Linking Prediction Algorithm Integrating with Lower and Higher Order Features

Xiaoxin Li [1,2], Zhonglin Ye [1,2], Haixing Zhao [1,2*] and Yanlin Yang [1,2]

[1] *Qinghai Normal University, Wusi West Road, Xining, 810008, China*
[2] *The State Key Laboratory of Tibetan Intelligent Information Processing and Application, Xining, 810008, Qinghai, China*

Abstract

In view of the fact that the existing link prediction methods only focus on the local or global structure of the network, a link prediction algorithm (LP-ILHA) integrating low order and high order features is proposed. Firstly, the representation vector of the network is obtained by matrix decomposition of DeepWalk algorithm, and the weight between nodes with connected edges is obtained by calculating the similarity value between nodes, and the weight value is taken as the local feature; Secondly, the connection probability between nodes is predicted by Katz algorithm to obtain the connection weight between connected nodes in the network, which is taken as the global feature. Finally, the local and global characteristics of the network are combined to calculate the connection probability between nodes. The experimental results show that LP-ILHA algorithm has significantly improved prediction performance compared with traditional algorithms, and its performance is more excellent in link prediction of actual networks.

Keywords

Link prediction, Weighted network, DeepWalk, Similarity matrix, Random walk

1. Introduction

With the development of network technology, the evolution and topology of complex networks have become a research hotspot. Most entities in the real world can be represented by networks, such as circuit networks, biological networks, social networks, mobile communication networks, etc. Link prediction is an important tool for analyzing network change patterns. In essence, it is to predict the possibility of link generation at future moments between pairs of nodes that have not yet generated links, using information about node properties, connected edge relationships between nodes, and network topology under existing network conditions[1]. Research on link prediction has contributed to the development of various fields. For example, link prediction replaces expensive experiments in protein network prediction by link prediction techniques[2,3], in social network analysis, link prediction makes scientific recommendations based on user personality[4-6], and in recommender systems, link prediction can generate targeted personalized recommendations based on different users' preferences[7,8]. With the development of information technology, the requirements for link prediction accuracy are getting higher and higher, and how to improve the accuracy of link prediction has become a key issue for research.

Traditional link prediction methods can be classified into three categories according to the network structure: similarity metrics based on local information, similarity metrics based on global information and similarity metrics based on quasi-local information[9]. The link prediction algorithm based on local information can be used for larger networks, but the prediction is not effective. Conversely, link prediction algorithms based on global information have higher computational accuracy, while not

ICBASE2022@3rd International Conference on Big Data & Artificial Intelligence & Software Engineering, October 21-23, 2022, Guangzhou, China
EMAIL: lxx19980425@163.com (Xiaoxin Li); zhonglin_ye@foxmail.com (Zhonglin Ye); h.x.zhao@163.com (Haixing zhao) yanlin_yang@foxmail.com (Yanlin Yang)
ORCID: 0000-0003-1060-4484 (Xiaoxin Li);

applicable to the case of larger network size. Paralocal metrics, on the other hand, find a balance between prediction accuracy and time complexity in it. The literature [10] argued at there is an impact of the network structure on the accuracy of link prediction algorithms, such as the density, node degree, and aggregation coefficient of the network. Also the dense and sparse nature of the network can have very different properties. The literature [11] combined network representation learning with network structure features and proposed Ego-Embedding link prediction algorithm to improve the accuracy of link prediction algorithm by combining contextual reconstruction embedding in network representation learning. The literature [12] compared the existing similarity metrics and algorithms to calculate the local feature similarity. And proposed to calculate the structural similarity of neighboring nodes between target nodes based on the structural similarity metric of neighboring nodes, so as to determine the probability of link generation between target nodes. Ye et al[13] first combined the DeepWalk algorithm with link prediction, which is based on matrix decomposition to obtain a vector representation of the nodes in the network and use the cosine similarity method to derive the similarity matrix. Yang et al[14] considered the perspective of the higher order structure of the network and combined the contextual representation vector with the network node representation vector to derive a higher order similarity matrix representation, thus improving the accuracy of the link prediction algorithm. All of the above literature shows that the network structure has an impact on the accuracy of link prediction.

There are still more problems with link prediction. First, most of the existing link prediction algorithms only consider the local characteristics of the network or only consider the global characteristics of the network and neglect to consider the local and global information of the network together. Second, traditional link prediction algorithms have difficulty in solving the problem of insufficient training set due to network sparsity. Finally, the extant link prediction algorithms take less account of the edge weights. Therefore, to solve the above problem, we combine the global structure information of the network with the local structure information of the network. And we propose a link prediction algorithm that fuses low-order and high-order features (Linking Prediction Algorithm Integrating with Lower and Higher Order Features, LP-ILHA). The algorithm fuses the local and global features of the network with each other to improve the link prediction performance. The network structure features are constructed by the network adjacency matrix, and the vector of each node is obtained by the SVD matrix decomposition, and the similarity value between any two points is obtained by multiplying the obtained vectors two by two, which contains the local features of the nodes. And then the Katz algorithm is used to obtain the full path similarity value between nodes, which contains the global characteristics of the nodes. Finally, the weights between any connected two points are obtained by fusing local features with global features, and a weighted random wandering link prediction algorithm is proposed. The algorithm was evaluated using the AUC rubric and compared with more than a dozen algorithms. The results obtained are better than the original calculation results, proving the feasibility and effectiveness of the algorithm.

Our major contributions are as follows:

1. We integrate both global and local features into the link prediction algorithm by considering the local features of the network with the global features. This results in more feature factors in the calculated similarity values.
2. We solve the problem of under-trained data due to sparse network structure based on the matrix decomposition form of DeepWalk algorithm. The method is more efficient and can better understand the relevance of the network structure.
3. We design a weighted random wandering algorithm for evaluating the prediction probability between nodes. None of the current mainstream link prediction algorithms consider the information of connected edge weights, and the connected edge weights considered in this paper come from the network itself, which is very effective for improving the performance of the link prediction algorithm.

2. Relevant knowledge

2.1. Similarity metrics based on local information

(1) The common neighbor metric[1] considers that if there are more common neighbor nodes between node x and node, the nodes are considered to be more similar to each other and more likely to generate

links in the future. Where $\Gamma(x)$ and $\Gamma(y)$ represent the set of neighbors of node x and node y, respectively, as shown in Equation (1),

$$S_{xy}^{CN} = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}. \tag{1}$$

(2) Similarity metrics based on Admic-Adar algorithm[1]. The AA metric considers that neighboring nodes with small degrees are more likely to have contiguous edges with other nodes, where k represents the degree of the node, As shown in equation (2),

$$S_{xy}^{AA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log k_z}. \tag{2}$$

(3) Similarity metrics based on resource allocation algorithms[1]. The RA metric assumes that resources will be distributed equally among nodes, so that nodes with more access to resources have a higher probability of producing contiguous edges, where k_z denotes the degree of node z, as shown in Equation (3),

$$S_{xy}^{RA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{k_z}. \tag{3}$$

(4) Similarity metric based on CN metrics of local plain Bayes[1] assumes that the possibility of generating connected edges between nodes depends on the set of common neighbors, where $C(z)$ represents the node aggregation coefficient and ρ represents the network density, as shown in Equation (4),

$$S_{xy}^{LNBCN} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \log(\frac{C(z)}{1-C(z)}) + \log(\frac{1-\rho}{\rho}). \tag{4}$$

(5) Similarity metrics of local plain Bayes based on AA metrics are similar to LNBCN metrics[15]. quantified the influence of neighboring nodes and combined the proposed role function with AA metrics, as shown in Equation (5),

$$S_{xy}^{LNBAA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log k_z} (\log(\frac{C(z)}{1-C(z)}) + \log(\frac{1-\rho}{\rho})). \tag{5}$$

(6) Similarity metrics based on RA metrics[15] with local plain Bayes are consistent with AA metrics except for different weight assignments, as shown in Equation (6),

$$S_{xy}^{LNBRA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{k_z} (\log(\frac{C(z)}{1-C(z)}) + \log(\frac{1-\rho}{\rho})). \tag{6}$$

(7) The preferred connection similarity metric[1] only considers the size of the node degree and does not need to calculate information about the neighborhood of each node, so the complexity of the calculation is small, where the probability of a new link connecting to node x is positively related to k_x, as shown in Equation (7),

$$S_{xy}^{PA} = k_x k_y. \tag{7}$$

(8) Based on the favorable similarity of nodes with large degree, it is believed that nodes with larger degree have more connected edges and therefore have higher probability of linking with other nodes, as shown in Equation (8),

$$S_{xy}^{HPI} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\min\{k_x, k_y\}}. \tag{8}$$

(9) In contrast to HPI based on unfavorable similarity of nodes with large degree, HDI considers nodes with small degree to have higher probability of linking with other nodes, as shown in Equation (9),

$$S_{xy}^{HDI} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\max\{k_x, k_y\}}. \tag{9}$$

(10) The node-pair assignment based similarity index[1] (Leicht Holme Newman Local Index, LHN-l) is similar to the CN index, but the LHN-l index considers that two nodes have more neighboring nodes with larger similarity scores, as shown in Equation (10),

$$S_{xy}^{LHN-1} = \frac{|\Gamma(x) \cap \Gamma(y)|}{k_x \times k_y}. \tag{10}$$

2.2. Similarity index based on global information

(1) The Katz metric[1] considers all interconnected paths between two nodes, where β represents the adjustable parameter controlling the path weights, $|paths_{x,y}^{i}|$ represents the number of paths of length between connecting node *x* and node *y*, and *I* represents the unit matrix, as shown in Equation (11),

$$S_{xy}^{katz} = \beta A_{xy} + \beta A_{xy}^2 + \ldots\ldots = (I - \beta A)^{-1} - I. \tag{11}$$

(2) Average Commute Time (ACT)[1], $m(x,y)$ represents the average number of steps to be taken to wander randomly from node *x* to node *y*, as shown in Equation (12),

$$n(x,y) = m(x,y) + m(y,x). \tag{12}$$

This value can be obtained by the pseudo-inverse of the Laplace matrix, and the formula $n(x,y) = M(l_{xx}^{+} + l_{yy}^{+} - 2l_{xy}^{+})$ can be obtained by the pseudo-inverse L^{+}(L=D-A) of the Laplace matrix, where l_{xx}^{+} denotes the element corresponding to the *x*-th row and *y*-th column position in matrix L^{+}, and the shorter the average commuting actual distance between two nodes indicates that the two nodes are closer, as shown in equation (13),

$$S_{xy}^{ACT} = \frac{1}{l_{xx}^{+} + l_{yy}^{+} - 2l_{xy}^{+}}. \tag{13}$$

(3) The SimRank metric[1] proposes that two nodes are more likely to be similar if their neighboring nodes are similar, and *C* represents the decay function of similarity transfer, taking values in the range [0,1], as shown in Equation (14),

$$S_{xy}^{SimR} = C \frac{\sum_{v_z \in \Gamma(x)} \sum_{v_z' \in \Gamma(y)} s_{zz'}^{SimR}}{k_x k_y}. \tag{14}$$

(4) Similarity metrics based on matrix forest theory take into account the multi-path weighting problem between nodes, where *L* represents the Laplace matrix and *I* is the unit matrix, as shown in Equation (15),

$$S_{xy}^{MFI} = (I + \partial L)^{-1}. \tag{15}$$

2.3. Similarity index based on quasi local information

(1) The local path-based similarity metric[1] considers the influence of second-order neighbors and third-order neighbors based on the common neighbor similarity metric and adds adjustable parameters, as shown in Equation (16),

$$S_{xy}^{LP} = A_{xy}^2 + \alpha \cdot A_{xy}^3, \tag{16}$$

where A represents the adjacency matrix of the network, A_{xy}^{n} represents the number of paths with path length n between node x and node y, α represents the adjustable parameters, and so on, which can be extended to *n*-th order paths, but the complexity of the local path metrics will be increasing as the order gradually increases.

(2) The transfer-based similarity metric[17] conveys similarity with intermediate nodes by improving the collaborative filtering algorithm, where ε represents the constant, *T* is the similarity matrix, and *S* is the similarity matrix passed, as shown in Equation (17),

$$S_{xy}^{TSCN} = \varepsilon TS + T. \tag{17}$$

3. Methodology framework

3.1. Problem description

Given an undirected network $G=(V,E)$, containing n nodes with m edges, where V represents the set of nodes in the network, which can be denoted as $V = \{v_1, v_2, v_3, v_4, \cdots, v_n\}$, and E represents the set of connected edges between nodes in the network, which can be denoted as $E = \{e_1, e_2, e_3, \cdots, e_m\}$. A_{n*n} represents the adjacency of the n-node network in the network, and W represents the weight matrix in the network. To facilitate the experiment, the data are divided into a training set G_{train} and the test set G_{test} , where G_{train} is the set of networks composed of known information.

3.2. Calculation of connected edge weights for networks based on low-order features

Word2Vec is a word embedding model proposed by Google in 2013, which obtains word vector representation by neural network algorithm, mainly consisting of Skip-Gram model (Continuous Skip-Gram) and CBOW model (Continuous Bag-of-Words Model), and the word vector obtained by this method has low dimensionality and high density. The DeepWalk algorithm is based on the idea of Word2vec algorithm, which uses the co-occurrence relationship between nodes to learn the vector representation of nodes, and uses the random walk strategy to obtain the vertex sequence and introduce it into the Skip-Gram model to learn the embedding of vertices. obtain the local data to be trained. Second, the Skip-Gram algorithm is used to learn the sampled data and generate a vectorized representation of the nodes. Meanwhile, the Skip-Gram model can map the network nodes into a low-dimensional vector space, thus reducing the complexity, and the node sequence of the network is obtained by random walk, which can be combined with the Skip-Gram model to effectively improve the accuracy of the link prediction model [18], and the Skip-Gram model is shown in Figure 1.

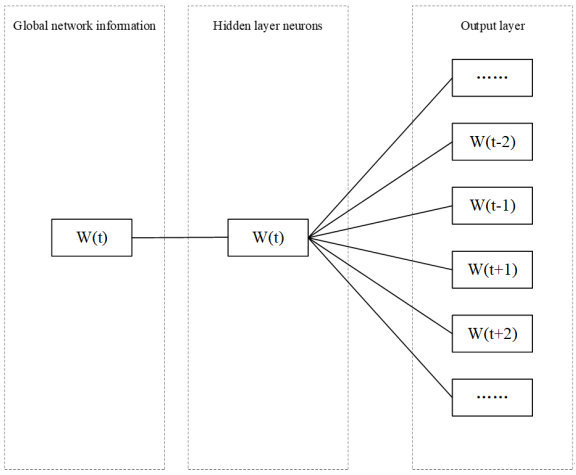

Figure 1: Skip-Gram model

The Skip-Gram model based on negative sampling is called SGNS (Skip-Gram with Negative Sampling), which is essentially a three-layer neural network in the network trained on contextual node relationships, and the objective function of SGNS is,

$$L(S) = \frac{1}{|S|}\sum_{i=1}^{|S|}\sum_{-t\le j\le t, j\ne 0} \log \Pr(v_{j+i} \mid v_i), \tag{18}$$

$$\Pr(v_j \mid v_i) = \frac{\exp(v_j' v_i)}{\sum_{v \in V} \exp(v' v_i)}, \tag{19}$$

where v_i is the network representation vector of the current node and v_j is the network representation vector of the context node. v represents the sum of the representation vectors of the context nodes.

The existing research results show that the SGNS algorithm is essentially a matrix decomposition, The specific formula is as follows.

$$M_{i,j} = \log \frac{N(v_i, c_j) \times |D|}{N(v_i) \times N(c_j)} - \log n, \tag{20}$$

where n represents the number of context negative sampling nodes, $|D|$ is the number of nodes in the whole training set, $N(v)$ represents the number of occurrences of nodes in the whole training set D, and $N(v_i,c_j)$ represents the number of occurrences of context nodes, current nodes in the whole training set at the same time.

Yang et.al[19] subjected the above equations to optimization and simplification operations and finally defined the calculation procedure of the network structure characteristic matrix as follows.

$$M = \frac{A + A^2 + A^3 + \ldots\ldots + A^n}{n}, \tag{21}$$

where A is the adjacency matrix, and Yang et al. have derived the simplified network structure feature matrix M by considering the complexity of the algorithm and the accuracy of the algorithm as follows.

$$M = \frac{A + A^2}{2}. \tag{22}$$

The reduced network structure feature matrix considers only the computational relationships between the low-order nodes, which greatly reduces the computational time complexity at the expense of a smaller degree of precision.

SVD (Singular Value Decomposition) is an information extraction method[20], which can simplify data, remove noise, remove redundant data, and reduce computation by eliminating the numbers with small singular values that may be noise. The topology of a network and the connected edge relationship are known, and the network can be represented by means of an adjacency matrix, through which the target matrix M of the network matrix decomposition can be obtained. In this paper, the target matrix M is decomposed into three multiplicative matrices of the form U, S, and V by the SVD method to reduce the data dimensionality and to obtain the important features in the data.

Based on the text information TADW (Text Associated DeepWalk) algorithm uses the induced matrix complementation algorithm to complement the network structure feature matrix and introduces the network text attribute information into the network representation learning. The literature [21] proposes the TELP algorithm based on the TADW algorithm and uses the cosine similarity algorithm to calculate the similarity between the nodes and reduce the complexity of matrix decomposition. The representation vector of each node is obtained by decomposing the similarity matrix, so that the connected edge weights between any two nodes i and j in the network are defined as,

$$sim(i, j) = \frac{Emb(j,:).\times Emb(i,:)}{\|Emb(j,:)\|_2 \times \|Emb(i,:)\|_2}. \tag{23}$$

In summary, this paper uses equation (23) as the low-order weight for the existence of connected edges between nodes.

3.3. Link prediction algorithm combining low order and high order features

In this paper, we propose link prediction algorithms that fuse low-order and high-order features, which incorporate local and global information of the network. First, based on the fact that the essence of DeepWalk algorithm is the matrix M decomposition of the structural feature matrix of the network, which generates the representation vector of the nodes, the representation vector is used to obtain the weight information of any connected node in the network, which contains the local features between

the connected edges. Second, the Katz algorithm is introduced to calculate the weights of connected edges based on the global features, which contain the global features between connected edges. Finally, a weighted random wandering algorithm is proposed based on the weight information, and the above processes are fused to obtain a link prediction algorithm that fuses low-order and high-order features.

The random walk algorithm[22] is to randomly select a node in the network to walk and perform a walk traversal of the network, and most of the existing link prediction algorithms do not consider the information of the connected edge weights of the network. Therefore, this paper proposes a weighted local random walk (WLRW) algorithm with less complexity than the global random walk strategy. Compared with the ACT algorithm and the SimR algorithm, the WLRW algorithm takes into account the network weights and is more consistent with the actual situation of the network. p_x is the node resource distribution and W is the weight matrix, where $\Pi x(0)$ is an N*1 vector, and the specific algorithm is shown in Equation (24),

$$s_{xy}^{WLRW}(t)=p_x\cdot W\cdot\pi_{xy}(t-1)+p_y\cdot W\cdot\pi_{yx}(t-1). \tag{24}$$

In this paper, by fusing the obtained low-order features of the network nodes with the high-order features of the network nodes, so that there are and only unique features among the nodes.

In order to make the reader understand the details of the algorithm in this paper more clearly, the specific algorithm framework is shown in Figure 2.

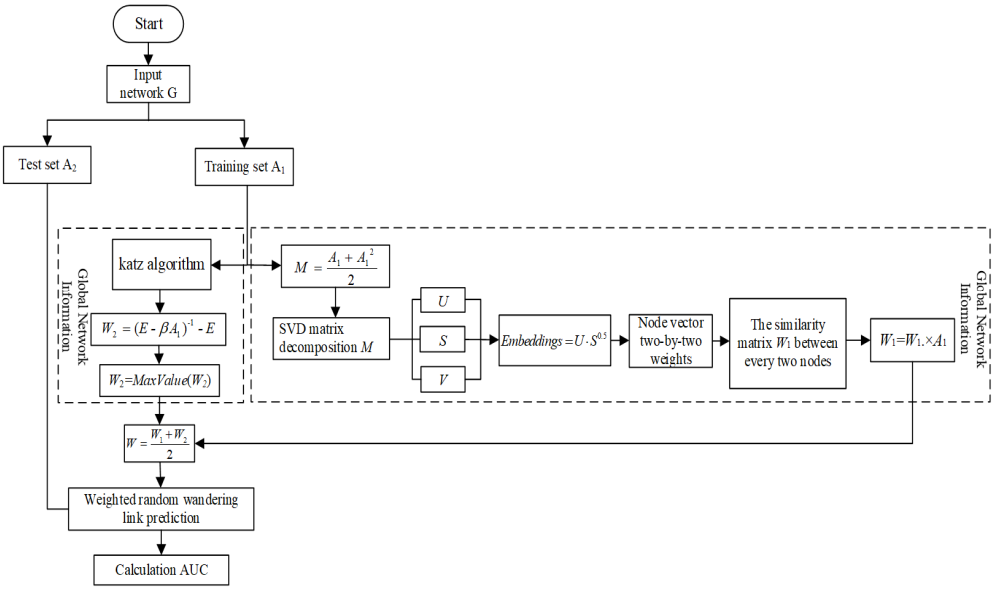

Figure 2: Weighted random wandering algorithm framework for link prediction

In this paper, a link prediction algorithm that fuses low-order and high-order features is proposed. First, the network G is divided into training set and test set, and the adjacency matrix A_1 of the training set and the adjacency matrix A_2 of the test set are derived. second, the matrix decomposition form of DeepWalk algorithm is used to obtain the network structure feature matrix M, which contains the first-order and second-order features of nodes, the feature matrix containing local network information. matrix decomposition is performed on the target matrix M to obtain the representation vector Emb of nodes The weight matrix W_1 containing local network information is obtained by calculating the similarity values between the vectors.The Katz algorithm is used to obtain the weight matrix W_2 containing global network information, the feature matrix containing global network information.The weight matrix W_1 containing low-order features is weighted and fused with the weight matrix W_2 containing high-order features to obtain the weight matrix W in which the local and global information are fused.Finally, the weight matrix W with fused low-order and high-order features is applied to the weighted random wandering prediction algorithm and evaluated with the AUC evaluation criterion.

The pseudo code of link prediction algorithm combining low order and high order features is as follows:

Input：Adjacency Matrix Representation A of Network G；
Output：AUC；

1 $M = \frac{A_1 + A_1^2}{2}$；
2 Decomposition matrix M into U, S and V matrices through SVD；
3 $Emb = U \times S^{0.5}$；
4 Create a local weight zero matrix W_1 of size $n*n$， $w_1(i,j) \in W_1$；
5 **for** $i \leftarrow 1$ **to** n :
  for $j \leftarrow 1$ **to** n:
    $$W_1(i,j) = \frac{Emb(j:) \cdot Emb(i:)}{\| Emb(j:) \|_2 \times \| Emb(i:) \|_2};$$
  end
end
6 $W_1 = W_1 . \times A_1$；
7 $W_2 = ((E - \beta A_1)^{-1} - E) . \times A_1$；
8 $W = \frac{W_1 + W_2}{2}$；
9 $s_{xy}^{LP-ILHA}(t) = q_x \cdot W \cdot \pi_{xy}(t-1) + q_y \cdot W \cdot \pi_{yx}(t-1)$；
10 Calculate AUC；

The framework can be divided into the following six steps, which are performed as follows.

1) Dividing the network dataset into two parts, the training set and the test set, and deriving the corresponding adjacency matrices A_1 and A_2, respectively, which in turn lead to the structural feature matrix M containing the first-order neighboring features and second-order neighboring features of the nodes through Equation (22).

2) Decompose the structural feature matrix $M_{|v|*|v|}$ into three matrices $U_{|v|*k}$, S_{k*k}, and $V_{k*|v|}$ by SVD matrix decomposition, and multiply the matrix $U_{|v|*k}$ with one-half power of the matrix S_{k*k} to obtain the node vector matrix Emb with v rows and k columns, and calculate the weight matrix W_1 between any two nodes using equation (20), $||\text{-}||_F$ represents F parametric, where the matrix W_1 contains the low-order features of the network, and the weight matrix of the first-order neighbor and second-order neighbor features with the presence of connected edges is obtained by W_1 dot product of the adjacency matrix A_1.

3) Calculate the adjacency matrix A_1 by Katz algorithm to obtain $W_2 = (E - \beta A_1)^{-1} - E$, the weight matrix W_2 containing global features.

4) the weight matrix W_1 containing the local information of the network and the weight matrix W_2 containing the global information of the network are weighted and fused to obtain the final weight matrix $W = (W_1 + W_2) ./ 2$; and

5) Using W as the weight transfer matrix of the WLRW algorithm.

6) The effectiveness of this method is verified by evaluating it with the AUC evaluation criteria.

4. Experimental results and analysis

To avoid the uncertainty of single kind of network experiments, this experiment is conducted using four real citation networks and some datasets from two social networks, and the reliability of the link prediction algorithm proposed in this paper by fusing low-order and high-order features is demonstrated by training the six datasets at training rates of 0.7, 0.8, and 0.9, respectively.

4.1. Data set

The experiments in this paper use four real citation network datasets[23], Citeseer, DBLP, Cora, Wiki network with some datasets from two social networks, Facebook network and NS Scientist

Collaborative Network, Table 1 lists the attributes of the datasets used in this experiment, where $|V|$ stands for the number of nodes, $|E|$ stands for the number of edges, K stands for the average degree, D stands for the network diameter, L stands for the average path length, P stands for the density, and C stands for the average clustering coefficient.

Table 1
Dataset Properties

Data set	$\|V\|$	$\|E\|$	K	D	L	P	C
Citeseer	3312	4732	2.857	28	9.036	0.001	0.257
DBLP	3119	39516	25.339	14	4.199	0.008	0.259
Cora	2708	5429	4.010	19	6.310	0.001	0.293
Wiki	2405	17981	14.953	9	3.650	0.006	0.480
FaceBook	1631	2603	2.603	24	7.833	0.002	0.246
NS	1461	2468	3.379	19	6.349	0.002	0.775

4.2. Evaluation criterion

We use the AUC (Area under the receiver operating characteristic curve) metric as the metric of link prediction accuracy in this paper, and the AUC is calculated as shown in Equation (25). The principle is to give each edge in the network a weight, AUC assigns a weight to each unknown edge, AUC evaluates the index that two edges are randomly selected, one of which is an edge that already exists in the network, i.e., an edge e_1 in the test set G_{test} . The other is an unknown edge that does not exist in the network at the moment, i.e., a random edge e_2 in the U-E.Each edge has its fractional value. If the fraction of e_1 is greater than the fraction of e_2, the prediction is accurate and 1 point is added. If the scores of both edges are equal, 0.5 point are added. When the comparison is repeated n times independently, the following formula is obtained assuming that n_1 times the fractional value of e_1 is greater than the fractional value of e_2 and n_2 times the fractional value of e_1 is equal to the fractional value of e_2.

$$AUC = \frac{n_1 + 0.5n_2}{n}. \tag{25}$$

In this paper, the LP-ILHA algorithm was evaluated by the AUC rubric, and the LP-ILHA algorithm was experimented on four datasets, CiteSeer, DBLP, Cora, and WiKi, with training rates of 0.7, 0.8, and 0.9 and compared the LP-ILHA algorithm with 17 other algorithms[12,13], as shown in Table 2.

Table 2
Citation network link prediction results(%)

Data set	CiteSeer			DBLP			Cora			Wiki		
Training rate	0.7	0.8	0.9	0.7	0.8	0.9	0.7	0.8	0.9	0.7	0.8	0.9
CN	68.1	72.0	74.6	85.4	88.4	90.6	69.5	72.3	78.1	86.4	88.8	90.6
Saltion	66.3	72.7	74.4	86.0	87.9	90.7	69.3	72.1	77.8	85.2	88.1	88.6
HPI	66.2	72.1	74.4	85.6	88.9	90.7	69.3	72.4	77.9	85.5	88.7	88.5
HDI	66.0	72.5	74.1	85.7	88.3	90.8	69.5	72.5	76.6	85.4	87.3	89.3
LNH-1	66.4	72.9	74.4	85.8	87.8	89.9	69.1	72.1	77.3	84.9	87.7	87.9
AA	66.3	72.2	74.3	86.0	88.2	90.9	69.3	72.6	77.6	87.2	88.1	91.4
RA	66.3	72.1	74.6	86.5	88.5	90.8	69.4	72.4	77.9	85.7	89.1	90.7
PA	78.9	79.0	79.5	76.3	77.1	77.5	71.5	71.9	71.5	82.8	82.2	82.6
LP	81.0	86.8	88.4	92.9	93.6	94.9	80.1	82.9	87.9	92.3	93.9	94.1
Katz	96.8	97.9	97.1	**93.4**	94.1	94.8	90.8	92.1	94.4	**93.6**	94.5	94.6
LNBAA	66.3	72.6	74.5	86.0	88.4	91.1	69.4	72.5	78.0	86.5	87.9	90.5
LNBCN	66.7	72.2	74.2	85.6	88.4	90.8	69.5	72.1	77.7	86.5	89.0	90.1
LNBRA	66.0	72.2	74.2	85.8	88.9	91.2	69.3	72.8	77.7	86.6	87.8	89.3

ACT	75.8	75.5	73.7	79.0	80.0	80.8	74.1	73.6	74.0	79.2	80.4	79.1
Cos+	88.5	89.3	88.4	91.5	93.4	95.0	90.2	90.9	93.2	91.4	91.3	92.4
MFI	96.6	**98.0**	97.8	95.1	**96.0**	**97.0**	93.1	94.2	**95.6**	93.3	**94.7**	94.6
TSCN	84.2	85.6	86.2	91.2	91.0	92.3	88.3	90.6	92.9	91.6	92.3	78.7
LP-ILHA	**97.4**	97.5	**97.9**	91.7	92.4	93.5	**93.4**	**94.6**	94.8	93.4	94.3	**95.0**

From the table, we can see that the LP-ILHA algorithm performs very well in most of the networks, and the AUC values obtained are above 91%. Its average AUC in the four citation networks is 94.6%. From the experiments, it is seen that the LP-ILHA algorithm has good results in all four real citation networks. And the LP-ILHA algorithm has a partial improvement compared with the other 17 algorithms. In the CiteSeer dataset, the LP-ILHA algorithm has the highest improvement of 31.4% compared with other algorithms, and the overall average improvement of 20% relative to other algorithms. In the DBLP dataset, LP-ILHA improves up to 16%, with an overall average improvement of 3.7% relative to the other algorithms. In the Cora dataset, LP-ILHA algorithm improves up to 24.3%, with an overall average improvement of 16% compared to other algorithms. In the Wiki dataset, the LP-ILHA algorithm improves up to 16.3%, compared to the overall average of 5.8% for the other algorithms. It is concluded from the experiments that the LP-ILHA algorithm can effectively improve the accuracy of link prediction algorithm by considering the local information and the global information of the network.

Table 3
Social network link prediction results(%)

Data set	FaceBook			NS		
Training rate	0.7	0.8	0.9	0.7	0.8	0.9
CN	65.3	71.8	79.5	91.4	96.0	96.7
Saltion	63.7	71.5	76.3	93.6	94.7	98.2
HPI	63.5	73.6	77.7	92.7	95.2	98.2
HDI	63.1	72.7	79.7	92.7	96.5	97.5
LNH-1	63.8	71.8	77.2	91.5	95.3	97.9
AA	64.2	71.3	76.6	91.2	96.2	98.4
RA	63.9	74.5	79.4	92.3	96.1	98.1
PA	93.6	94.2	95.6	73.5	73.3	72.7
LP	78.6	87.0	89.4	98.3	98.7	99.5
Katz	99.1	**99.4**	**99.5**	99.7	99.8	99.8
LNBAA	65.0	71.4	78.8	92.0	94.8	96.5
LNBCN	64.3	71.9	77.8	91.9	95.3	97.1
LNBRA	65.3	73.2	78.3	92.3	96.2	97.3
ACT	54.2	57.7	56.2	60.2	58.2	56.6
MFI	99.1	99.3	99.4	99.9	**99.8**	**99.9**
TSCN	97.1	98.6	98.9	**99.9**	99.1	99.3
LP-ILHA	**99.2**	99.1	99.2	99.2	99.6	99.6

From Table 3, it can be seen that the link prediction algorithm proposed in this paper fusing low-order and high-order features works extremely well in FaceBook and NS social networks. The LP-ILHA algorithm proposed in this paper has an average AUC of 99.3% in both social networks. From the experiments it is seen that the LP-ILHA algorithm has better results in both real social networks. And the LP-ILHA algorithm also has partial improvement compared with other 16 algorithms, which indicates that our proposed LP-ILHA algorithm is also applicable in social networks.

5. Summary

We propose a link prediction algorithm that fuses low-order and high-order features. The algorithm is divided into two parts, the first part is to obtain the low-order features of the network nodes and the

second part is to obtain the high-order features of the network nodes. The fusion of higher-order features with lower-order features is weighted so that there is one and only one feature between each node. At this point, all edges in the network are given a unique weight. We consider the weights between each node pair and propose a weighting-based random wandering link prediction algorithm, which leads to a great improvement in the overall link prediction performance. The algorithm is embedded into the link prediction algorithm proposed in this paper that fuses low-order and high-order, thus aiding and enhancing the accuracy of the link prediction algorithm in this paper. We conducted simulation experiments in four real citation networks, Citeseer, DBLP, Cora, and Wiki, along with two social networks, Facebook and NS. The experimental results show that the model in this paper has improved compared with most existing link prediction algorithms, which verifies the effectiveness of the algorithm in this paper.

6. Acknowledgements

This work was financially supported by the National Key R&D Program of China (2020YFC1523300) and the Qinghai Provincial Natural Science Foundation Youth Program (2021-ZJ-946Q).

7. References

[1] L. Y. LÜ, Link Prediction on Complex Networks, Journal of University of Electronic Science and Technology of China 39 (2010) 651–661.
[2] H. Y. Yu, P. Braun, M. Yildirim, et. Al, High-Quality Binary Protein Interaction Map of the Yeast Interactome Network, Science 322 (2008) 104–110.
[3] M. P. Stumpf, T. thorne, S. E. De, et al. Estimating the size of the human interactome, Proceedings of the National Academy of Sciences of the United States of America 105 (2008) 6959–6964.
[4] X. Q. XIE, Y. J. LI, Z. Q. ZHANG, et al. A joint link prediction method for social network, Intelligent Computation in Big Data Era 503 (2015) 56–64.
[5] J. L. SCHAFER, J. W. GRAHAM. Missing data: our view of the state of the art, Psychol Methods 7 (2002) 147–152.
[6] G. Kossinets. Effects of missing data in social networks, Social Networks 28 (2006) 247–268.
[7] T. ZHOU, L. Y. LÜ, Y. C. ZHANG, Predicting missing links via local information, European Physical Journal B 71 (2009) 623–630.
[8] J. LESKOVEC, D. HUTTENLOCHER, J. Kleinberg, Predicting positive and negative links in online social networks[C]. Proceedings of the 19th International Conference on World Wide Web. New York: ACM Press (2010) 641–650.
[9] K. AJAY, S S Singh, K S, B BHASKAR, Link prediction techniques, applications, and performance: A survey, Physica A: Statistical Mechanics and its Applications (2020) 553.
[10] S. N. Wu, H. J. Pu, R. N. Tian, W. Q. Liang, Q. Yu, Network Structure's Impacts on Link Prediction Algorithm from Meta-Analysis Perspective, Data Analysis and Knowledge Discovery 5 (2021) 102–113.
[11] M. Zhao, J. K. Zhao, J. N. Liu, Link Prediction Algorithm Based on Ego Networks Structure and Network Representation Learning, Computer Science 48 (2021) 211–217.
[12] Y. C. Yong, M. Xu, H. L. Fu, S. N. Sun, Local Structure Similarity Algorithm for Link Prediction, Journal of Chinese Computer Systems 39 (2018) 27–31.
[13] Z. L. Ye, R. Cao, H. X. Zhao, K. zhang, Y. Zhu, Link Prediction Based on Matrix Factorization for DeepWalk, Application Research of Computer 37 (2020) 424–429.
[14] Y. L. Yang, Z. L. Ye, H. X. Zhao, L. Meng, Link Prediction Algorithm Based on High-order Proximity Approximation, Journal of Computer Applications 39 (2019) 2366–2373.
[15] R. F. Wang, Z. Q. Chen, Z. X. Liu, Link prediction in complex networks with syncretic naive Bayes methods, CAAI Transactions on Intelligent Systems 14 (2019) 99–107
[16] P. CHEBOTAREV, E. SHAMIS, The matrix-forest theorem and measuring relations in small social groups, Automat & Remote Control 58 (1997) 1505–1514.

[17] D. SUN, T. ZHOU, J. G. LIU, et al, Information filtering based on transferring similarity, Physical Review E 80 (2009) 17101–17101.
[18] C. Zhao, F. X. Zhu, S. C. Liu, A Link Prediction Method Based on SkipGram Model, Computer Applications and Software 34 (2017) 241–247.
[19] C. Yang, Z. Liu, Comprehend Deepwalk as Matrix Factorization, Computer Science, 2015.
[20] B. Fu, X. X. Zhang, X. X. Fan, X. L. Zhao, L. He, Stability Analysis of Second Order Discrete Time-Varying Linear System Based on SVD Decomposition, Computer Applications and Software 37 (2020) 37–45.
[21] R. Cao, H. X. Zhao, Z. L. Ye, Link Prediction Algorithm Based on Text Enhanced of Network Nodes, Computer Applications and Software 36 (2019) 227–235.
[22] H. Y. Zhao, J. Zhang, J. Cao. Personalized App Recommendation Algorithm Based on Topic Grouping and Random Walk, Application Research of Computers 35 (2018) 2277–2280.
[23] L. Y. LÜ, T Zhou. Link prediction, 2013, China, 2013.