statistics of the tested Disease-Gene Network can be found in Table 2.

5.2.3. *Climate network* [56]

The climate network we tested in the in the experiment is constructed from 5*5 latitude-longitude gridded climate data, where each node is a physical location and edges represent similarity with respect to one of the seven climate variables. The variables included are temperature (SST), sea level pressure (SLP), horizontal (HWS) and vertical (VWS) wind speed, precipitable water (PW), relative humidity (RH), and geopotential height (GH), each of which is represented by a distinct edge type. In the data preprocessing, loose connected links are deleted according to the nodes similarity measured in terms of Pearson correlation. By experiments, we found 0.3 is suitable to be a threshold value. Every edge type can overlap with every other, and a single node pair may have up to seven edges. Overall, the climate network included 1701 location nodes. Each node is connected to others by different types of edges. Topological features of the Climate Network are provided in Table 3.

5.3. *Measurement of the prediction results*

In the test, we use AUC score, precision, recall and F-measure to evaluate the quality of the results given by LPMR and the other algorithms.

5.3.1. *AUC score*

The AUC (Area under Curve) score is a commonly used measurement to evaluate the quality of the results of link prediction. After the algorithms calculate and rank the similarities of all of the node pairs, which represent all of the existent and nonexistent links, the AUC value can be interpreted as the probability that a randomly chosen existing link is given a higher score than a randomly chosen non-existing link. At each time, we randomly pick an existing link and a non-existing link to compare their similarity scores. If among n independent comparisons there are n' times the existing links have higher scores and n'' times they have the same score, the AUC value is

$$AUC = (n' + 0.5n'')/n \tag{32}$$

In general, a larger AUC value indicates higher performance. Hence, the AUC value of the perfect result is 1.0, while the AUC of the result by a random predictor is 0.5. To evaluate the accuracy of the results, a random 10-fold cross validation (CV) is used in our experiments. In 10-fold cross-validation, the original links are randomly partitioned into 10 subsets. Of the 10 subsets, a single subset is retained as the validation data for testing the algorithms, and the remaining 9 subsets are used as training data. The cross-validation process is then repeated 10 times. The 10 results from the folds are averaged to produce the final estimation.

Table 4
Comparison of AUC scores by LPMR with other methods for YouTube.

	CN	JC	PA	AA	LPMR
CNU	0.783	0.781	<u>0.865</u>	0.784	**0.947**
FR	0.984	**0.988**	0.934	0.986	<u>0.985</u>
SBN	0.98	<u>0.982</u>	0.944	0.981	**0.983**
SBR	<u>0.988</u>	0.987	0.946	0.985	**0.989**
VID	0.971	0.968	0.957	<u>0.973</u>	**0.975**

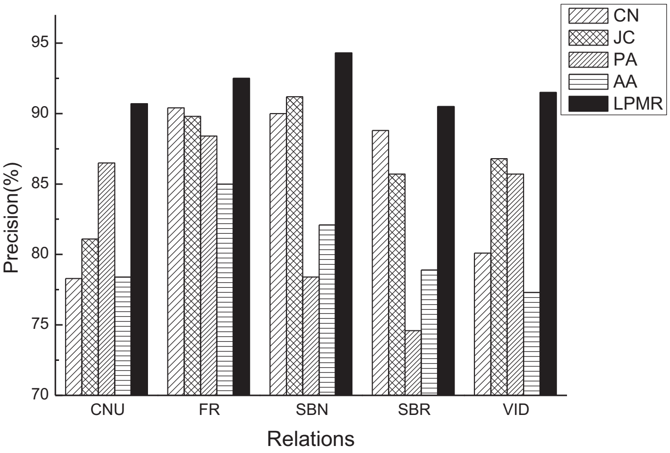

Fig. 1. Precisions of the five algorithms on YouTube.

5.3.2. Precision

Let V_L be the set of node pairs that have the top-L highest scores. If only m node pairs in V_L represent existing edges, the precision of the result is defined as:

$$precision = \frac{m}{L} \tag{33}$$

5.3.3. Recall

Let $M = |E|$ be the number of existing edges in the network. Suppose m existing edges in E are predicted by the algorithm; the recall of the result is defined by:

$$recall = \frac{m}{M} \tag{34}$$

5.3.4. F-measurement

Based on precision and recall of a link prediction result, the F-measurement can be defined as:

$$F = \frac{2 \times precision \times recall}{precision + recall} \tag{35}$$

Because F-measurement combines precision and recall collectively, it is a comprehensive measurement for the quality of the link prediction results.

5.4. Experimental results

5.4.1. Test results for YouTube

The AUC values of the YouTube dataset test results given by LPMR and the other algorithms are shown in Table 4. In the table, each row shows the results of tests using one type of link as the target relation. On each row, the highest AUC score by the five algorithms is emphasized in bold-face, and the second-highest AUC score is underlined.

As shown in Table 4, we can see that of the five algorithms, LPMR obtains the highest AUC scores on five of the relations and obtains the second-highest AUC in predicting the relation FR. For example, on the relation CNU, the LPMR algorithm receives the highest AUC score of 0.947, which is much higher than the second-highest score, 0.781. In predicting the relation FR, LPMR obtains the second-highest AUC score of 0.985, which is slightly lower than the highest score, 0.988. This shows that the algorithm can achieve high-quality results.

Precision, recall, and the F-measures of the five algorithms on the YouTube dataset are shown in Fig. 1 to Fig. 3, respectively.

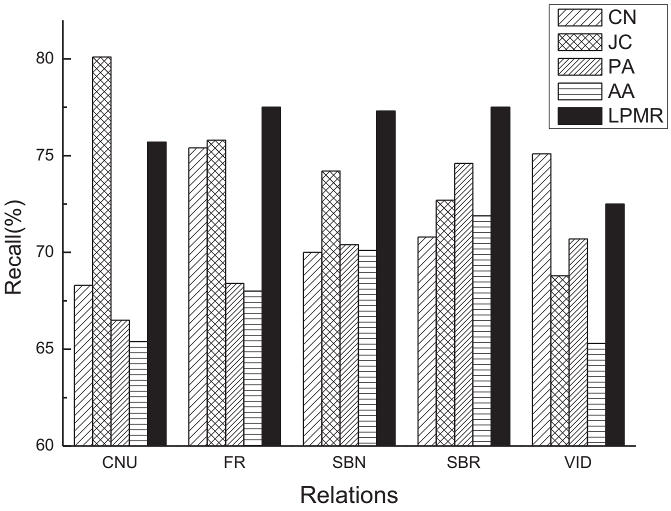

Fig. 2. Recalls of the five algorithms on YouTube.

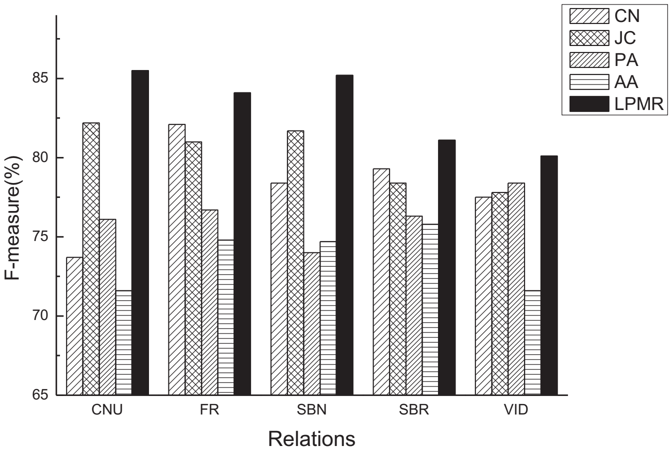

Fig. 3. F-measures of the five algorithms on YouTube.

From Fig. 1, we can see that LPMR obtains the highest precision among the five algorithms in predicting all of the relations in YouTube. Fig. 2 shows that the LPMR recalls are higher than those of all the other algorithms in the FR, SBN and SBR relations. For the CNU and VID relations, the LPMR recalls are the second highest, being slightly lower than those of JC and CN, respectively. This is because the measurement of recall only considers the percentage of detected links among the existing ones. For the sparsely connected link types, our algorithm may produce slightly lower recalls. However, the overall quality of LPMR is higher than that of the other algorithms. This overall quality can be evaluated by the F-measure. Fig. 3 shows that the LPMR's F-measure values are the highest among these algorithms in predicting all of the relations. For instance, the LPMR's F-measure values on the relations are 5% to 12% higher than those of AA.

We use a *t*-test to show that the LPMR F-measures are significantly different from those of the other algorithms. Table 5 shows the F-measures of the results of 18 tests using different algorithms on YouTube-FR. We randomly divide the dataset into 18 parts. In each test, one part is used as the test set, and the rest are used as the training set.

Table 6 compares the LPMR F-measure *t*-values with those of the other algorithms.

Because each group of data has 18 samples, the degree of freedom in the *t*-test is (18–1) *2 = 34. We set the significance level α=0.05, and the confidence level p is 97.5%. From the *t*-distribution table, we can obtain $t_{0.975}(34)$= 2.03224. From Table 6, we can see that all of the t values are greater than $\mathrm{t}_{0.975}(34)$. This indicates that there are significant differences between the LPMR F-measures and those of the other algorithms. Therefore, the quality of the results given by our LPMR algorithm is obviously higher than those given by the other algorithms.

Table 5
Comparison of the F-measures in 18 tests of the results given by different algorithms on YouTube-FR.

No	CN	JC	PA	AA	LPMR
1	0.825	0.827	0.771	0.753	0.838
2	0.837	0.787	0.797	0.725	0.828
3	0.821	0.815	0.777	0.751	0.856
4	0.823	0.800	0.760	0.738	0.841
5	0.831	0.825	0.777	0.753	0.846
6	0.799	0.829	0.735	0.763	0.857
7	0.829	0.802	0.785	0.754	0.823
8	0.813	0.780	0.759	0.718	0.824
9	0.817	0.819	0.762	0.757	0.850
10	0.830	0.805	0.776	0.743	0.837
11	0.794	0.802	0.737	0.761	0.833
12	0.813	0.818	0.759	0.755	0.839
13	0.837	0.788	0.796	0.716	0.829
14	0.826	0.820	0.772	0.758	0.851
15	0.821	0.803	0.757	0.741	0.843
16	0.836	0.820	0.770	0.758	0.841
17	0.798	0.826	0.744	0.768	0.857
18	0.828	0.814	0.772	0.752	0.845

Table 6
t-values of F-measures given by LPMR compared with other algorithms on YouTube-FR.

Algorithm	CN	JC	PA	AA
t-values compared with LPMR	4.9752	6.9408	15.2987	21.2814

Table 7
Comparison of LPMR AUC scores with other methods on Disease-Gene Network.

	CN	JC	PA	AA	LPMR
G	0.951	0.957	0.903	0.956	**0.972**
P	0.909	0.771	**0.943**	0.911	0.938
PPI	0.788	0.786	0.827	0.789	**0.831**

5.4.2. *Test results on disease-gene network*

The AUC values of the test results on the Disease-Gene Network given by LPMR and the other algorithms are shown in Table 7.

As shown in Table 7, we can see that among all five algorithms, LPMR has the highest AUC scores on two of the relations and obtains the second-highest AUC in predicting relation P. For example, on the relation G, the LPMR algorithm obtains an AUC score of 0.972, which is much higher than the second-highest score of 0.957. In predicting relation P, LPMR obtains the second-highest AUC score of 0.938, which is slightly lower than the highest score, 0.943.

A comparison of the precision, recall, and F-measures of the test results given by the algorithms on the Disease-Gene Network are shown in Figs. 4–6, respectively.

From Fig. 4, we can see that LPMR can obtain results with the highest precision among the five algorithms on predicting relations G and P in the Disease-Gene Network. For relation PPI, the LPMR's precision is the second highest, slightly lower than that of PA. Fig. 5 shows that the LPMR's recalls are higher than those of all of the other algorithms in the relations of G and P. For the relation of PPI, the recall by LPMR is the second highest, slightly lower than that of CN. From Fig. 6, we can see that the F-measure values of LPMR are the highest among these algorithms in predicting all of the relations. For instance, the LPMR's F-measure values on the relations are approximately 10% higher than those of JC.

We use the *t*-test to show that the LPMR's F-measures by are significantly different from those of the other algorithms. Table 8 shows the F-measures in 18 tests of the results given by different algorithms on Disease-Gene Network-P. Table 9 compares the *t*-value results of the F-measures given by LPMR with those of the other algorithms.

We set the significance level α=0.05, and the confidence level p is 97.5%. From Table 9, we can see that all of the t values are greater than $t_{0.975}(34) = 2.03224$. This indicates that there are significant differences between the F-measures given by LPMR and those of the other algorithms.

5.4.3. *Test results on climate network*

The LPMR's AUC values for the test results on the Climate Network are compared with those of the other algorithms in Table 10.

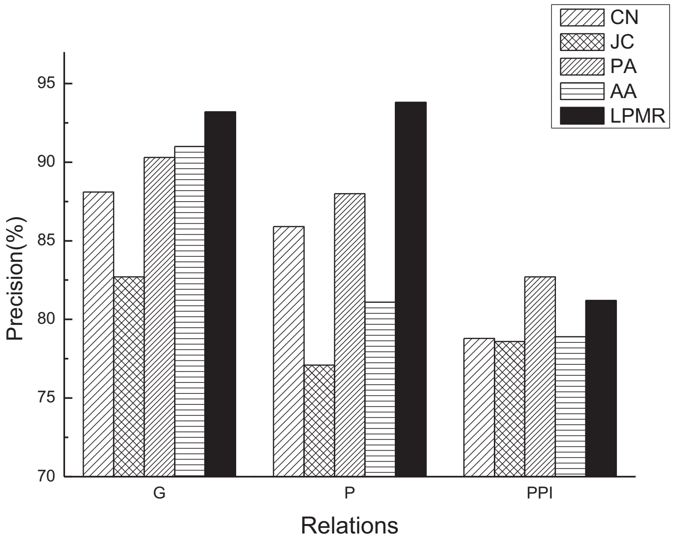

Fig. 4. Precisions of the five algorithms on the Disease-Gene Network.

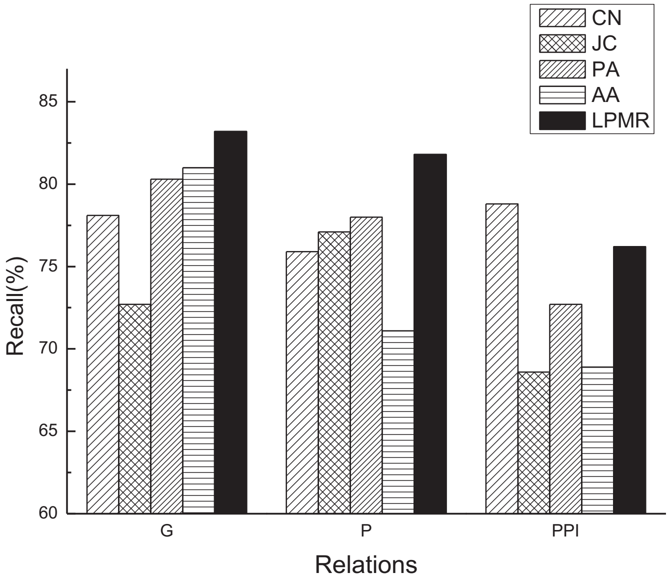

Fig. 5. Recalls of the five algorithms on the Disease-Gene Network.

We can see from Table 10 that of the five algorithms, LPMR has the highest or the second-highest AUC scores on all of the relations. For example, on the relation GH, the LPMR algorithm obtains an AUC score 0.992, which is much higher than the second-highest score of 0.987. In predicting relation RH, LPMR obtains the second-highest AUC score of 0.994, which is slightly lower than the highest one, 0.995.

A comparison of the precision, recall, and F-measures of the test results by the algorithms on the Climate Network are shown in Figs. 7–9, respectively.

Fig. 7 shows that LPMR achieves the highest or second-highest precision among the five algorithms in predicting all of the relations in the Climate Network. Fig. 8 shows that LPMR's recalls are higher than those of all of the other algorithms in five relations and obtains the second-highest recall in two relations. For the PW and SLP relations, the LPMR's recalls are slightly lower than the highest one. From Fig. 6, we can see that the F-measure values of LPMR are the highest among these algorithms on predicting all of the relations. For instance, the F-measure value of LPMR on the relations is 3% to 10% higher than those of CN.

It is worth mentioning that in the tests on the relations of all datasets, LPMR's F-measures are always the highest among the five algorithms. Because the F-measure is a good tradeoff between recall and precision, it can reflect the overall quality

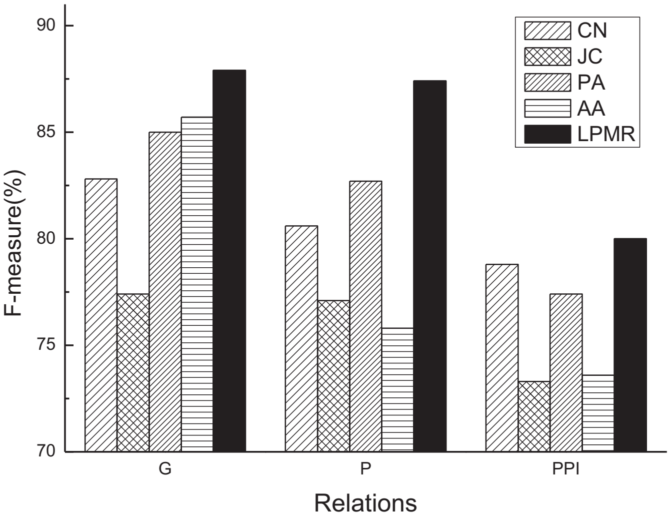

Fig. 6. F-measures of the five algorithms on the Disease-Gene Network.

Table 8
Comparison of the F-measures in 18 tests of the results given by different algorithms on Disease-Gene Network-P.

No	CN	JC	PA	AA	LPMR
1	0.791	0.775	0.821	0.773	0.888
2	0.825	0.787	0.853	0.735	0.860
3	0.799	0.785	0.817	0.751	0.874
4	0.81	0.764	0.839	0.748	0.876
5	0.807	0.781	0.837	0.783	0.865
6	0.821	0.739	0.815	0.753	0.882
7	0.790	0.779	0.824	0.764	0.876
8	0.790	0.763	0.818	0.748	0.872
9	0.814	0.766	0.823	0.767	0.871
10	0.793	0.780	0.826	0.753	0.860
11	0.803	0.741	0.810	0.771	0.867
12	0.788	0.783	0.829	0.765	0.876
13	0.818	0.800	0.833	0.726	0.878
14	0.816	0.776	0.842	0.758	0.879
15	0.811	0.761	0.817	0.751	0.877
16	0.801	0.764	0.842	0.778	0.878
17	0.811	0.758	0.806	0.748	0.874
18	0.82	0.776	0.834	0.772	0.879

Table 9
t-values of F-measures given by LPMR compared with other algorithms on Disease-Gene Network-P.

Algorithm	CN	JC	PA	AA
t-values compared with LPMR	20.4999	25.4173	13.7486	29.6528

Table 10
Comparison of the AUC scores given by LPMR with those of the other methods on the Climate Network.

	CN	JC	PA	AA	LPMR
GH	0.985	0.987	0.783	0.986	**0.992**
VWS	0.935	**0.954**	0.802	0.942	0.949
PW	0.990	**0.995**	0.717	0.992	**0.995**
RH	0.992	**0.995**	0.681	0.993	0.994
SST	0.956	0.973	0.776	0.962	**0.975**
SLP	0.979	0.985	0.698	0.981	**0.988**
HWS	0.990	0.973	0.731	0.992	**0.995**

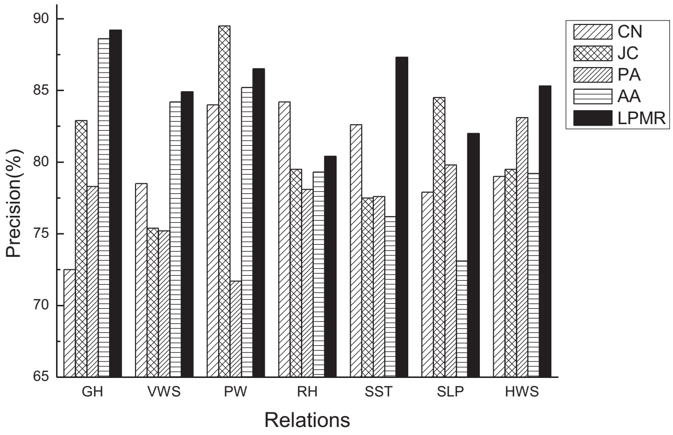

Fig. 7. Precisions of the five algorithms on Climate.

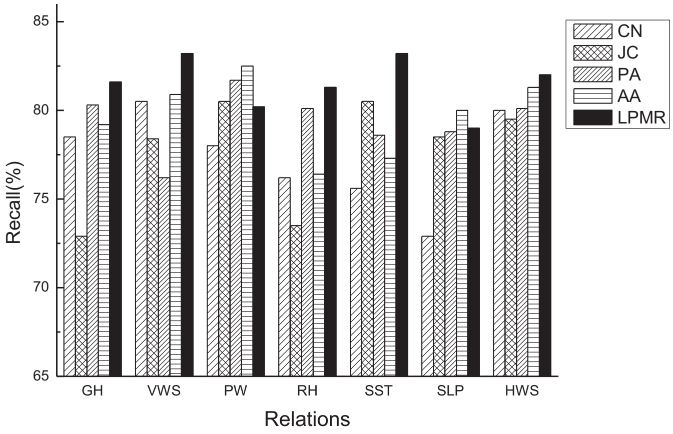

Fig. 8. Recalls of the five algorithms on Climate.

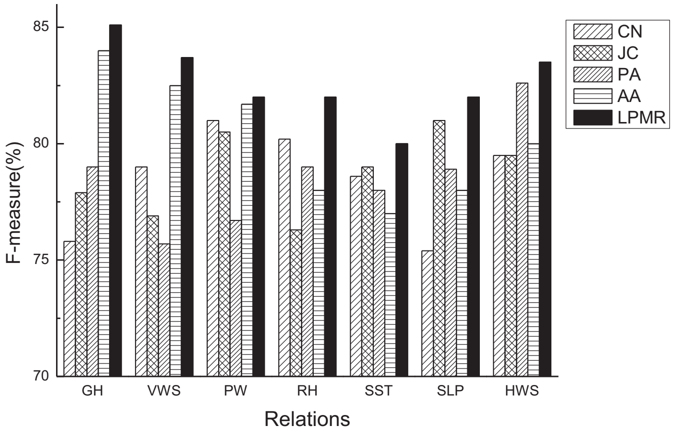

Fig. 9. F-measures of the five algorithms on Climate.

of the results. Therefore, we can see from the experiments that LPMR can obtain higher-quality prediction results than the other algorithms on the networks with multi-relations.

We use the *t*-test to show that the LPMR's F-measures are significantly different from those of the other algorithms. Table 11 shows the F-measures in 18 tests of the results given by the different algorithms on Climate-PW. Table 12 compares the results of the LPMR's F-measure *t*-values with those of the other algorithms.

Table 11
Comparison of the F-measures of the results given by different algorithms on Climate-PW.

No	CN	JC	PA	AA	LPMR
1	0.8	0.79	0.777	0.81	0.824
2	0.824	0.824	0.797	0.818	0.826
3	0.803	0.798	0.772	0.807	0.82
4	0.819	0.81	0.765	0.819	0.822
5	0.806	0.805	0.777	0.808	0.81
6	0.825	0.82	0.735	0.824	0.828
7	0.795	0.79	0.765	0.8	0.822
8	0.812	0.79	0.759	0.814	0.818
9	0.809	0.811	0.772	0.808	0.816
10	0.79	0.782	0.776	0.795	0.809
11	0.813	0.812	0.737	0.815	0.813
12	0.792	0.787	0.769	0.794	0.812
13	0.833	0.827	0.78	0.835	0.836
14	0.81	0.805	0.772	0.812	0.825
15	0.815	0.81	0.757	0.817	0.82
16	0.795	0.8	0.77	0.8	0.815
17	0.815	0.81	0.751	0.814	0.817
18	0.824	0.819	0.775	0.826	0.827

Table 12
t-values of F-measures given by LPMR compared with the other algorithms on Climate-PW.

Algorithm	CN	JC	PA	AA
t-values compared with LPMR	2.9682	5.1043	12.83697	2.6431

We set the significance level α=0.05, and the confidence level p is 97.5%. Table 12 shows that all of the t values are greater than $t_{0.975}(34) = 2.03224$. This indicates that there are significant differences between the LPMR's F-measures and those of the other algorithms. Therefore, the quality of results given by our LPMR algorithm is obviously higher than that of the other algorithms.

6. Conclusions

In this paper, we investigate the problem of link prediction in multi-relational networks. We propose a multi-relational network link prediction algorithm called LPMR that is based on relational similarity. In the algorithm, we consider the similarity and influence between different relations. We first present a belief propagation method to calculate the belief of each node in each type of relation and then construct the belief vector for each type of link. We use the similarity between belief vectors to measure the influence between different types of relations. Based on the influence between different relations, we present a nonnegative matrix factorization-based LPMR algorithm for link prediction in multi-relational networks. The convergence and correctness of the algorithm are proved. Our experimental results show that LPMR can achieve higher-quality prediction results than other similar algorithms.

Acknowledgements

This research was supported in part by the Chinese National Natural Science Foundation under grant Nos. 61379066, 61070047, 61379064, and 61472344; the Natural Science Foundation of Jiangsu Province under contracts BK20130452, BK2012672, and BK2012128; and the Natural Science Foundation of the Education Department of Jiangsu Province under contracts 12KJB520019, 13KJB520026, and 09KJB20013.

References

[1] N.M. Ahmed, L. Chen, An efficient algorithm for link prediction in temporal uncertain social networks, Inf. Sci. 331 (2016) 120–136.
[2] W. Ahn, W.S. Jung, Accuracy test for link prediction in terms of similarity index: the case of WS and BA models, Physica A 429 (1) (2015) 177–183.
[3] L.M. Aiello, A. Barrat, R. Schifanella, et al., Friendship prediction and homophily in social media, ACM Trans. Web (TWEB) 6 (2) (2012) 9.
[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, 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] M. Berlingerio, M. Coscia, F. Giannotti, A. Monreale, D. Pedreschi, Foundations of multidimensional network analysis, in: ASONAM, 2011, pp. 485–489.
[7] M.H. Bhuyan, D.K. Bhattacharyya, J.K. Kalita, A multi-step outlier-based anomaly detection approach to network-wide traffic, Inf. Sci. 348 (2016) 243–271.

[8] 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) (2014) 750–764.
[9] B. Bringmann, M. Berlingerio, F. Bonchi, A. Gionis, Learning and predicting the evolution of social networks, in: IEEE Intelligent Systems, 2010, pp. 26–34.
[10] F. Buccafurri, L. Fotia, G. Lax, et al., Analysis-preserving protection of user privacy against information leak age of social-network Likes, Inf. Sci. 328 (2016) 340–358.
[11] F. Buccafurri, G. Lax, A. Nocera, D. Ursino, Discovering missing me edges across social networks, Inf. Sci. 319 (2015) 18–37.
[12] B.L. Chen, L. Chen, B. Li, A fast algorithm for predicting links to nodes of interest, Inf. Sci. 329 (2016) 552–567.
[13] D. Darcy, L. Ryan, N.V. Chawla, Multi-relational link prediction in heterogeneous information networks, in: Proceedings of 2011 International Conference on Advances in Social Networks Analysis and Mining, 2011, pp. 281–288.
[14] J.Y. Ding, L.C. Jiao, J.S. Wu, Y.T. Hou, Y.T. Qi, Prediction of missing links based on multi-resolution community division, Physica A 417 (1) (2015) 76–85.
[15] J. Fournet, A. Barrat, Contact patterns among high school students, PLoS One 9 (9) (2014) 107878.
[16] S. Gao, L. Denoyer, P. Gallinari, Temporal link prediction by integrating content and structure information, in: Proceedings of CIKM'11, 2011, pp. 1169–1174.
[17] M.Q. Ge, A. Li, M.H. Wang, A bipartite network-based method for prediction of long non-coding RNA–protein interactions, Genomics Proteomics Bioinf. 14 (1) (2016) 62–71.
[18] R. Guimera, M. Sales-Pardo, Missing and spurious interactions and the reconstruction of complex networks, Proc. Natl. Acad. Sci. 106 (52) (2010) 22073–22078.
[19] Y.L. He, J.N.K. Liu, Y.-x. Hu, Xi-z. Wang, OWA operator based link prediction ensemble for social network, Expert Syst. Appl. 42 (1) (2015) 21–50.
[20] M. Hoffman, D. Steinley, M.J. Brusco, A note on using the adjusted Rand index for link prediction in networks, Soc. Netw. 42 (2015) 72–79.
[21] F.Y. Hu, H.S. Wong, Labeling of human motion based on CBGA and probabilistic model, Int. J. Smart Sens. Intell. Syst. 6 (2) (2013) 583–609.
[22] N.M.A. Ibrahim, L. Chen, Link prediction in dynamic social networks by integrating different types of information, Appl. Intell. 42 (4) (2015) 738–750.
[23] B. Kaya, M. Poyraz, Age-series based link prediction in evolving disease networks, Comput. Biol. Med. 63 (2015) 1–10.
[24] B. Kaya, M. Poyraz, Supervised link prediction in symptom networks with evolving case, Measurement 56 (2014) 231–238.
[25] B. Kaya, M. Poyraz, Unsupervised link prediction in evolving abnormal medical parameter networks, Int. J. Mach. Learn. Cybern. 7 (1) (2016) 145–155.
[26] 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.
[27] D.D. Lee, HS. Seung, Algorithms for non-negative matrix factorization, in: Proceeding of the 13th Advances in Neural Information Processing Systems, 2001, pp. 556–562.
[28] J. Li, L.L. Zhang, F. Meng, F.H. Li, Recommendation algorithm based on link prediction and domain knowledge in retail transactions, Procedia Comput. Sci. 31 (2014) 875–881.
[29] 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.
[30] H. Liao, A. Zeng, Y.C. Zhang, Predicting missing links via correlation between nodes, Physica A. 436(2015) 216–223.
[31] W.P. Liu, L. Lü, Link prediction based on local random walk, Euro. Phys. Lett. 89 (5) (2010) 58007.
[32] L. Lü, C.H. Jin, T. Zhou, Similarity index based on local paths for link prediction of complex networks, Phys. Rev. E Stat. Nonlin. Soft Matter Phys. 80 (4) (2009) 046122.
[33] L.Y. Lü, T. Zhou, Link prediction in complex networks: a survey, Physica A 390 (2011) 1150–1170.
[34] X. Ma, J.L. Liao, S.M. Djouadi, Q. Cao, LIPS: link prediction as a service for data aggregation applications, Ad Hoc Netw. 19 (2014) 43–58.
[35] L.J. Michał, S. Jaroszewicz, Ł. Ostrowski, A. Wierzbicki, Verifying social network models of Wikipedia knowledge community, Inf. Sci. 339 (2016) 158–174.
[36] A. Nigam, N.V. Chawla, Link Prediction in a Semi-bipartite network for recommendation, Lect. Notes Comput. Sci. 9622 (2016) 127–135.
[37] A. Papadimitriou, P. Symeonidis, Y. Manolopoulos, Fast and accurate link prediction in social networking systems, J. Syst. Softw. 85 (9) (2012) 2119–2132.
[38] M. Pujari, R. Kanawati, in: Supervised Rank Aggregation Approach for Link Prediction in Complex Networks, 2012, pp. 1189–1196.
[39] K. Saito, M. Kimura, K. Ohara, H. Motoda, Super mediator, A new centrality measure of node importance for information diffusion over social network, Inf. Sci. 329 (2016) 985–1000.
[40] A.D. Sarma, A.R. Molla, G. Pandurangan, Distributed computation in dynamic networks via random walks, Theor. Comput. Sci. 58 (1) (2015) 45–66.
[41] E. Sherkat, M. Rahgozar, M. Asadpour, Structural link prediction based on ant colony approach in social networks, Physica A 419 (1) (2015) 80–94.
[42] V. Stroele, G. Zimbrao, J.M. Souza, Group and link analysis of multi-relational scientific social networks, J. Syst. Softw. 86 (2013) 1819–1830.
[43] Y. Sun, R. Barbery, M. Gupta, C.C. Aggarwal, J. Han, Co-author relationship prediction in heterogeneous bibliographic networks, in: Proceedings of 2011 International Conference on Advances in Social Networks Analysis and Mining, (ASONAM 2011), 2011, pp. 121–128.
[44] Z. Sun, Q. Peng, J. Lv, et al., A prediction model of post subjects based on information lifecycle in forum, Inf. Sci. 337-338 (2016) 59–71.
[45] A. Vidmer, A. Zeng, M. Medo, Y.C. Zhang, Prediction in complex systems: the case of the international trade network, Physica A 436 (2015) 188–199.
[46] D.Q. Vu, A.U. Asuncion, D.R. Hunter, P. Smyth, Continuous-time regression models for longitudinal networks, in: Advances in Neural Information Processing Systems 24: Proceedings of the 25th Annual Conference on Neural Information Processing Systems, 2011, pp. 1–9.
[47] P. Wang, B.W. Xu, Y.R. Wu, X.Y. Zhou, Link prediction in social networks: the state-of-the-art, Sci. China Inf. Sci. 58 (1) (2015) 1–38.
[48] X.J. Wang, X. Zhang, C.L. Zhao, et al., Predicting link directions using local directed path, Physica A 419 (2015) 260–267.
[49] X.M. Wang, Y. Liu, F. Xiong, Improved personalized recommendation based on a similarity network, Physica A 456 (15) (2016) 271–280.
[50] F. Xie, Z. Chen, J.X. Shang, X.P. Feng, J. Li, A link prediction approach for item recommendation with complex number, Knowl. Based Syst. 81 (2015) 148–158.
[51] Yang Yang, Nitesh Chawla, Yizhou Sun, Jiawei Han, Predicting links in multi-relational and heterogeneous networks, 2012, in: IEEE 12th International Conference on Data Mining, 2012, pp. 756–764.
[52] Z.Z. Zeng, Ke-Jia Chen, Shaobo Zhang, Haijin Zhang, A link prediction approach using semi-supervised learning in dynamic networks, in: Proceedings of Sixth International Conference on Advanced Computational Intelligence (ICACI), 2013, pp. 276–280.
[53] L. Zhang, P.N. Suganthan, A survey of randomized algorithms for training neural networks, Inf. Sci. 364 (2016) 146–155.
[54] http://www.linkprediction.org/index.php/link/resource/data.
[55] http://symatlas.gnf.org.
[56] http://www.cgd.ucar.edu/cas/catalog/climind/.