IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 5, 2020, accepted June 17, 2020, date of publication June 19, 2020, date of current version June 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003822

Multi-Resolution Prediction Model Based on Community Relevance for Missing Links Prediction

JINGYI DING, (Member, IEEE), JIAN SONG, LICHENG JIAO, (Fellow, IEEE), JIANSHE WU, (Senior Member, IEEE), AND FANG LIU, (Senior Member, IEEE)

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an 710071, China

Corresponding author: Jingyi Ding (jyding87@163.com)

This work was supported in part by the China Postdoctoral Science Foundation Funded Project under Grant 2018M643586, and in part by the National Natural Science Foundation of China under Grant 61902294.

ABSTRACT The existing research demonstrates that the link prediction algorithm which based on community similarity has better prediction performance than that of other node similarity-based methods, and it is more suitable for predicting the probability of the missing links between node-pairs with far distance. However, the disadvantage of these community similarity-based methods is the resolution of prediction accuracy is very low, which resulting in the existence probability of the missing links between node-pairs within a community or between a specific pair of communities is the same. In addition, the link prediction algorithms which based on multi-resolution community division can calculate the existence probability of missing links under different resolutions, but the relevance between communities had not taken into account, which makes it difficult to predict the existence probability of target links if the number of interconnections between communities is small. Combining the advantages of these two algorithms, we propose a more realistic link prediction model which based on a novel quasi-local community relevance index under multi-resolution community division. The performance of our algorithms is demonstrated by comparing with other well-known methods on two kinds of networks in different scales. The experiment results indicate that our approaches are very competitive.

INDEX TERMS Link prediction, complex networks, community relevance, multi-resolution community division.

I. INTRODUCTION

Now with the increasing applications of big data and large-scale network communication in our life, the influence of internet on people's life is greater than before. Users are eager to get effective recommendations in their life, so more attention has been put on the improvement of link prediction algorithms which as the supporting technology [1]–[3] in recommendation systems. Designing a fast and accurate prediction algorithm is extremely necessary. Nowadays, some link prediction algorithms have been proposed. Those methods were classified into five categories such as similarity-based methods, maximum likelihood-based methods, graph representation-based methods, graph neural network-based methods and community-based algorithms.

The similarity-based methods are the most commonly used methods for link prediction. In despite of its simplicity, the study on similarity-based algorithms is the mainstream issue. In fact, the definition of node similarity is a non-trivial challenge. Similarity index can be very simple or very complicated and it may work well for some networks while fails for some others. Similarity-based algorithms are inspired by node similarity and structural similarity. The idea of node similarity is quite straightforward: two nodes are similar if they share many common features [4]. However, the attributes of nodes are generally hidden. Therefor we focus on another group of similarity indices which named structural similarity indices. The structural similarity indices based solely on the network structure. They can be divided

The associate editor coordinating the review of this manuscript and approving it for publication was Alberto Cano.

into two categories: local similarity indices [5]–[8] and global similarity indices [9]–[11]. The link prediction algorithm based on local similarity with low time complexity, but its prediction accuracy is not good enough. On the contrary, the link prediction algorithm based on global similarity is of high prediction accuracy, but its time complexity is high too.

Maximum likelihood-based methods often try to describe the network structure with some likelihood model. This kind of algorithms is able to describe the hierarchical organization structure and community structure of the networks [12]–[14]. However, the runtime of this kind of algorithms increase exponentially along with the number of vertices increases.

Graph representation-based methods pay more attention to the characteristics of the network. Most of the traditional machine learning methods try to classify the non-existence edges according to the labels of the observed edges by extracting the structural features of the network [15], [16]. In order to improve the computational efficiency of the graph algorithm, some scholars have proposed the graph embedding techniques that represent a graph as a low dimensional vector, and they are successfully applied in the link prediction problem [17]–[21]. However, this kind of graph embedding methods also has its own drawbacks, because the accumulation of error functions in the model will result in the phenomenon of gradient explosion and gradient disappearance, moreover, the performance of this kind of methods is sensitive to the initial node selection.

Because that the graph neural network model can preserve the structural characteristics of the network very well, some link prediction algorithms based on graph neural network model have been proposed recently [22]–[24]. However, although this kind of methods considers some neighborhood information of the network, the topological structure of the graph is not fully utilized. In addition, the networks with too few connections cannot be predicted.

Most of the link prediction algorithms mentioned before try to predict the probability of missing links from the micro perspective of the network. There is also a kind of algorithms try to predict the link probability from the middle perspective of the network, that is the community structure-based link prediction algorithms [25]–[27]. This kind of algorithms assumes that links in the same community nodes are more similar and have higher existing probability than links between different community nodes, moreover, they are improved by taking into account the association between communities. However, the performance of this kind of methods is limited by the resolution of community division, and such algorithms are failed when the community structure of the network is not good enough.

In this paper, the main objects of our study are undirected unweighted networks. In view of the defects of the predictive algorithms based on community structure, we attempt to propose a more realistic link prediction model based on a novel quasi-local community relevance index under multi-resolution community division. In addition, a quantitative analysis is made to explain the performance ranking of the proposed algorithms.

The contributions of this work are:

1) Combined the link prediction model which based on community relevance with the link prediction model which based on multi-resolution community division to overcome the low-resolution prediction of the traditional link prediction model, and the new model is more realistic.

2) A novel quasi-local community relevance metric is proposed to improve the prediction accuracy of the link prediction algorithms.

3) The proposed algorithms can be used not only to distinguish the probability of the missing links between node-pairs within a community or between a specific pair of communities, but also to predict the probability of the missing links between node-pairs with far distance.

The rest of the paper is organized as follows:

Section 2 introduces the related work. Section 3 describes the problem formulation. Section 4 introduces the novel link prediction approach and explains the new community relevance indices. The experiments and analyses are provided in Section 5. Finally, Section 6 concludes the paper with some discussions and promising future work.

II. RELATED WORK

It is difficult to avoid the errors or redundant edges in networks when we reconstruct the complex systems in the real world. Some missing links in a static network need us to find out [28], in the meantime, some potential connections which will appear in the future also need us to predict in a dynamic network [29], [30]. This is the purpose of the network link prediction [31], [32].

Since Newman proposed the similarity index Common Neighbors (CN) for missing links prediction in 2001 [5], lots of well-known indices based on structural similarity appear. such as Preferential Attachment (PA) [5], Adamic–Adar (AA) [7] and Resource Allocation (RA) [8]. All of them are local similarity indices with low time complexity. Duo to their low prediction accuracy, Leicht proposed another similarity index which named Leicht–Holme–Newman Index (LHN2) [10] in 2006. This is a global similarity index which has an obvious improvement in prediction accuracy. But its time complexity is high too. Later, some global similarity indices are proposed, such as Katz Index [9] and Local Path Index (LP) [8], [11]. In 2008, Stochastic Block Model (SBM) [13], [14] is proposed by Airoldi. It is a maximum likelihood-based method and is able to describe the community structure of networks, but the whole process of SBM is very time consuming and it is impossible to sum over all partitions even in a small network. Then, some scholars have proposed the graph embedding techniques that represent a graph as a low dimensional vector, and they are successfully applied in the link prediction. For example, Node2vec methods which proposed by Grover in 2016 implicitly preserve higher order proximity between nodes by generating multiple random walks which connect nodes at various distances due

to its stochastic nature [18]. SDNE algorithm which proposed by Wang in 2016 utilize the ability of deep auto-encoder which is modeling non-linear structure in the data to generate an embedding model that can capture non-linearity in graphs [20]. Because that the graph neural network model can preserve the structural characteristics of the network very well, some link prediction algorithms based on graph neural network model have been proposed recently. Zhang et. proposed an algorithm (SEAL) in 2018 [23]. By extracting a local subgraph around each target link, they aim to learn a function mapping the subgraph patterns to link existence, thus automatically learning a ''heuristic'' that suits the current network. Pan et. proposed a novel adversarial graph embedding framework (ARVGA) for graph data in 2019 [24]. They have mostly considered the data distribution of the latent codes from the graphs, which often results in inferior embedding in real-world graph data.

Since Yan proved that the community structure of the network is significant for link prediction in their paper which published in PLE in 2012 [25], a new kind of link prediction algorithms based on network community structure has appeared. For example, an algorithm based on the multi-resolution community structure information of the network was proposed by Ding in 2014 [26]. This method fully considers the multi-resolution community structure information in setting the prediction parameters, but it is not suitable for predicting the existence probability of target links if the number of interconnections between communities is small because it doesn't consider the relevance of the communities. Then, a link prediction method based on the community relevance is proposed in 2016 [27]. Compared with other community structure-based prediction methods, this method is highly efficient because it only considers the local information of the network in computing the community relevance matrix. However, the advantages of prediction accuracy are not obvious. In addition, the performance of this method is limited by the low resolutions, because in the proposed methods node-pairs within a community or between a specific pair of communities will achieve the same score. Later, Shashank proposed a community-based link prediction method by considering the utility and effect of information diffusion in network in 2020 [33]. It is proved that community structure information combined with information dissemination can improve the prediction performance. However, this algorithm only considers the influence probabilities among the users when dividing the community structure, it doesn't consider the relevance among communities too.

Motivation: The traditional community structure-based algorithms which take into account the relevance between different communities are suitable for predicting the existence probability of target links if the number of interconnections between communities is small. However, the prediction resolution of this community similarity-based methods is very low, which result in the existence probability of the missing links between node-pairs within a community or between a specific pair of communities is the same. In order to retain the predictability of the methods for the target links with far distance and overcome the disadvantage of low-resolution of the traditional community relevance-based methods, we combine the multi-resolution community division model and the traditional community relevance-based methods to propose a more realistic link prediction model which based on a novel quasi-local community relevance index under multi-resolution community division.

It takes into account the multi-resolution community structure of the network and uses the relationship between different communities to predict the missing connections. Our method is not only applicable to predict the existence probability of target links if the number of interconnections between communities is small, but also applicable to accurate predict the existence probability of the missing links between node-pairs within a community or between a specific pair of communities.

III. PROBLEM FORMULATION

Considering a network $G = G(V, E)$, where V is the set of nodes and E is the set of undirected links. For simplicity, we suppose that the network is unweighted, and multiple links and self-connections are not allowed. Assuming that $|V|$ denotes the number of nodes, $|E|$ denotes the number of links, $A = (a_{ij})_{N\times N}$ denotes the adjacency matrix of the network.

$$A=(a_{ij})_{N\times N}=\begin{cases}1, & \textit{There is a link betwee n nodes i and j}\\ 0, & \textit{There is no link between nodes i and j}\end{cases} \tag{1}$$

The task of the link prediction algorithm is to predict the existence probability of nonexistent link between node x and node y, $x, y \in V$. The existence probability P_{xy} can be represented by score S_{xy}. Given that the existence probability of missing links is proportional to the community relevance. So, we can construct a prediction model by using this proportional relationship.

$$P_{xy} = S_{xy} \tag{2}$$

$$S_{xy_{x\in c_i, y\in c_j}} \propto CR_\lambda(c_i, c_j) \tag{3}$$

where, $CR_\lambda(c_i, c_j)$, $(i, j = 1, 2, \ldots, m)$ is a symmetric matrix which represents the similarity between different communities. The value of the parameter λ determines the resolution of the community division.

We adopt AUC, a standard metric, to quantify the performance of different link prediction methods [34]. It is related to the sensitivity (true positive rate) and specificity (true negative rate) of a classifier [35]. The meaning of the formula below is that among n independent comparisons, the probability that a randomly chosen missing link is given a higher score than a randomly chosen nonexistent link. Here, n' is the number of the occurrences that the missing link having a higher score and n'' is the number of the occurrences that the missing link and nonexistent link having the same score. Thus, the value of degree exceeds 1/2 indicates how much

better the algorithm performs than pure chance.

$$AUC = \frac{n' + 0.5n''}{n} \tag{4}$$

In addition, we also use precision to quantify the performance of different link prediction methods [32]. Given the ranking of the non-observed links, the precision is defined as the ratio of relevant items selected to the number of items selected. The precision is defined as follows:

$$P = \frac{L_r}{L} \tag{5}$$

where L is the number of links as the predicted ones, L_r is the number of links which are predicted right.

IV. ALGORITHM

After the problem formulation, then we explain our algorithms in this section. There are three steps in this method. Firstly, extract the multi-resolution community structure of the network. Secondly, calculate the relevance of each pair of communities under different resolutions and get the characteristics of community structure. Finally, a simple prediction model is applied to estimate the probability of the missing links.

A. MULTI-RESOLUTION COMMUNITY DIVISION

As one of the most important characteristics of community structure, community relevance is used in link prediction problem [27]. However, the performance of this method is limited by the low resolutions, because in the method node-pairs within a community or between a specific pair of communities will achieve the same score. In order to solve the problem of low-resolution of this traditional community relevance-based methods, we combine the multi-resolution community division model and calculate the relevance of communities under different resolutions to predict the probability of missing links.

The first step of our algorithm is extracting the community structure of the network by using the community division method. Here, we use the method based on the optimization of modularity density to detect communities of the network at different resolutions.

The modularity density D_λ is defined as follows [30]:

$$D_\lambda = \sum_{i=1}^{m} \frac{2\lambda L(V_i, V_i) - 2(1-\lambda)L(V_i, \overline{V_i})}{|V_i|}, \quad 0 \le \lambda \le 1 \tag{6}$$

where $L(V_i, V_i)$ is the number of the links inside the community c_i, $L(V_i, \overline{V_i})$ is the number of the links between the nodes in the community c_i and the nodes out of the community c_i, $|V_i|$ is the number of the nodes in the community c_i, the value of the parameter λ determines the resolution of the community division.

The pseudo code of the community division algorithm (MR-CD) is as follows:

Input: Network $G = G(V, E)$, λ.
Output: the community division of different resolutions $C = \{\{c_1, c_2, \ldots, c_{m_1}\}_{\lambda_1}, \{c_1, c_2, \ldots, c_{m_2}\}_{\lambda_2}, \ldots, \{c_1, c_2, \ldots, c_{m_k}\}_{\lambda_k}\}$

For $\lambda = 0.1 : 0.1 : 1$
Step 1: Initialize the communities. // $N = |V|$
$C_0 = \{v_1, v_2, \ldots, v_N\}$ // divide the communities by each node.
For $i = 1 : N$
For each $j \in \Gamma(i)$ // $\Gamma(i)$ means the neighbors of the node i.
If $\Delta D_\lambda(j \to c_i) \ge 0$
$j \in c_i$// the gain of modularity density is positive, then node j is placed in the community c_i
End
End
End
Step 2: Rebuild the network
$v_i \leftarrow c_i$// The nodes of the new network are the communities found from the first step.
Self-links $= L(v_i, v_i)$ // The number of the links between nodes in the same community
Between-links $= L(v_i, \overline{v_i})$ // The number of the links between nodes in the corresponding two communities.
Continue step 1
End

B. COMMUNITY RELEVANCE INDEX

After getting the community division results of the different resolutions, we then need to calculate the similarity between two different communities and get the symmetric community relevance matrix $CR_\lambda(c_i, c_j), (i, j = 1, 2, \ldots, m)$. In this paper, we extend local community relevance (CR) index to quasi-local community relevance index which contained more information of the network, both local and global information, to improve the performance of the traditional CR algorithm [27]. And propose three new indices which named CR-Katz, CR-LHN2 and CR-LP.

The detailed definitions are as follows:

Definition 1 (CR-Katz): Follows the basic ideal that two communities more relevant if they have more paths with different length between them. This index is based on the ensemble of paths, which directly sums over the collection of paths and is exponentially damped by length to give the shorter paths more weights.

$$CR(c_i, c_j)^{Katz} = \sum_{l=1}^{n} \beta^l \cdot \left| paths(c_i, c_j)^{<l>} \right| \tag{7}$$

$$\left| paths(c_i, c_j)^{<1>} \right| = \left| (\Gamma(c_i) \cup V(c_i)) \cap V(c_j) \right| \tag{8}$$

$$\left| paths(c_i, c_j)^{<2>} \right| = \left| (\Gamma(c_i) \cup V(c_i)) \cap (\Gamma(c_j) \cup V(c_j)) \right| \tag{9}$$

$$CR(c_i, c_j)^{Katz} = \sum_{x \in c_i, y \in c_j} \beta A_{xy} + \beta^2 A^2_{xy} + \ldots + \beta^n A^n_{xy} \tag{10}$$

where, $paths(c_i, c_j)^{<l>}$ is the set of all paths of length l connecting the community c_i and the community c_j, and β is a free parameter controlling the path weights. $A = [a_{xy}]$ is the adjacency matrix of the communities. Note that, for the previous community similarity-based algorithms, if community x and community y are connected in the network, then $a_{xy} = 1$, otherwise $a_{xy} = 0$. This setting ignores the influence of connection density within and between communities on the predicted results. Our proposed method is different from the previous community similarity algorithm. Here A is a matrix which the diagonal elements are not 0 but equal to the number of links in the same community. For example, A_{11} is the number of links in the community c_1, and A_{12} is the number of links between the communities c_1 and c_2.

β must be lower than the reciprocal of the largest eigenvalue of matrix A to ensure the convergence of Eq. 9. So, $\beta < 1/\lambda_{\max}$, and $\lambda_{\max}$ is the maximum eigenvalue of A [32]. The value of parameter n indicates the range of network structure information used by the index.

Definition 2 (CR-LHN2): This index is a variant of the CR-Katz index. Based on the concept that two communities are similar if their immediate neighbors are themselves similar, one obtains a self-consistent matrix formulation.

$$CR(c_i, c_j)^{LHN2} = \psi \sum_{x \in c_i, y \in c_j} I + \beta A_{xy} + \beta^2 A_{xy}^2 \ldots + \beta^n A_{xy}^n \tag{11}$$

where ψ and β are free parameters controlling the balance between the two components of the similarity. The matrix A is the same as definition 1 and the value of parameter n indicates the range of network structure information used by the index.

Definition 3 (CR-LP): To provide a good tradeoff of accuracy and computational complexity, we here introduce an index that takes consideration of local paths.

$$CR(c_i, c_j)^{LP} = \sum_{x \in c_i, y \in c_j} A_{xy}^2 + \ldots + \beta A_{xy}^3 \tag{12}$$

where, β is a free parameter like the ones above.

C. PREDICTION MODEL

As know that there is of great relationship between the strength of community relevance and the existence probability of missing links. So, we can construct a prediction model by using this relationship.

$$S_{xy} = Predict_{x \in c_i, y \in c_j}(\sum_{\lambda=1}^{k} \alpha_\lambda \cdot CR_\lambda(c_i, c_j)) \tag{13}$$

$$\text{Where, } \sum_{\lambda=1}^{k} \alpha_\lambda = 1 \tag{14}$$

Here, α_λ is a parameter that represents the importance of $CR_\lambda(c_i, c_j)$ to S_{xy} under different λ. It is known that the higher the resolution of the community division, the greater the influence of this community relevance on the prediction results [26]. Therefore, with the increase of λ, the resolution of community partition becomes higher, and the corresponding weight α is larger.

The pseudo code of the link prediction algorithm (MLCD-LP) is as follows:

Input: the community division of different resolutions $C = \{\{c_1, c_2, \ldots, c_{m_1}\}_{\lambda_1}, \{c_1, c_2, \ldots, c_{m_2}\}_{\lambda_2}, \ldots, \{c_1, c_2, \ldots, c_{m_k}\}_{\lambda_k}\}$
Output: An array of existence probability $[i, j, S_{v_i v_j}]$ of the missing links in the network G

Step 1: Calculate the community relevance matrices
For each $\{c_1, c_2, \ldots, c_{m_i}\}_{\lambda_i}, i = 1 : k$
$LSCR(x, y)_{\lambda_i} = CR(c_x, c_y)$; // The relevance between different communities is equal to the value of community relevance indices.
$LSCR(x, x)_{\lambda_i} = 1$; // The self-relevance of each community is 1.
end
Step 2: Predict the existence probability of the missing links
If $x \in c_x, y \in c_y$
$S_{xy} = \sum_{i=1}^{k} \alpha_{\lambda_i} \cdot LSCR(x, y)_{\lambda_i}, \sum_{i=1}^{k} \alpha_{\lambda_i} = 1$; // The existence probability of the missing links is proportional to the relevance of the communities in which the nodes are located.
End

V. EXPERIMENTS

In this section, we evaluate our proposed algorithm in terms of validity, parameter analysis and time complexity. In addition, we also compared with other similarity-based methods, maximum likelihood-based methods, graph representation-based methods, graph neural network-based methods and community-based algorithms to verify the predictive accuracy of our algorithm by experiment on benchmark networks and real-world networks. The compared algorithms are Katz, LHN2, LP, Stochastic Block Model (SBM), Structural Perturbation Method (SPM) [36], node2vec, SEAL, ARVGA, Yan's algorithm, CLP-ID and the algorithms which based on community relevance. The benchmark networks we used in this section is Lancichinetti-Fortunato-Radicchl (LFR) benchmark dataset [37], [38], which is 100, 1000 and 10000 in size, respectively. The real networks for testing are electrical power-grid network of the western US (power), Ssc network, protein-protein interaction network (protein), co-authorship network of scientists (netscience), Amazon e-commerce network (amazon) and PGP network [39], which listed in TABLE 1. Where N is the number of nodes, E is the number of edges, Q is the modularity of the network, $< k >$ is the average degree of the nodes, C is the clustering coefficient, d is the average path length of the network [39], and $\lambda_{\max}$ is the maximum eigenvalue of the matrix A. $d = \inf$ indicate the network is a disconnected network.

TABLE 1. Parameters of the networks.

Network	N	E	Q	$< k >$	C	d	$\lambda_{\max}$
ssc	961	1192	0.695	4.550	0.132	inf	5.562
netscience	1589	5484	0.956	3.754	0.638	inf	9.344
protein	2112	4406	0.845	2.387	0.060	inf	4.914
amazon	2879	7772	0.669	2.700	0.074	3.433	20.15
power	4941	13188	0.934	2.669	0.080	18.99	4.355
PGP	10680	548632	0.852	4.554	0.266	7.486	21.78

A. PARAMETERS ANALYSIS

In the fourth chapter, the requirement of the index CR-Katz is the parameter $\beta < {}^{1}/_{\lambda_{\max}}$ to ensure the convergence of Eq. 10, here, $\lambda_{\max}$ is the maximum eigenvalue of the matrix A. However, it is very time consuming to get the characteristic value of the matrix in large network, so we hope to fix the value of the parameter β. As show in table 1, we set the value of β as $\beta = 0.001$ in the following experiments.

Our methods considered the path information of the network, the smaller the average path of the network is, the better the network connectivity is. Because that the farther path away from the prediction link, the less influence on similarity index. So in this paper we value the parameter $n = \lceil d \rceil$, where d represents the average path length of the network, it means that we set $\beta = 0$ when n is greater than the average path length of the network. We can find that most of the average path length of the connected networks range from 2 to 3 [27], so when the network is unconnected, which means $d = \text{inf}$, we set $n = 4$.

For each network we use the 10-fold cross validation method to create the training sets and testing sets. And then we attempt to predict missing links based on the remaining connections.

B. COMPARED WITH OTHER STATE-OF-ART METHODS

In this part, our proposed methods are mainly compared with Katz, LHN2, LP, SBM, SPM, Node2vec, SEAL and ARVGA algorithms. Firstly, the definitions of indices Katz, LHN2 and LP are given here.

Katz Index: This index is based on the ensemble of all paths, which directly sums over the collection of paths and is exponentially damped by length.

$$S_{xy}^{Katz} = \beta A_{xy} + \beta^2 A_{xy}^2 + \beta^3 A_{xy}^3 + \ldots \tag{15}$$

where β is a free parameter, and A is the adjacency matrix of the network.

LHN2 Index: This index is a variant of the Katz index. Based on the concept that two nodes are similar if their immediate neighbors are themselves similar.

$$S_{xy}^{LHN2} = \psi(I + \phi A_{xy} + \phi^2 A_{xy}^2 + \ldots) \tag{16}$$

where ψ and ϕ are free parameters controlling the balance of the similarity.

LP Index: To provide a good tradeoff of accuracy and computational complexity.

$$S_{xy}^{LP} = A_{xy}^2 + \ldots + \varepsilon A_{xy}^3 \tag{17}$$

where ε is a free parameter, and this measure degenerates to CN when $\varepsilon = 0$.

Then, the prediction results on the benchmark networks which the number of nodes is 100, 1000 and 10000 are shown in FIGURE 1. Here the parameter μ is 0.10, 0.20, 0.30, 0.40 and 0.50 respectively.

The figure 1 shows that the prediction accuracy of our algorithms decreases with the increase of the parameter μ. It's because that the parameter μ determines the quality of community structure of the network. However, the performance of our proposed algorithms is based on the results of community division. The community division will be inaccurate if the community structure of the network is not good. So, the performance of our algorithm will change for the worse with the weakening of the network community structure. Moreover, from the sub-figure (a) and (c), we find that our proposed methods perform better than other comparison algorithms except the graph neural network-based algorithms. It is because that the algorithms SEAL and ARVGA also considered the neighborhood structure of the networks in the process of training duo to their special structures. However, they also have their disadvantages, that is the performance is not good for the networks which are sparse. This can be proved from the results in the Appendix A of the supplementary material, for networks of the same size, the smaller the average degree of the nodes is, the worse the prediction performance of the algorithms SEAL and ARVGA is. On the contrary, the performance of our algorithms is good in the same case. It is because that our algorithms try to predict the link probability from the middle perspective of the network, and is taken into account the association between communities, so it is more suitable for predicting the existence probability of target links if the number of interconnections between communities is small. In addition, we can find that the variance of our proposed algorithms is smaller than that of the comparison algorithms from the results in the Appendix A.

From the sub-figure (b), (d) and (f), we find that the Precision of our proposed methods is larger than other comparison algorithms. Here, the Precision is calculated by using the formula (5), where the denominator L is the top 5% of the probability ranking of the non-observed links. Even the denominator is only 5% of the total predicted edges, it is still a very large value compared with the numerator L_r, so, the value of precision is very small.

Then, we attempt to verify the performance of these algorithms in the real-world networks. FIGURE 2 shows the histogram of mean and variance of prediction accuracy in different networks. X-axis represents the different algorithms. Y-axis represents the AUC & Variance and Precision & Variance, respectively. This section only shows the prediction accuracy on the Amazon and Power networks, and the

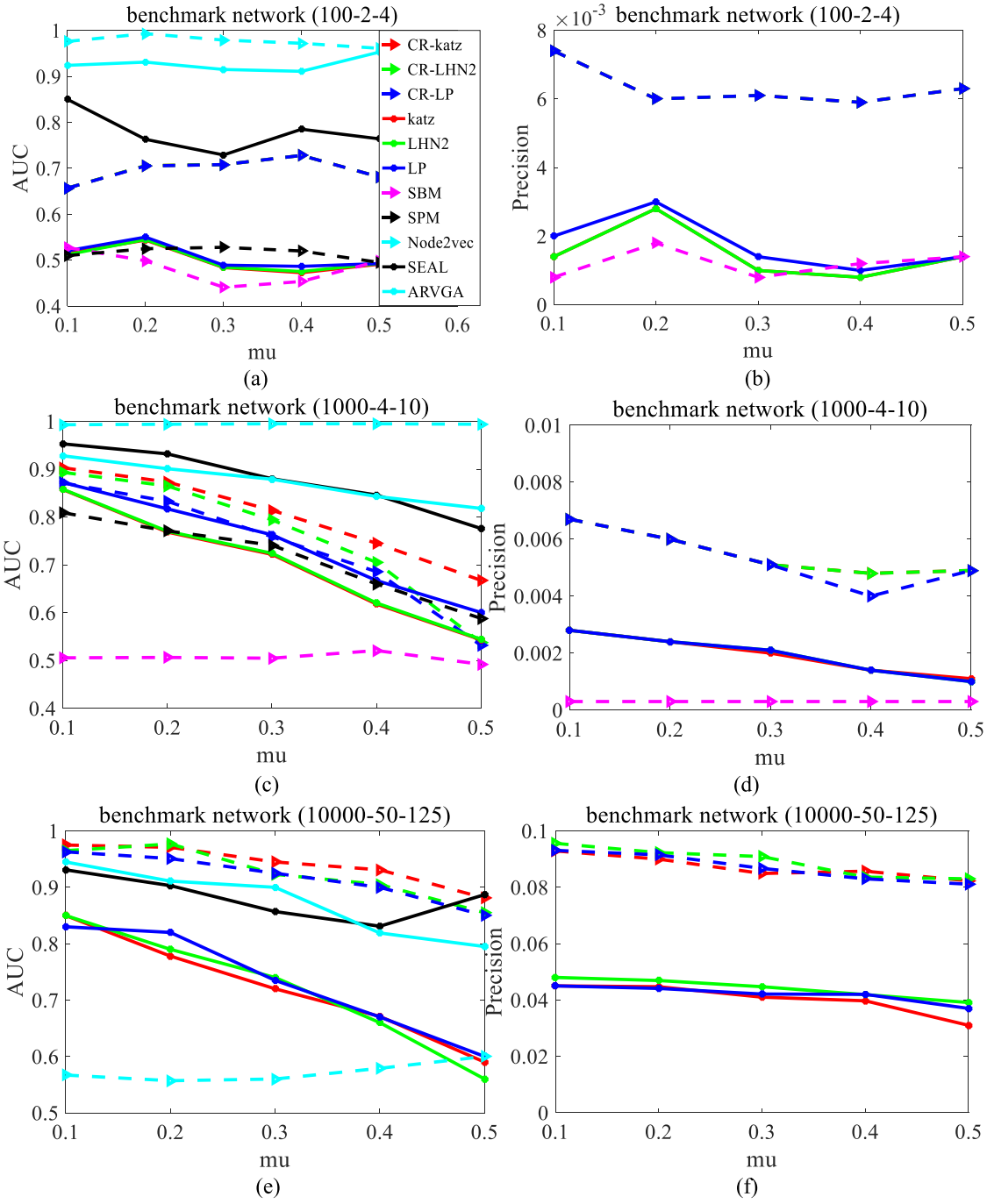

FIGURE 1. The performance comparison of different algorithms on benchmark networks. The lines with different colors indicate different algorithms. The red dashed line represents the CR-Katz algorithm; The green dashed line represents the CR-LHN2 algorithm; The blue dashed line represents the CR-LP algorithm; The red solid line indicates the Katz algorithm; The green solid line represents the LHN2 algorithm; The blue solid line represents the LP algorithm; The rose red dashed line represents the SBM algorithm; The black dashed line indicates the SPM algorithm; The cyan dashed line indicates the Node2vec algorithm; The black solid line indicates the SEAL algorithm; The cyan solid line indicates the ARVGA algorithm. (a), (b) show the results of benchmark networks. Here, $N = 100$, $< k >= 2$, $k_{\max} = 4$. (c), (d) $N = 1000$, $< k >= 4$, $k_{\max} = 10$. (e), (f) $N = 10000$, $< k >= 50$, $k_{\max} = 125$. Where N represents the number of nodes, $< k >$ represents the average degree of the nodes, k_{max} represents the maximum degree and μ is the mixing parameter. In each subfigure the X axis represents the parameter μ, and Y axis represents AUC and Precision, respectively.

results of other networks can be found in the Appendix B of the supplementary material.

From the FIGURE 2 we can see that the variance of our proposed algorithms is smaller than that of the other comparison algorithms. In addition, the prediction performance of our proposed algorithms is better than that of other comparison algorithms except the algorithms which based on graph neural network. This is the same as previously speculated in benchmark networks.

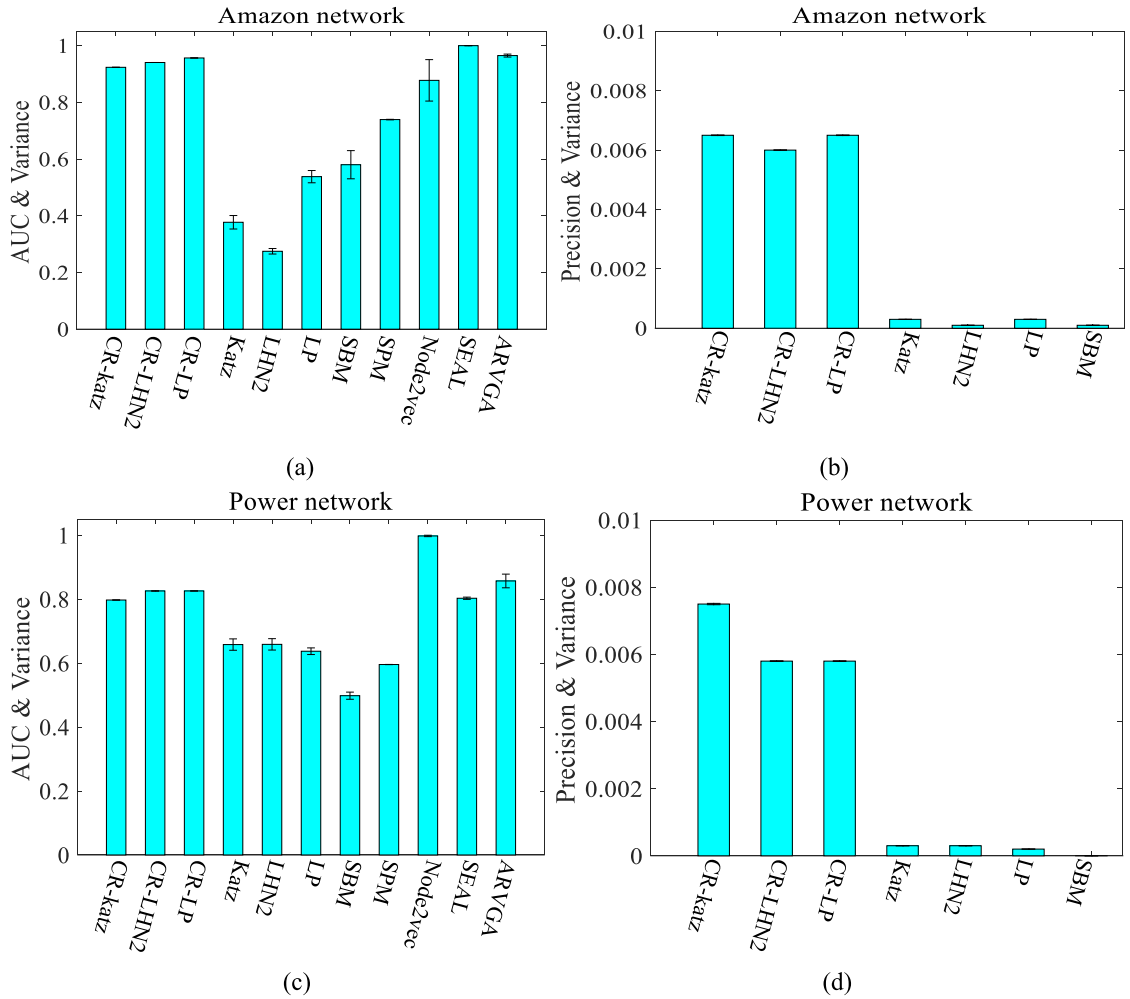

FIGURE 2. The performance comparison of different algorithms on real-world networks. (a), (b) Show the results of Amazon e-commerce network. Here, $N = 2879$ and $E = 7772$. (c), (d)Show the results of electrical power-grid of the western US. Here, $N = 4941$ and $E = 13188$. In each subfigure the X axis represents different algorithms, and Y axis represents AUC & Variance and Precision & Variance. The compared algorithms are CR-Katz, CR-LHN2, CR-LP, Katz, LHN2, LP, SBM, SPM, node2vec, SEAL and ARVGA respectively.

In general, according to the results which obtained from FIGURE 1, FIGURE 2, Appendix A and Appendix B, it is found that the prediction accuracy values of our proposed algorithms CR-Katz, CR-LHN2 and CR-LP are very close. In addition, the prediction performance of our proposed algorithms, including the AUC, Precision and the variance, is better than that of other comparison algorithms except the algorithms which based on graph neural network. For sparse network, our proposed algorithms have more advantages.

Then, we compare the performance of our algorithms with other link prediction methods which based on community structure of the network.

C. COMPARED WITH OTHER COMMUNITY STRUCTURE-BASED ALGORITHMS

In this part, our proposed algorithms are mainly compared with CP, CR-JC, CAR, Yan's algorithm and CID-IP algorithms. Firstly, the definitions of indices CR-JC and CAR are given here.

CR-JC Index: It measures the probability that both c_i and c_j share a common neighbors.

$$CR(c_i, c_j)^{CR-JC} = \frac{\left|(\Gamma(c_i) \cup V(c_i)) \cap (\Gamma(c_j) \cup V(c_j))\right|}{\left|(\Gamma(c_i) \cup V(c_i)) \cup (\Gamma(c_j) \cup V(c_j))\right|} \quad (18)$$

where $\Gamma(c_i)$ denotes the set of neighbors of community c_i, and $V(c_i)$ indicates the set of nodes in community c_i.

CAR Index: It suggests that two nodes are more likely to link together if their common-first-neighbors are members of a strongly inner-linked cohort.

$$S_{xy}^{CAR} = S_{xy}^{CN} \cdot \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{|\gamma(z)|}{2} \quad (19)$$

where $\Gamma(x)$ denotes the set of neighbors of node x, $\gamma(z)$ refers to the sub-set of neighbors of z that are also common neighbors of x and y.

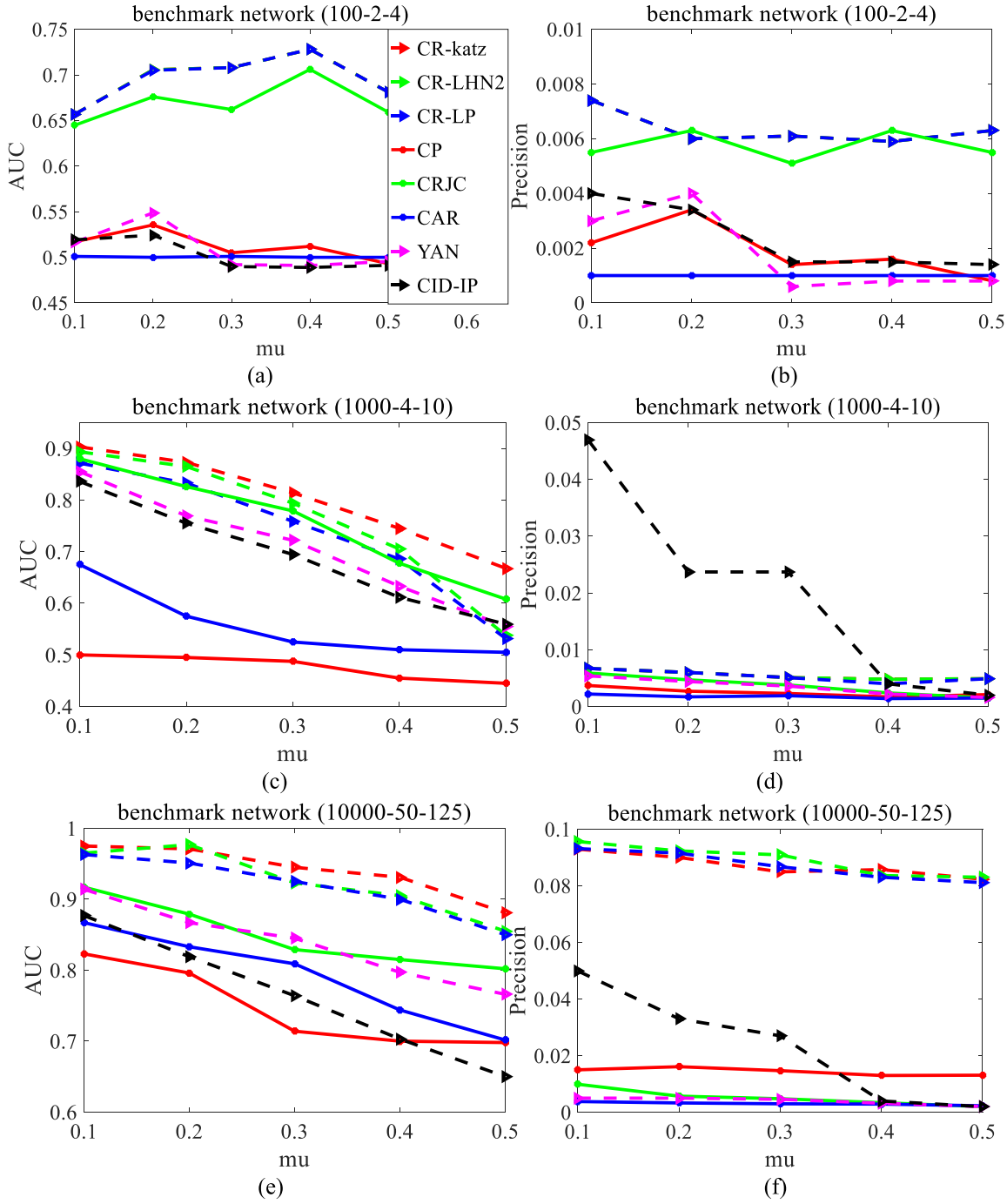

FIGURE 3. The performance comparison of different algorithms on benchmark networks. The lines with different colors indicate different algorithms. The red dashed line represents the CR-Katz algorithm; The green dashed line represents the CR-LHN2 algorithm; The blue dashed line represents the CR-LP algorithm; The red solid line indicates the CP algorithm; The green solid line represents the CRJC algorithm; The blue solid line represents the CAR algorithm; The rose red dashed line represents the YAN'S algorithm; The black dashed line indicates the CID-IP algorithm; (a), (b) show the results of benchmark networks. Here, $N = 100$, $< k >= 2$, $k_{max} = 4$. (c), (d) $N = 1000$, $< k >= 4$, $k_{max} = 10$. (e), (f) $N = 10000$, $< k >= 50$, $k_{max} = 125$. Where N represents the number of nodes, $< k >$ represents the average degree of the nodes, k_{max} represents the maximum degree and μ is the mixing parameter. In each subfigure the X axis represents the parameter μ, and Y axis represents AUC and Precision, respectively.

Then, the prediction results on the benchmark networks and the real-world networks are shown in FINGURE 3 and FIGURE 4, respectively. The experimental settings are similar to those above.

It can be seen from FIGURE 3 that when the number of nodes is 100, the prediction results of algorithms CR-Katz, CR-LHN2 and CR-LP are very close, so the AUC and Precision curves almost coincide. In addition, our algorithms have

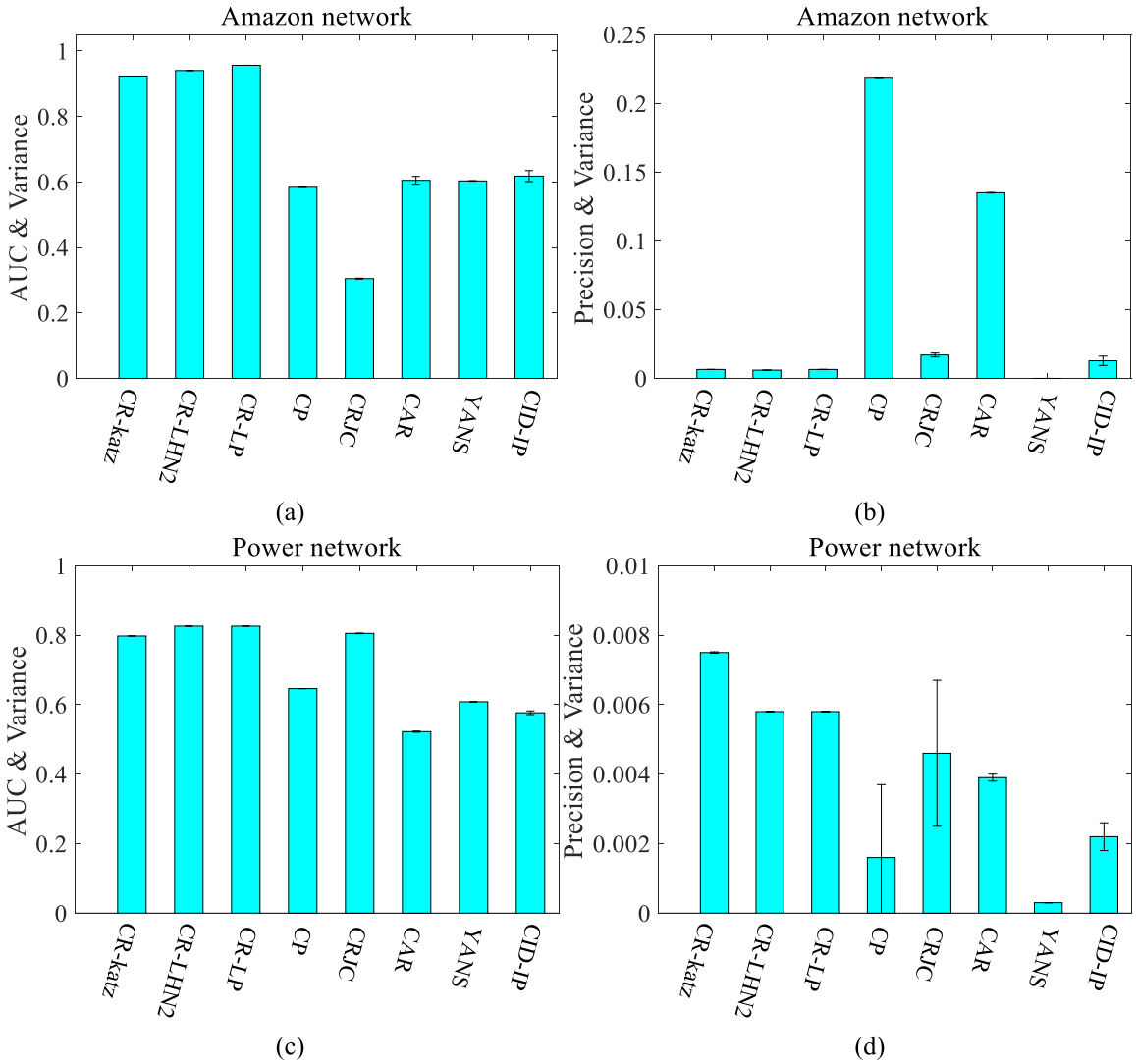

FIGURE 4. The performance comparison of different algorithms on real-world networks. (a), (b) Show the results of Amazon e-commerce network. Here, $N = 2879$ and $E = 7772$. (c), (d)Show the results of electrical power-grid of the western US. Here, $N = 4941$ and $E = 13188$. In each subfigure the X axis represents different algorithms, and Y axis represents AUC & Variance and Precision & Variance. The compared algorithms are CR-Katz, CR-LHN2, CR-LP, CP, CRJC, CAR, Yan's algorithm and CID-IP algorithms, respectively.

advantages in prediction accuracy and variance by compared with other algorithms. This conclusion can be proved by the results in Appendix C of the supplementary material.

Then, we attempt to verify the performance of these algorithms in the real-world networks. FIGURE 4 shows the histogram of mean and variance of prediction accuracy in different networks. X-axis represents the different algorithms. Y-axis represents the AUC & Variance and Precision & Variance, respectively. This section only shows the prediction accuracy on the Amazon and Power networks, and the results of other networks can be found in the Appendix D of the supplementary material.

In FIGURE 3 and FIGURE 4, we find that our algorithms have obvious advantages in prediction accuracy by comparing with the CRJC which only consider the local information for calculating the community relevance index. Moreover, in Amazon network, it is shown that the AUC of our proposed algorithms is better than that of other algorithms, but the Precision of our proposed method is not as good as that of CID-IP algorithm. In Power network, both the AUC and the Precision of our proposed algorithms are better than that of other methods. In addition, the variance of our proposed algorithms is smaller than that of the other comparison algorithms. This conclusion can be proved by the results in Appendix D of the supplementary material.

From FIGURE 1 to FIGURE 4, it is shown that the three community relevance indices we defined in this paper also have different performance. Each of them has good or bad performance. Then, we analyze this phenomenon.

D. COMPARED OF THREE DIFFERENT COMMUNITY INDICES

Compared these three community relevance indices, CR-Katz performs the best, followed by CR-LHN2 and

TABLE 2. Count the numbers of effective solutions in the community relevance matrixes by using different community relevance indices for the benchmark networks when the observations is 90%, here 100_2_4 means $N = 100$, $< k >= 2$, $k_{max} = 4$ 100_10_25 means $N = 100$, $< k >= 10$, $k_{max} = 25$, 1000_4_10 means $N = 1000$, $< k >= 4$, $k_{max} = 10$, 1000_20_50 means $N = 1000$, $< k >= 20$, $k_{max} = 50$, 10000_50_125 means $N = 10000$, $< k >= 50$, $k_{max} = 125$, 10000_200_500 means $N = 10000$, $< k >= 200$, $k_{max} = 500$, μ values from 0.1 to 0.5.

network	μ	CR_katz	CR_LHN2	CR_LP	network	μ	CR_katz	CR_LHN2	CR_LP
100_2_4	0.1	2	2	2	1000_20_50	0.1	525	527	740
100_2_4	0.2	4	4	4	1000_20_50	0.2	2429	2713	9286
100_2_4	0.3	2	2	2	1000_20_50	0.3	3871	4922	3891
100_2_4	0.4	14	14	12	1000_20_50	0.4	3139	3786	2890
100_2_4	0.5	4	4	4	1000_20_50	0.5	1981	1846	1344
100_10_25	0.1	20	20	20	10000_50_125	0.1	19880	19906	19880
100_10_25	0.2	92	92	89	10000_50_125	0.2	29935	30001	29935
100_10_25	0.3	400	396	326	10000_50_125	0.3	26197	19517	26197
100_10_25	0.4	493	493	390	10000_50_125	0.4	16046	10357	16046
100_10_25	0.5	483	483	375	10000_50_125	0.5	8683	4973	8683
1000_4_10	0.1	188	188	164	10000_200_500	0.1	629	629	629
1000_4_10	0.2	272	272	216	10000_200_500	0.2	2234	2236	2234
1000_4_10	0.3	262	262	200	10000_200_500	0.3	101528	95068	101528
1000_4_10	0.4	216	216	171	10000_200_500	0.4	97129	90317	97129
1000_4_10	0.5	175	175	140	10000_200_500	0.5	41016	31552	41016

TABLE 3. Count the numbers of effective solutions in the community relevance matrixes which we calculated according to the different community relevance indices for the real networks when the observations is 90%.

network	CR_Katz	CR_LHN2	CR_LP
Amazon	1180	1220	1222
Netscience	178	178	170
PGP	4167	5390	4255
Power	443	443	335
Protein	399	399	339
Ssc	286	286	255

CR-LP. According to the principle of our proposed prediction methods, we know that except over fitting, only the number of effective solutions of the community relevance matrix is abundant could ensure the prediction results accurate. Here the effective solutions mean the elements in the community relevance matrix whose value is not zero. For quantitative analysis, we count the numbers of effective solutions of the community relevance matrixes which obtained from the community division when $\lambda = 0.5$. In TABLE 2 and TABLE 3, we found that the predictive performance of the three indicators is different because the numbers of effective solutions in the community relevance matrixes are not the same.

In TABLE 2, the number of effective solutions of the community relevance matrixes obtained by different community relevance indices are the same when the number of the nodes is 100, the corresponding prediction accuracy curves of the three proposed algorithms almost coincide in figure 1 and figure 3. With the increase of the number of nodes, the number of effective solutions is different. Among them, CR-Katz index corresponds to more effective solutions. So, CR-Katz performs the best, followed by CR-LHN2 and CR-LP. From TABLE 3, we find that it is similar when we experiment on real-world networks.

E. COMPUTATIONAL COMPLEXITY

The runtime of CR-Katz, CR-LHN2 and CR-LP are mainly spent in the first and second processes. For a network with N nodes, the time complexity of community division is $O(cN^2)$, where, c is the number of λ. In the process of calculating the community relevance matrix, the time complexity is $O(1) \sim O(N^3)$. So, in the best case, the time complexity of our algorithm is $O(cN^2)$, and in the worst case, the time complexity of our algorithm is $O(cN^2 + N^3)$.

Then, we summarize the time complexity of the algorithms and show them in Table 4.

VI. CONCLUSION

In this paper, we combine the multi-resolution community division model and the traditional community relevance-based methods to propose a more realistic link prediction model which based on a novel quasi-local community relevance index under multi-resolution community division. This realistic link prediction model can be used not only to predict the different probability of the missing links between node-pairs within a community or between a specific pair of

TABLE 4. The time complexity of the algorithms.

algorithm	Time complexity	algorithm	Time complexity
Yan's algorithm	It takes $O(N^2)$ to calculate the probability of missing links.	CRJC	$O(N^2)$
CAR	$O(N^2 \sim N^4)$	CP	$O(kN^2)$
CID-IP	$O(D_{avg}(\mid E \mid)+\tau \mid V \mid +l^2 C_{avg})$ a*	node2vec	$O(\mid V \mid d)$ c*
SBM	In the worst case, it takes exponential time to sample different partitions. b*	SPM	$O(N^3)$ d*
SEAL	It takes $O(\mid V \mid \mid E \mid d)$ to train the graph neural network.	ARVGE	It takes $O(TK \mid V \mid d)$ to train the graph neural network. e*

a* D_{avg} denotes the average degree of nodes, τ represents the number of iterations in the partition phrase. l represents the number of communities.

b*The number of distinct partitions of N elements into groups is $\sum_{k=1}^{N} \frac{1}{k!} \sum_{l=1}^{k} \binom{k}{l} (-1)^{k-l} l^N$, which grows faster than any finite power of N [16].

c* d express the dimensionality of the embedding.

d* The time complexity of computing the eigenvalues of the matrix is $O(N^3)$.

e* T express the number of iterations, K express the number of steps for iterating discriminator, d express the dimensionality of the embedding.

communities, but also to predict the probability of the missing links between node-pairs with far distance.

The results show that the performance of our algorithm will change for the worse with the weakening of the network community structure. It is because that the performance of the proposed algorithms is based on the results of community division. The community division will be inaccurate if the community structure of the network is not good. This inaccurate leads to the deterioration of the prediction accuracy. Therefore, the performance of our algorithm is sensitive to the network community structure. Moreover, in the networks with the same scale, the sparser the network is, the better the prediction performance of the proposed algorithms is. It is because that our algorithms try to predict the link probability from the middle perspective of the network, and is taken into account the association between communities. So, it is more suitable for predicting the existence probability of target links if the number of interconnections between communities is small. In addition, the prediction performance of our proposed algorithms, including the AUC, Precision and the variance, is better than that of other comparison algorithms. According to the statistics of effective solutions of the community relevance matrix, it is shown that CR-Katz performs the best, followed by CR-LHN2 and CR-LP.

REFERENCES

[1] S. Li, X. Song, H. Lu, L. Zeng, M. Shi, and F. Liu, "Friend recommendation for cross marketing in online brand community based on intelligent attention allocation link prediction algorithm," *Expert Syst. Appl.*, vol. 139, Jan. 2020, Art. no. 112839, doi: 10.1016/j.eswa.2019.112839.

[2] X. Zhu, X. Yang, C. Ying, and G. Wang, "A new classification algorithm recommendation method based on link prediction," *Knowl.-Based Syst.*, vol. 159, pp. 171–185, Nov. 2018.

[3] A. Kumar, S. S. Singh, K. Singh, and B. Biswas, "Link prediction techniques, applications, and performance: A survey," *Phys. A, Stat. Mech. Appl.*, vol. 553, Sep. 2020, Art. no. 124289, doi: 10.1016/j.physa.2020.124289.

[4] J. Tang, J. Zhang, R. Jin, Z. Yang, K. Cai, L. Zhang, and Z. Su, "Topic level expertise search over heterogeneous networks," *Mach. Learn.*, vol. 82, no. 2, pp. 211–237, Feb. 2011.

[5] M. E. J. Newman, "Clustering and preferential attachment in growing networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 64, no. 2, Art. no. 025102, Jul. 2001.

[6] C. V. Cannistraci, G. Alanis-Lobato, and T. Ravasi, "From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks," *Sci. Rep.*, vol. 3, no. 1, pp. 1–14, Dec. 2013.

[7] Q. Zhang, M. Li, and Y. Deng, "Measure the structure similarity of nodes in complex networks based on relative entropy," *Phys. A, Stat. Mech. Appl.*, vol. 491, pp. 749–763, Feb. 2018.

[8] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *Eur. Phys. J. B*, vol. 71, no. 4, pp. 623–630, Oct. 2009.

[9] S. Li, J. Huang, J. Liu, T. Huang, and H. Chen, "Relative-path-based algorithm for link prediction on complex networks using a basic similarity factor," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 30, no. 1, Jan. 2020, Art. no. 013104.

[10] E. A. Leicht, P. Holme, and M. E. J. Newman, "Vertex similarity in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 73, no. 2, Feb. 2006, Art. no. 026120.

[11] L. Lü, C.-H. Jin, and T. Zhou, "Similarity index based on local paths for link prediction of complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 4, Oct. 2009, Art. no. 046122.

[12] A. Clauset, C. Moore, and M. E. J. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, pp. 98–101, May 2008.

[13] R. Guimerà and M. Sales-Pardo, "Missing and spurious interactions and the reconstruction of complex networks," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 52, pp. 22073–22078, Dec. 2009.

[14] M. E. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *J. Mach. Learn. Res.*, vol. 9, pp. 1981–2014, Sep. 2008.

[15] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 243–252.

[16] L. Duan, S. Ma, C. Aggarwal, T. Ma, and J. Huai, "An ensemble approach to link prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 11, pp. 2402–2416, Nov. 2017.

[17] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proc. KDD*, Aug. 2016, pp. 1105–1114.

[18] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. KDD*, Aug. 2016, pp. 855–864.

[19] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. KDD*, 2014, pp. 701–710.
[20] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. KDD*, 2016, pp. 1225–1234.
[21] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *Proc. AAAI*, 2016, pp. 1145–1152.
[22] T. N. Kipf and M. Welling, "Variational graph auto-encoders," in *Proc. NIPS*, 2016, pp. 1–3.
[23] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *Proc. NIPS*, 2018, pp. 5165–5175.
[24] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 1–7.
[25] B. Yan and S. Gregory, "Finding missing edges in networks based on their community structure," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 85, no. 5, May 2012.
[26] J. Ding, L. Jiao, J. Wu, Y. Hou, and Y. Qi, "Prediction of missing links based on multi-resolution community division," *Phys. A, Stat. Mech. Appl.*, vol. 417, pp. 76–85, Jan. 2015.
[27] J. Ding, L. Jiao, J. Wu, and F. Liu, "Prediction of missing links based on community relevance and ruler inference," *Knowl.-Based Syst.*, vol. 98, pp. 200–215, Apr. 2016.
[28] P. Wang, B. Xu, Y. Wu, and X. Zhou, "Link prediction in social networks: The state-of-the-art," *Sci. China Inf. Sci.*, vol. 58, no. 1, pp. 1–38, Jan. 2015.
[29] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *J. Amer. Soc. for Inf. Sci. Technol.*, vol. 58, no. 7, pp. 1019–1031, 2007.
[30] J. Zhang, Z. Fang, W. Chen, and J. Tang, "Diffusion of 'following' links micro-blogging networks," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 8, pp. 2093–2106, Feb. 2015.
[31] R. N. Lichtenwalter, "New precepts and method in link prediction," in *Proc. ACM KDD*, vol. 10, 2010, pp. 243–252.
[32] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Phys. A, Stat. Mech. Appl.*, vol. 390, no. 6, pp. 1150–1170, Mar. 2011.
[33] S. S. Singh, S. Mishra, A. Kumar, and B. Biswas, "CLP-ID: Community-based link prediction using information diffusion," *Inf. Sci.*, vol. 514, pp. 402–433, Apr. 2020.
[34] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.
[35] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
[36] L. Lü, L. Pan, T. Zhou, Y.-C. Zhang, and H. E. Stanley, "Toward link predictability of complex networks," *Proc. Nat. Acad. Sci. USA*, vol. 112, no. 8, pp. 2325–2330, Feb. 2015.
[37] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 78, no. 4, Oct. 2008, Art. no. 046110.
[38] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 5, Nov. 2009, Art. no. 056117.
[39] L. Ma, M. Gong, J. Liu, Q. Cai, and L. Jiao, "Multi-level learning based memetic algorithm for community detection," *Appl. Soft Comput.*, vol. 19, pp. 121–133, Jun. 2014.

JINGYI DING (Member, IEEE) received the B.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2009 and 2017, respectively. She is currently a Lecturer at the School of Artificial Intelligence, Xidian University. From 2019 to 2020, she was a Visiting Scholar with the Computational Optimisation and Learning Lab, Department of Computer Science, University of Nottingham, U.K. Her research interests include network science, machine learning, and graph classification.

JIAN SONG is currently pursuing the degree with the School of Artificial Intelligence, Xidian University, Xi'an, China. He is also a Graduate Student at Xidian University. His main learning content is graph neural networks.

LICHENG JIAO (Fellow, IEEE) received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 1982, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

From 1990 to 1991, he was a Postdoctoral Fellow with the National Key Laboratory for Radar Signal Processing, Xidian University, Xi'an. Since 1992, he has been a Professor with the School of Electronic Engineering, Xidian University, where he is currently the Director of the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education of China, International Research Center of Intelligent Perception and Computation. His current research interests include intelligent information processing, image processing, machine learning, and pattern recognition. He is the President of the Computational Intelligence Chapter, the IEEE Xi'an Section, and the IET Xi'an Network, the Chairman of the Awards and Recognition Committee, the Vice Board Chairperson of the Chinese Association of Artificial Intelligence, a Councilor of the Chinese Institute of Electronics, a Committee Member of the Chinese Committee of Neural Networks, a member of the IEEE Xi'an Section Execution Committee, and an Expert of the Academic Degrees Committee of the State Council. He was a recipient of the Second Prize of the National Natural Science Award, in 2013.

JIANSHE WU (Senior Member, IEEE) received the B.S. degree from the Department of Physics, Sichuan University, China, in 1992, and the M.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2001 and 2007, respectively. He is currently a Professor at the School of Electronic Engineering, Xidian University, Xi'an. He is the author or coauthor of more than 50 papers in journals and conferences. His research interests include complex networks, data mining, computational intelligence, and recommendation systems.

FANG LIU (Senior Member, IEEE) received the B.S. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 1984, and the M.S. degree in computer science and technology from Xidian University, Xi'an, in 1995. She is currently a Professor at the School of Computer Science, Xidian University. She is the author or coauthor of five books and more than 80 papers in journals and conferences. Her research interests include intelligent information processing, perception and image recognition, and data mining.