Chaos ARTICLE scitation.org/journal/cha

Network embedding for link prediction: The pitfall and improvement

Cite as: Chaos **29**, 103102 (2019); doi: 10.1063/1.5120724
Submitted: 20 July 2019 · Accepted: 10 September 2019 ·
Published Online: 8 October 2019

Ren-Meng Cao, Si-Yuan Liu, and Xiao-Ke Xu[a)]

AFFILIATIONS

College of Information and Communication Engineering, Dalian Minzu University, Dalian 116600, China

Note: This paper is part of the Focus Issue, "When Machine Learning Meets Complex Systems: Networks, Chaos and Nonlinear Dynamics."
[a)]**Electronic mail:** xuxiaoke@foxmail.com

ABSTRACT

Link prediction plays a significant role in various applications of complex networks. The existing link prediction methods can be divided into two categories: structural similarity algorithms in network domain and network embedding algorithms in the field of machine learning. However, few researchers focus on comparing these two categories of algorithms and exploring the intrinsic relationship between them. In this study, we systematically compare the two categories of algorithms and study the shortcomings of network embedding algorithms. The results indicate that network embedding algorithms have poor performance in short-path networks. Then, we explain the reasons for this phenomenon by computing the Euclidean distance distribution of node pairs after a given network has been embedded into a vector space. In the vector space of a short-path network, the distance distribution of existent and nonexistent links are often less distinguishable, which can sharply reduce the algorithmic performance. In contrast, structural similarity algorithms, which are not restricted by the distance function, can represent node similarity accurately in short-path networks. To address the above pitfall of network embedding, we propose a novel method for link prediction aiming to supplement network embedding algorithms with local structural information. The experimental results suggest that our proposed algorithm has significant performance improvement in many empirical networks, especially in short-path networks. AUC and Precision can be improved by 36.7%–94.4% and 53.2%–207.2%, respectively.

Link prediction has become a hot topic in the field of complex networks, because it has not only extensive applications in many fields but also profound theoretical significance. Currently, widely used algorithms of link prediction can be divided into two categories: structural similarity algorithms in the network domain and network embedding algorithms introduced from the field of machine learning. In previous studies, most researchers were enthusiastic about proposing more accurate prediction algorithms, paying less attention to the inherent relationship between the two categories of algorithms, especially the drawbacks of network embedding algorithms. In this study, by comparing the two categories of algorithms, it is found that network embedding algorithms do not outperform structural information algorithms in short-path networks. Therefore, a novel approach has been proposed to address this pitfall. In the proposed method, we supplement network embedding algorithms with local structural similarity to improve predictive performance. The experimental results show that the addition of local structure information in the network domain can not only address the pitfall of network embedding algorithms in short-path networks but also can bring significant performance improvement.

I. INTRODUCTION

Link prediction aims at predicting the likelihood of the existence of links between nodes based on the known network structural information, which has a wide range of applications in diverse domains.[1] For example, it can discover new protein-protein interactions in biological networks[2,3] and help people find potential friends in social networks.[4,5] Therefore, a large number of link prediction algorithms have been proposed in recent years.[6–8]

The existing link prediction algorithms can be divided into two categories: structural similarity algorithms in network domain[1] and network embedding algorithms in the field of machine learning.[9] Among the algorithms of structural similarity, local similarity indices

are the most commonly used because of their low computational complexity, such as Common Neighbor (CN) and Local Path (LP) indices.[1]

Recently, the technique of network embedding has been widely applied in link prediction.[9–11] It aims to map network data into a low dimensional space in which the network neighborhood information is maximumly preserved.[9] By representing nodes in a network as vectors, a wide variety of machine learning algorithms can be used to provide a standard, general and effective solution for link prediction.[12] Besides, network embedding has benefited many applications like network visualization,[13,14] node classification,[15,16] and node clustering.[17,18] However, in the current researches, there is a lack of systematic comparison of the two algorithms (structural similarity vs network embedding), and few of them study the shortcomings of network embedding algorithms.

To solve this problem, we systematically compare the two categories of link prediction algorithms. The results demonstrate that network embedding algorithms have poor performance in short-path networks. A short-path network refers to the network where the shortest path length between most pairs of nodes is short. When a short-path network is embedded in a vector space, there is an indistinguishable similarity distribution between the existent and nonexistent links. In contrast, structural similarity algorithms are not constrained by the distance function, and local structural information like the number of common neighbors, high-order paths as well as community information can efficiently characterize node similarity accurately in short-path networks. Therefore, we propose a novel method to address this shortcoming of network embedding algorithms, which supplements network embedding algorithms with local structure information. The newly proposed method is parameter-free and an accurate prediction result can be obtained by setting a fixed parameter. Experimental results on real-life networks demonstrate that the proposed method can greatly improve the accuracy of link prediction, especially in short-path networks.

The paper is organized as follows. The empirical network data and the algorithms of link prediction are introduced in Sec. II. In Sec. III, we report the main results of this study. We finally offer the conclusion in Sec. IV.

II. DATA AND METHODS

A. Data description

- Email.[17] This network was generated using the emails from a large research institution. Nodes represent the institution members and edges represent the communications between them. It has 167 nodes and 3257 edges.
- Ht09.[19] This dataset was collected during the 2009 ACM Hypertext Conference. Nodes represent attendees and edges represent contacts between them. The network is dynamic and has 113 nodes and 2196 edges.
- WorldCities (WC).[20] This dataset was used for global urban analysis. Nodes represent cities and firms and edges indicate that there is a service relationship between them. The network has 413 nodes and 7517 edges.
- Power.[21] It is an electrical power grid of western US. Nodes represent generators and transformers and edges indicate the high voltage transmission lines between them. The network has 4941 nodes and 6594 edges.
- Bcspwr10 (BP10).[22] In the Bcspwr10 network, nodes represent generators and edges represent the high voltage transmission lines. The network has 5300 nodes and 8271 edges.
- Minnesota (MN).[22] It is a road network with nodes representing intersections and edges representing roads connecting the intersections. It has 3309 nodes and 2642 edges.

B. Problem description and evaluation metrics

In this study, we only consider the problem of link prediction in undirected networks. We define an undirected network $G(V, E)$, where V denotes the set of nodes and E denotes the set of links. The universal set U contains all $\frac{|V|\cdot(|V|-1)}{2}$ possible links, where $|V|$ denotes the number of elements in set V. Then, the set of nonexistent links is $U - E$. To evaluate the accuracy of a link prediction method, we randomly divide real network data into two parts: the training set E^T and testing set E^P, where $E = E^T \cup E^P$ and $E^T \cap E^P = \emptyset$. In the experiments, 90% of the links are selected as the training set, and the remaining 10% of the links constitute the testing set.

We use two standard metrics, the Area Under the receiver operating characteristic Curve (AUC)[23] and Precision,[24] to measure the accuracy of link prediction algorithms. A link prediction algorithm gives a score S_{xy} for each link in the testing set to qualify its existence likelihood. The value of AUC measures the accuracy of algorithms based on the entire testing set. The value of Precision focuses on whether the L links with the highest scores are predicted accurately. In our experiments, L is equal to 10% of the number of links in the testing set. In the following, a detailed description of the two metrics is given.

The AUC can be viewed as the probability that the score of an existent link (i.e., a link in E^P) is higher than a randomly chosen nonexistent link (i.e., a link in $U - E$). In the implement, we compare the scores of a randomly chosen existent link and a randomly chosen nonexistent link, and compare them independently for n times. If there are n' times that the score of a missing link is higher than that of a nonexistent link, and n'' times they have the same score, the value of AUC is defined as

$$AUC = \frac{n' + 0.5n''}{n}. \tag{1}$$

Ranked the predicted links by the scores in a descending order, the Precision is defined as the radio of links predicted correctly in the set of top-L links (i.e., the top L links with score ranks). If there are m links being predicted accurately in the set of top-L links, that is to say, the m links in top-L links belong to the set of existent links, and then the value of Precision is defined as

$$Precision = \frac{m}{L}. \tag{2}$$

The value of Precision is related to the parameter L. For a given L, the higher the value of Precision, the more accurate the algorithm is. If two algorithms have a similar AUC value but the Precision value of Algorithm1 is higher than Algorithm2, Algorithm1 is superior to Algorithm2. A more accurate prediction result can be obtained by utilizing Algorithm1, which only needs to check a small part of links with top ranks.

C. Structural similarity algorithms

1. Common neighbor

In the algorithms based on structural similarity, the simplest index for measuring the likelihood of generating a link in the future between two nodes is the Common Neighbor (CN) index.[25] The more the common neighbors two nodes share, the higher the probability of generating a link between them is. The CN index can be defined as follows. For a given pair of nodes (x, y), $\Gamma(x)$ and $\Gamma(y)$ represent the sets of neighbors of nodes x and y, respectively. The similarity between nodes x and y can be calculated as the number of their common neighbors, namely,

$$S_{xy}^{CN} = \left|\Gamma(x) \cap \Gamma(y)\right|. \tag{3}$$

2. Local path

The index of Local Path (LP) for link prediction was proposed by Zhou *et al.*,[26] which uses the second-order neighbor to preserve more structural information. The LP index considers the next nearest neighbors based on common neighbors, which is defined as

$$S = A^2 + \alpha A^3, \tag{4}$$

where α denotes an adjustable parameter and A is the adjacency matrix. A^2 represents the number of paths with a length of 2 between node x and y and A^3 represents the number of paths with a length of 3 between node x and y. The parameter of α is usually taken as 0.01.

3. Community common neighbor

Soundarajan and Hopcroft proposed a new index for link prediction, named Community Common Neighbor (CCN), which proves the inclusion of community information can improve the accuracy of local similarity indices.[27] The CCN index consist of two parts: the similarity index and the number of common neighbors in the same community, which is defined by

$$CCN(x, y) = CN(x, y) + \sum_{i \in \Gamma(x,y)} \left|C(x) \cap C(y)\right|, \tag{5}$$

where $CN(x, y)$ is the number of neighbors shared by x and y and $\Gamma(x, y)$ denote the neighborhood shared by x and y. Here, $C(x)$ denote the community that node x belongs to and $C(y)$ is the community that node y belongs to. There is no need in this equation that node i is a common friend of nodes x and y. If node i belongs to both the community of node x and the community of node y, the second half is going to add 1.

D. Network embedding algorithms

Recently, the technique of network embedding, aiming to convert a network into low dimensional vectors, has been widely applied in link prediction.[9–11] It can map network data into a low dimensional space in which the network neighborhood information is maximumly preserved.[9] After representing nodes in a network as vectors, a wide variety of machine learning algorithms can be used to provide a standard, general, and effective solution for link prediction.[12] In this study, we use the four classical network embedding algorithms containing Node2vec,[28] LargeVis,[14] LINE,[29] and GraphWave[30] for link prediction. The output vectors of these embedding algorithms can preserve rich network neighborhood information including different orders of structural similarity (e.g., Node2vec, LargeVis, and LINE) and multiscales of structure equivalence (e.g., GraphWave), so they can be applied to predict missing links and have achieved high performance in link prediction.[9]

1. Node2Vec

This algorithm of Node2Vec adopts a flexible search strategy, named the second-order random walks, which can explore diverse network structural information.[28] For example, nodes with high connection density and belonging to the same community or cluster are tightly embedded together, that is, homogeneity. Using another way, the nodes with similar structural roles in a network can be embedded together, that is, structural equivalence. Obtaining the two types of structure similarity information based on a special random walk strategy, this algorithm is constructive to link the prediction problem and helps to preserve diverse types of neighborhood information, which makes the prediction results more accurate.

2. LargeVis

The highlight of the algorithm is utilizing a random projection tree for the construction of KNN graphs, which significantly reduces time and space costs.[14] Then, the algorithm utilizes a principled probabilistic model to map a network into a low dimension space. The model maximizes the existence likelihood of existent links, which makes similar nodes keep close to together in the low dimensional space. At the same time, it also minimizes the existence likelihood of all the nonexistent links, which makes dissimilar nodes far away from each other. Therefore, LargeVis is conductive to link prediction which makes the same category nodes close and different category nodes far away from each other.

3. LINE

This algorithm preserves the structural information of a network from two different aspects: first-order proximity and second-order proximity.[29] The former refers to the similarity of the connection information between two nodes. For example, in social networks, people who are friends are more likely to possess similar interests and hobbies. The second-order proximity refers to the similarity of common neighbor information between two nodes. For example, the words used with many other similar words also tend to have a similar meaning. Hence, the rich structure information preserved by LINE can handle the link prediction tasks.

4. GraphWave

GraphWave learns a multidimensional vector representation for each node according to the diffusion of a spectral graph wavelet centered at the node.[30] When a node propagates an energy unit on the graph, node similarity is characterized by the diffusion pattern of graph wavelets. GraphWave mathematically proves that nodes with similar neighbor information have the similar diffusion pattern, which can be closely embedded together in the vector space. In the problem of link prediction, structural equivalence is a kind of

TABLE I. The AUC results of network domain algorithms and network embedding algorithms with different average shortest path length, results averaging over 10 systematic experiments. $\langle d \rangle$ denotes the statistic of average shortest path length. In the table, we divided networks into two parts: short-path networks ($\langle d \rangle < 3$) and long-path networks ($\langle d \rangle > 15$). It is obvious that network embedding algorithms perform poorly in short-path networks.

AUC	Ht09	Email	WC	Power	BP10	MN
CN	**0.776**	**0.932**	0.171	0.615	0.600	0.528
LP	0.757	0.920	**0.953**	0.689	0.695	0.553
CCN	0.760	0.910	0.912	0.872	0.915	0.736
Node2Vec	0.531	0.546	0.667	0.863	0.938	0.801
LargeVis	0.506	0.478	0.556	**0.933**	**0.966**	**0.842**
LINE	0.618	0.730	0.919	0.613	0.466	0.486
GraphWave	0.484	0.627	0.732	0.507	0.541	0.587
$\langle d \rangle$	1.65	1.96	2.22	18.98	20.85	35.30

TABLE II. The Precision results of network domain algorithms and network embedding algorithms with different average shortest path length, results averaging over 10 systematic experiments. $\langle d \rangle$ denotes the statistic of average shortest path length. In the table, we divided networks into two parts: short-path networks ($\langle d \rangle < 3$) and long-path networks ($\langle d \rangle < 15$). It can be seen that network embedding algorithms have a bad performance in short-path networks.

Precision	Ht09	Email	WC	Power	BP10	MN
CN	0.690	0.930	0.000	0.950	**1.000**	0.696
LP	**0.750**	**0.960**	**0.990**	0.970	**1.000**	**0.969**
CCN	0.740	**0.960**	0.930	0.940	0.960	0.901
Node2vec	0.383	0.376	0.407	0.969	0.987	**0.969**
Largevis	0.309	0.430	0.366	**0.984**	0.987	**0.969**
LINE	0.572	0.660	0.890	0.570	0.360	0.330
GraphWave	0.484	0.627	0.732	0.507	0.541	0.587
$\langle d \rangle$	1.65	1.96	2.22	18.98	20.85	35.30

extremely important structural information which does not focus on connectivity. Even if nodes belong to different parts of the network, they might have the same structural roles.

III. MAIN RESULTS

A. The pitfall of network embedding for short-path networks

In this section, we firstly compare these two categories of algorithms through link prediction in six real-life networks. Table I displays the AUC results of these algorithms. In each column, the best result is highlighted in bold. In all the empirical networks, some networks have relatively longer $\langle d \rangle$ and some have shorter ones. Actually, many real-life networks show "small-world" phenomena.[21] However, as far as we know, up to now there is no clear definition of short-path and long-path networks. In order not to cause a controversy, the network with $\langle d \rangle < 3$ is considered as short-path networks in this study. Meanwhile, networks with $\langle d \rangle < 15$ are called long-path networks.

As is shown in Table I, it can be seen that for short-path networks (i.e., Ht09, Email, and WC networks), structural similarity algorithms (e.g., CN, LP, and CCN) can achieve excellent predictive performance. Furthermore, structural similarity algorithms significantly outperform network embedding algorithms in short-path networks. Conversely, for long-path networks, the best predictive performance is obtained from network embedding algorithms (e.g., LargeVis and Node2Vec). In other words, network embedding algorithms have a serious deficiency when performing link prediction in short-path networks.

Note that the poor performance of CN in the WC network is due to the lack of common neighbor information, that is, there are no links with a distance of 2. Except for WC and Email networks, LINE performs poorly in both short-path and long-path networks, which is caused by the inflexibility of preserving network structure information.[28] Likewise, the poor performance of GraphWave shown in empirical networks demonstrates that structural equivalence is not suitable for link prediction. In summary, since LINE and GraphWave perform poorly in both long-path and short-path networks, no complementary information between these algorithms and network domain algorithms can be utilized to address the pitfall of network embedding algorithms. In the following experiments, we do not fuse the information induced from the two methods for link prediction.

Next, the results of Precision for the two categories of algorithms are shown in Table II. In each column, the best results are also highlighted in bold. From the table, the Precision values of network embedding algorithms in short-path networks, especially Node2vec and LargeVis, are much lower than the counterparts in long-path networks. Overall, the results in Tables I and II illustrate that network embedding algorithms have a serious pitfall when performing link prediction in short-path networks.

To explain the phenomenon that network embedding algorithms have a pitfall when performing link prediction in short-path networks, six real networks are embedded into vector spaces and the Euclidean distances of node pairs are calculated. Figure 1 shows the distance distribution of existent and nonexistent links in different networks, and three short-path networks with lower $\langle d \rangle$ and three long-path networks with larger $\langle d \rangle$ are shown in Figs. 1(a)–1(c) and Figs. 1(d)–1(f), respectively. It is found that in short-path networks, the distribution of existent and nonexistent links overlaps to a large extent, which sharply reduces the algorithmic performance. Conversely, in long-path networks, the distances of existent links are mainly between 2 and 6, while the distances of nonexistent links are mainly between 6 and 8. These two categories of links are highly distinguishable, thus better predictive performance can be achieved in these networks.

B. Improving the performance of network embedding by supplementing local structure information

When a network is embedded into a low dimensional vector space, the similarity between two nodes can be characterized by a distance function. The closer the two nodes are embedded, the more similar they are. In Sec. III A, we find that for short-path networks, the distance distribution of existent and nonexistent links are less distinguishable, which can sharply reduce the performance of network embedding algorithms. In contrast, structural similarity algorithms

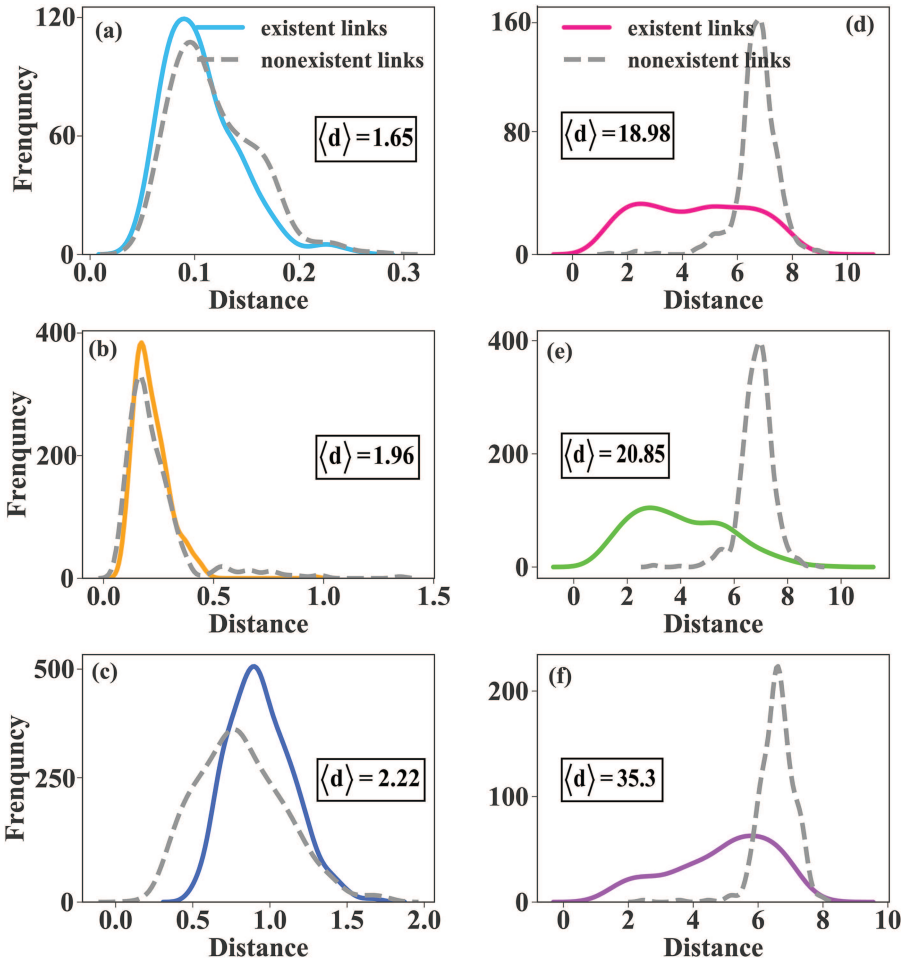

FIG. 1. The Euclidean distance distributions of node pairs (existent and nonexistent links) in the vector space after network embedding. (a) Ht09, (b) Email, (c) WorldCities, (d) Power, (e) Bcspwr10, and (f) Minnesota. (a)–(c) belong to short-path networks, (d)–(f) belong to long-path networks. It can be seen that the distribution of existent and nonexistent links overlaps a large degree in short-path networks. That is to say, it is hard to distinguish the two categories of links, which can greatly reduce the performance of network embedding algorithms.

are not restricted by the distance function but characterize node similarity by local structural information, which can give more accurate predictions in these networks.

Based on the above facts, we propose a novel link prediction method to improve the performance of network embedding algorithms, namely, Network Embedding Supplement the similarity information in the Network Domain (NESND), which supplements local structure information with the network embedding algorithm and is defined as

$$S_{NESND} = S_{NE} + \lambda S_{ND}, \tag{6}$$

where S_{NE} denotes the network embedding information represented by the Euclidean distance of node pairs and S_{ND} denotes local structure information from the network domain. S_{NESND} denotes the combined result and λ is a parameter that adjusts how much local structure information is added. $\lambda = 0$ denotes that none of local structure information is added, so Eq. (6) is equal to S_{NE}. $\lambda < 0$ denotes that local structure information is added to embedding algorithms, and the larger the λ is, the more the information is added. Note that we also take $\lambda < 0$ into account, this is because we attempt to analyze the correlation between the supplement of network domain information and the performance of NESND. If the supplement of network domain information improves the accuracy of NESND, which means that there is a positive correlation between local structure similarity and network embedding information, otherwise a negative correlation between them can be verified. Similarly, Shang *et al.* considered the case where the parameter is less than 0 in the newly proposed method of link prediction for treelike networks.[31] Moreover, Liu and Zhou also incorporated negative values into the selection of parameters to investigate the role of weak ties in link prediction.[32] To analyze the correlation between the supplement of network domain information and the performance of NESND, we set this parameter λ from negative to positive values. In some special cases (e.g., the WorldCities network), there is a negative correlation between structural information and network embedding. Therefore, when the parameter λ is less than zero, the performance of Node2vec+CN and LargeVis+CN can be improved as shown in Fig. 2(c).

We use the proposed algorithm for link prediction and the results are shown in Fig. 2. The AUC performance of short-path networks are shown in Figs. 2(a)–2(c), and the results of long-path networks are illustrated in Figs. 2(d)–2(f), respectively. The most

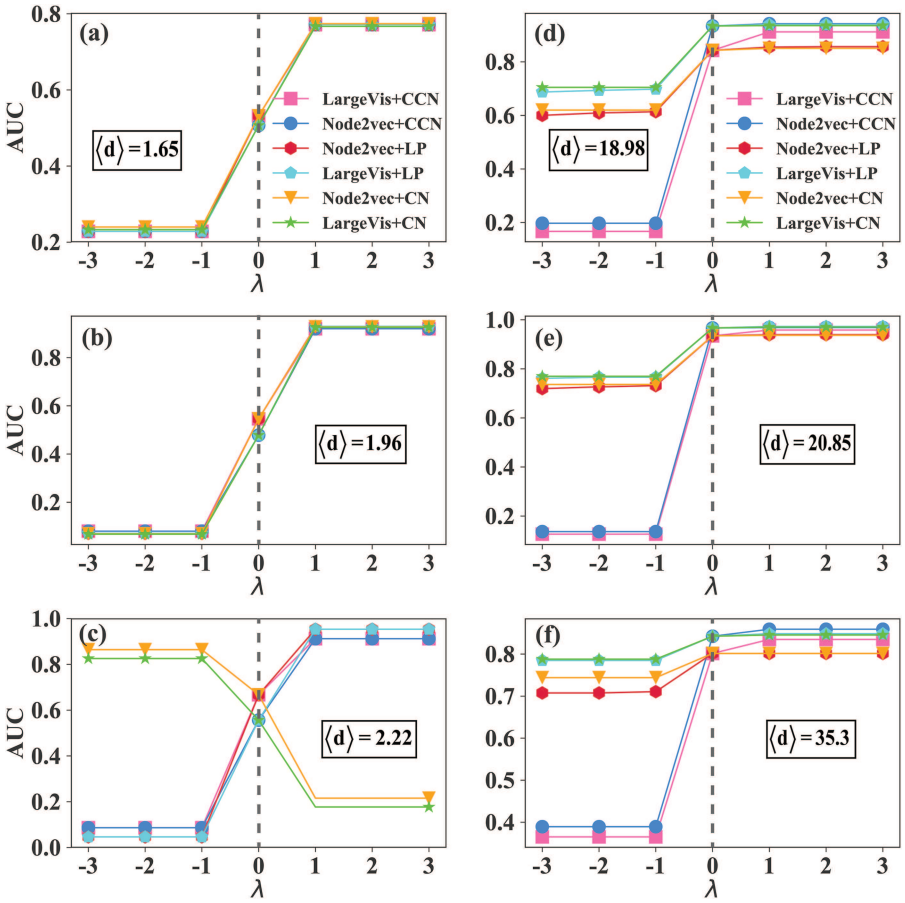

FIG. 2. The AUC results of different combination of network domain algorithms and network embedding algorithms in short-path and long-path networks, (a) Ht09, (b) Email, (c) WorldCities, (d) Power, (e) Bcspwr10, and (f) Minnesota. (a)–(c) belong to short-path networks and (d)–(f) belong to long-path networks. The results indicate that the accuracies of NESND are greatly improved in short-path network, which proves the validity of our proposed algorithm.

striking result to emerge from Figs. 2(a)–2(b) is that the accuracies of NESND are greatly improved when $\lambda > 0$. But there is an exception that Node2vec+CN and LargeVis+CN perform exactly opposite to other methods for the WorldCities network in Fig. 2(c). The key to our proposed method (i.e., NESND) is to utilize the complementary of structure similarity information for network embedding algorithms to improve predictive performance. Based on the analysis in Sec. III A, network embedding algorithms perform poorly in short-path networks. To enhance the performance of them, we have to select some network domain algorithms which perform well for complementing useful information. For example, the CN index performs well while Node2vec performs poorly in Ht09 and Email networks, supplementing common neighbor information with the information preserved by Node2vec can bring performance improvement shown in Figs. 2(a) and 2(b), respectively. Conversely, CN, Node2vec, and LargeVis all perform poorly in the WorldCities network, supplementing common neighbor information with embedding information making less sense in performance improvement illustrated in Fig. 2(c).

In addition, the experimental results mentioned above suggest that the method of NESND is independent of the value of λ when it is greater than zero, and we can get a more accurate prediction result. Therefore, NESND is a strong robustness method for link prediction and we set parameter λ to 1 in the following experiments.

Table III list all the AUC values in detail. In each column, the best results are highlighted in bold. For the short-path networks, the performance improvements brought by the introduction of CN and LP are greater than CCN. This is because in short-path networks, compared with community structure information, the number of common neighbor and 3-order paths can more accurately characterize the node similarity. By contrast, for long-path networks, especially in Power and MN networks, the enhanced performance brought by the CCN index is more significant than CN and LP, this is because the special meso-scale structure information (i.e., community structure) can accurately represent node similarity in long-path networks. The proposed method has 0.2%–8.3% improvement in long-path networks, while 36.7%–94.4% improvement can be obtained in short-path networks.

Similarly, NESND can also bring significant enhancement in short-path networks for Precision, as shown in Fig. 3 when $\lambda > 0$. This is because supplementing network embedding algorithms with local structure information can make the existent and nonexistent links more distinguishable and more predictable. Conversely, for long-paths networks, the method of NESND has no strong effect on performance improvement. We conjecture that it is mainly because

TABLE III. The AUC improvement by the proposed NESND method. The AUC results of NESND are much greater than the baselines (Node2vec and LargeVis), which demonstrates that the proposed method can address the pitfall of network embedding algorithms for link prediction in short-path networks.

AUC	Ht09	Email	WC	Power	BP10	MN
Node2vec	0.531	0.546	0.667	0.842	0.934	0.801
LargeVis	0.506	0.478	0.556	0.933	0.966	0.842
Node2vec + CN	**0.774**	0.927	0.215	0.850	0.937	0.801
LargeVis + CN	0.767	0.926	0.177	0.935	0.970	0.845
Node2vec + LP	0.772	**0.929**	**0.953**	0.856	0.939	0.801
LargeVis + LP	0.772	**0.929**	**0.953**	0.939	**0.972**	0.848
Node2vec + CCN	0.772	0.920	0.912	0.912	0.958	0.834
LargeVis + CCN	0.772	0.920	0.912	**0.942**	0.968	**0.859**
$\langle d \rangle$	1.65	1.96	2.22	18.98	20.85	35.30

TABLE IV. The Precision improvement by the proposed NESND method. The AUC results of NESND are much greater than the baselines (Node2vec and LargeVis), which demonstrates that the proposed method can address the pitfall of network embedding algorithms for link prediction in short-path networks.

Precision	Ht09	Email	WC	Power	BP10	MN
Node2vec	0.528	0.315	0.425	0.910	0.987	0.954
LargeVis	0.442	0.346	0.367	0.975	0.987	**0.970**
Node2vec + CN	0.809	**0.969**	0.0	0.927	0.987	0.961
LargeVis + CN	0.809	**0.969**	0.0	0.973	0.987	**0.970**
Node2vec + LP	0.809	**0.969**	**0.987**	0.955	0.987	**0.970**
LargeVis + LP	0.809	**0.969**	**0.987**	0.962	0.987	**0.970**
Node2vec + CCN	**0.857**	**0.969**	0.933	**0.984**	0.987	0.909
LargeVis + CCN	**0.857**	**0.969**	0.933	**0.984**	0.987	0.961
$\langle d \rangle$	1.65	1.96	2.22	18.98	20.85	35.3

of the high performance of network embedding algorithms in these networks, adding structural information makes no sense for representing networks more accurately.

We list all the results measured by Precision in Table IV. Although the AUC values of Ht09 shown in Tables III and IV are almost the same, its Precision values of Node2vec+CCN and LargeVis+CCN are higher than other algorithms. This result demonstrates that the predictive ability of Node2vec+CCN and LargeVis+CCN is superior to others for the Ht09 network. Note that the Precision values of Node2vec+CN and LargeVis+CN is close

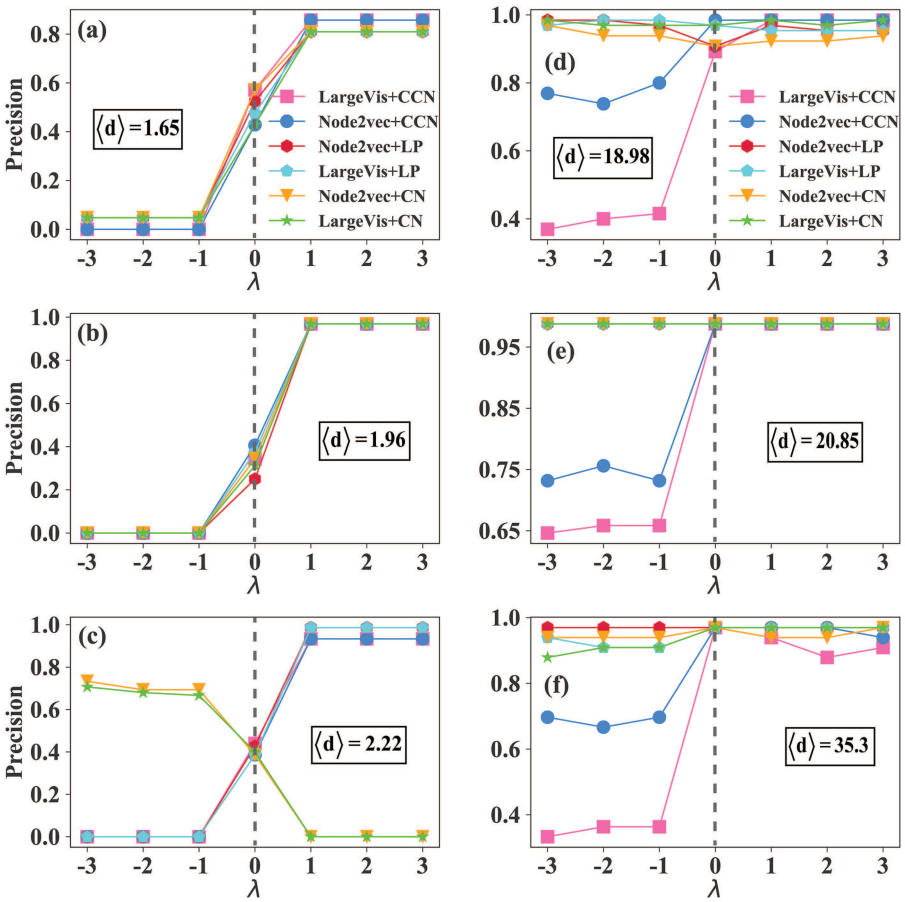

FIG. 3. The Precision results of different combination of network domain algorithms and network embedding algorithms in short-path and long-path networks. (a) Ht09, (b) Email, (c) WorldCities, (d) Power, (e) Bcspwr10 and (f) Minnesota. (a)–(c) belong to short-path networks, (d)–(f) belong to long-path networks. The results indicate that the accuracies of NESND can be greatly improved in short-path networks.

to 0, which is caused by the defective representation of the CN index, the AUC value is lower than 0.5 in the WC network. The results in Table IV show that NESND has 0.7%–8.1% improvement in long-path networks, while 53.2%–207.6% improvement can be obtained in short-path networks.

IV. CONCLUSION

In summary, we systematically compared the performance of structural similarity algorithms in the network domain and network embedding algorithms in the field of machine learning. The results reveal the pitfall of network embedding for link prediction: network embedding algorithms have poor performance in short-path networks. Then, six real networks are embedded into vector spaces and the Euclidean distances of node pairs are calculated to explain the above phenomenon. The results demonstrate that for short-path networks, the distance distribution of existent and nonexistent links are less distinguishable, thus network embedding algorithms perform poorly in these networks. On the contrary, structural similarity algorithms are not restricted by distance function but characterize node similarity by local structure information, thus good performance can be achieved in the category of short-path networks. Finally, we proposed a novel method supplementing network embedding algorithms with local structure information to address this pitfall. The results demonstrate that our proposed method can efficiently improve the accuracy of link prediction, especially in short-path networks.

In this work, various structure information is integrated into network embedding algorithms, which can provide new insights into link prediction. In the future research, we will analyze the relationship between other newly proposed embedding algorithms[33,34] and structural similarity algorithms and attempt to explore whether these novel algorithms have been overcome the shortcoming found in this study. Our proposed method can also be applicable to other fields which network embedding algorithms have been widely used in such as network visualization,[13,14] node classification,[15,16] and node clustering.[17,18]

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (NNSFC) (Grant Nos. 61603073 and 61773091), the Key Research and Development Plan of Liaoning Province (No. 2018104016), the LiaoNing Revitalization Talents Program (No. XLYC1807106), and the Program for the Outstanding Innovative Talents of Higher Learning Institutions of Liaoning (No. LR2016070).

REFERENCES

[1] Z.-M. Ren, A. Zeng, and Y.-C. Zhang, "Structure-oriented prediction in complex networks," Phys. Rep. **750**, 1–51 (2018).
[2] I. A. Kovács, K. Luck, K. Spirohn, Y. Wang, C. Pollis, S. Schlabach, W. Bian, D.-K. Kim, N. Kishore, and T. Hao *et al.* Network-based prediction of protein interactions," Nat. Commun. **10**, 1240 (2019).
[3] G. Crichton, Y.-F. Guo, S. Pyysalo, and A. Korhonen, "Neural networks for link prediction in realistic biomedical graphs: a multi-dimensional evaluation of graph embedding-based approaches," BMC Bioinf. **19**, 176 (2018).
[4] L. M. Aiello, A. Barrat, R. Schifanella, C. Cattuto, B. Markines, and F. Menczer, "Friendship prediction and homophily in social media," ACM Trans. Web **6**, 9 (2012).
[5] H.-B. Hu and X.-F. Wang, "Disassortative mixing in online social networks," Europhys. Lett **86**, 18003 (2009).
[6] K.-K. Shang, T.-C. Li, M. Small, D. Burton, and Y. Wang, "Link prediction for tree-like networks," Chaos **29**, 061103 (2019).
[7] X. Chen, L. Fang, T.-H. Yang, J. Yang, Z.-R. Bao, D.-Z. Wu, and J. Zhao, "The application of degree related clustering coefficient in estimating the link predictability and predicting missing links of networks," Chaos **29**, 053135 (2019).
[8] Z. Liu, W.-K. Dong, and Y. Fu, "Local degree blocking model for link prediction in complex networks," Chaos **25**, 013115 (2015).
[9] H.-Y. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," IEEE Trans. Knowl. Data Eng. **30**, 1616–1637 (2018).
[10] R. Brochier, A. Guille, and J. Velcin, "Link prediction with mutual attention for text-attributed networks," in *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19 (ACM, New York, NY, 2019), pp. 283–284.
[11] X.-M. Liang, D.-F. Li, M. Song, A. Madden, Y. Ding, and Y. Bu, "Predicting biomedical relationships using the knowledge and graph embedding cascade model," PLoS ONE **14**, 1–23 (2019).
[12] P. Cui, X. Wang, J. Pei, and W.-W. Zhu, "A survey on network embedding," IEEE Trans. Knowl. Data Eng. **31**, 833–852 (2018).
[13] L. V. D. Maaten and G. Hinton, "Visualizing data using t-sne," J. Mach. Learn. Res. **9**, 2579–2605 (2008).
[14] J. Tang, J.-Z. Liu, M. Zhang, and Q.-Z. Mei, "Visualizing large-scale and high-dimensional data," in *Proceedings of the 25th International Conference on World Wide Web, WWW '16* (IW3C2, Republic and Canton of Geneva, Switzerland, 2016), pp. 287–297.
[15] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node classification in social networks," in *Social Network Data Analytics*, edited by C. C. Aggarwal (Springer US, Boston, MA, 2011), pp. 115–148.
[16] S.-S. Cao, W. Lu, and Q.-K. Xu, "Grarep: Learning graph representations with global structural information," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15* (ACM, New York, NY, 2015), pp. 891–900.
[17] A. Tsitsulin, D. Mottin, P. Karras, and E. Müller, "Verse: Versatile graph embeddings from similarity measures," in *Proceedings of the 2018 World Wide Web Conference, WWW '18* (IW3C2, Republic and Canton of Geneva, Switzerland, 2018), pp. 539–548.
[18] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, "Local higher-order graph clustering," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17* (New York, NY, 2017), pp. 555–564.
[19] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, and W. Van den Broeck, "What's in a crowd? Analysis of face-to-face behavioral networks," J. Theor. Biol. **271**, 166–180 (2011).
[20] P. J. Taylor, "The new geography of global civil society: NGOs in the world city network," Globalizations **1**, 265–277 (2004).
[21] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," Nature **393**, 440–442 (1998).
[22] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI '15* (AAAI Press, 2015), pp. 4292–4293.
[23] J. A. Hanley and B. J. Mcneil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," Radiology **143**, 29–36 (1982).
[24] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. Riedl, "Evaluating collaborative filtering recommender systems," ACM Trans. Inf. Syst. **22**, 5–53 (2004).
[25] F. Lorrain and H. C. White, "Structural equivalence of individuals in social networks," J. Math. Sociol. **1**, 49–80 (1971).
[26] T. Zhou, L.-Y. Lü, and Y.-C. Zhang, "Predicting missing links via local information," Eur. Phys. J. B **71**, 623–630 (2009).
[27] S. Soundarajan and J. Hopcroft, "Using community information to improve the precision of link prediction methods," in *Proceedings of the 21st International Conference on World Wide Web, WWW '12 Companion* (New York, NY, 2012), pp. 607–608.

[28]A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'16* (New York, NY, 2016), pp. 855–864.
[29]J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web, WWW '15* (Republic and Canton of Geneva, Switzerland, 2015), pp. 1067–1077.
[30]C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, "Learning structural node embeddings via diffusion wavelets," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'18* (ACM, New York, NY, USA, 2018), pp. 1320–1329.
[31]K.-k. Shang, T.-c. Li, M. Small, D. Burton, and Y. Wang, "Link prediction for tree-like networks," Chaos **29**, 061103 (2019).
[32]L. Lü and T. Zhou, "Link prediction in weighted networks: The role of weak ties," Europhys. Lett. **89**, 18001 (2010).
[33]L. Qiao, H. Zhao, X. Huang, K. Li, and E. Chen, "A structure-enriched neural network for network embedding," Expert Syst. Appl. **117**, 300–311 (2018).
[34]Y. Wang, Y. Yao, H. Tong, F. Xu, and J. Lu, "A brief review of network embedding," Big Data Mining Anal. **2**, 35–47 (2019).