Received June 23, 2020, accepted July 2, 2020, date of publication July 8, 2020, date of current version August 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3007886

SINE: Second-Order Information Network Embedding

ZIQI WANG[1], YUANYUAN ZHANG[1,2], SHUDONG WANG[2], AND JUNLIANG SHANG[3], (Member, IEEE)
[1]School of Information and Control Engineering, Qingdao University of Technology, Qingdao 2665203, China
[2]College of Computer and Communication Engineering, China University of Petroleum (East China), Qingdao 266580, China
[3]School of Information Science and Engineering, Qufu Normal University, Rizhao 276826, China

Corresponding author: Yuanyuan Zhang (yyzhang1217@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61902430, Grant 61873281, and Grant 61972226, and in part by the Natural Science Foundation of Shandong Province under Grant ZR2018PF004.

ABSTRACT As an important data type, the demand of network analysis and learning is increasingly prominent. A key problem of network analysis is to study how to reasonably represent the feature information in the network, that is network embedding. However, in the study of network embedding, only the first-order proximity relationship of nodes is characterized, while the second-order proximity relationship hidden in the network nodes is ignored. Therefore, we propose an algorithm, called SINE, to realize the representation learning of nodes in the network which fuses the first-order and second-order proximity of nodes in the original network. Through applying SINE to three real networks, we obtain the feature representation of the nodes in the networks and cluster the nodes based on these features. Comparing with the existing network embedding learning methods—Node2vec, Large-Scale Information Network Embedding (LINE) and Structural Deep Network Embedding (SDNE), the SINE algorithm showed better performance in clustering tasks.

INDEX TERMS Network embedding, second-order proximity, representation learning.

I. INTRODUCTION

In the era of big data, network data is everywhere in the real world. The research and analysis of network data has attracted extensive attention from all walks of life. However, the network data is often complex, huge, and unstructured in real world, which makes it extremely difficult to obtain effective information in the network. In order to analyze and process network information, network embedding algorithm came into being [1]. The algorithm is responsible for learning from the network data to the vector embedding of each node in the network and is a bridge connecting the original network data with the network application task. Traditional methods can use high-dimensional sparse vectors to represent a node in the network, but the limitation is that it is difficult to measure proximity between nodes and it will increase the time and space complexity of the model. With the development of embedding learning technology, researchers have turned to mapping nodes in the network into vectors with certain reasoning ability into low-dimensional space, aiming to preserve the original relationship between nodes as much as possible in a more intuitive and efficient way, so as to conveniently serve as inputs to machine learning models, such as node classification [2], [3], community identification [4], link prediction [5] and network visualization [6] and so on.

At present, the methods of network embedding can be divided into two categories, as shown as Fig 1.

The first category can be divided into three sub-categories according to different methods: based on matrix eigenvectors and matrix decomposition, based on shallow neural network and based on deep learning. The first sub-category included LLE algorithm [7] (Locally Linear Embedding), Laplacian Eigenmaps algorithm [8], and GraRep algorithm [9] (Learning Graph Representations), where LLE algorithm and Laplacian Eigenmaps algorithm can only handle undirected network, but in real word, many networks are directed [8]. The main disadvantage of this kind of method is its complexity. Calculating eigenvectors of large-scale matrix consumes a lot of time and space. The second sub-category include DeepWalk [10], LINE [11] and Node2vec [12] algorithms. LINE explicitly define the corresponding loss function to maintain

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

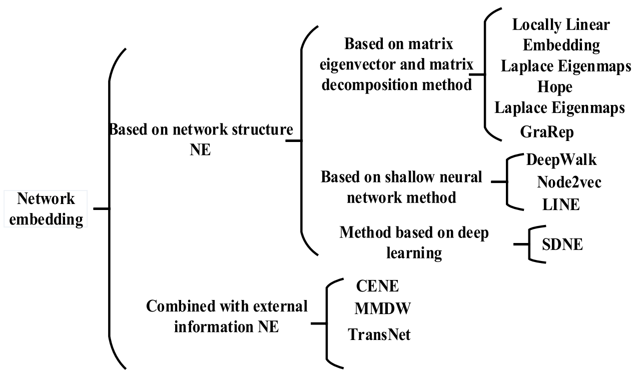
Network embedding
Based on network structure NE
Based on matrix eigenvector and matrix decomposition method
Locally Linear Embedding
Laplace Eigenmaps
Hope
Laplace Eigenmaps
GraRep
Based on shallow neural network method
DeepWalk
Node2vec
LINE
Method based on deep learning
SDNE
Combined with external information NE
CENE
MMDW
TransNet

FIGURE 1. Classification of network embedding algorithms.

the first and second order proximity between nodes. The performance of DeepWalk and LINE algorithm in network node label prediction task has surpassed the base traditional spectrum and optimization method, but DeepWalk and LINE algorithm lack of application cases in other social network analysis tasks, and the universality is slightly insufficient. The third sub-category is SDNE [13] algorithm (structural deep network embedding), which combines the first an d second order proximity to retain the local and global information of the network structure at the same time. The common learning methods of network embedding based on network structure are shown in TABLE 1.

TABLE 1. Applicable network and experimental data set of algorithm.

Algorithm	Network properties			Experimental data set		
	directed	undirected	Weighted	Cora	PPI	Flickr
GreRep		√	√			
HOPE	√	√				
DeepWalk		√		√		√
Node2vec	√	√	√		√	
LINE	√	√	√			√
SDNE	√	√				√

The second category is network embedding combined with external information. In network data, nodes contain not only label information, but also some text information as a supplement to network information. MMDW [14] algorithm (Max Margin DeepWalk), which learns embedding of network nodes in some tags. Cheng and others proposed the TADW algorithm [15] (text associated deep walk), which introduced the text information of nodes into network embedding based on matrix decomposition. Tu et al put forward a new CENE [16] algorithm (a general framework for content enhanced network embedding), which uses the text information of network nodes to explain the relationship between nodes and learn context dependent network representation for network nodes according to different neighbors. The common learning algorithm of homogeneous information network representation combined with external information is shown in TABLE 2.

TABLE 2. Applicable network and experimental data set of algorithm.

Algorithm	Network properties				Experimental data set		
	dire cted	undirect ed	Node text	Node label	Cora	Wiki	DBL P
MMDW	√			√	√	√	
TADW	√		√		√	√	
CENE			√				√

Considering the high sparsity in the real network, in order to better integrate the characteristics of the network structure, we propose the SINE algorithm through fusing the second-order proximity to reconstruct the network to compensate for the sparsity of the network. Then the existing network embedding algorithm is used to obtain the vector representation of the node. In order to show the performance of SINE, input the real information network reconstructed by SINE algorithm into the existing network embedding algorithm to get the vector representation of nodes. Node vector representation as feature are applied to clustering tasks, according to the three kinds of clustering measure assess the effectiveness of learning embedded. Compared SINE with SDNE, LINE and Node2vc algorithm, the clustering effect is obtained when the number of clusters m is different under different embedding dimension d, It is shown that the going algorithm is applied to clustering task overall better clustering results.

II. METHOD

This section will introduce the method for implementing the SINE algorithm. Firstly, the second-order proximity of the network is calculated and the reconstructed network is obtained. Secondly, the vector representation of nodes is obtained by using the network embedding algorithm. Finally, the clustering measure is introduced. The specific process of the SINE algorithm is to input the original network graph and reconstruct the original network to fuse the features of the first-order and second-order nodes in the network by two second-order proximity calculation methods. The vector representation of each node in the network is obtained from the reconstructed network, and the input of k-Means clustering algorithm is node feature and then the clustering effect diagram is draw Fig 2.

Let $G = (V, E)$ denote the undirected, weighted network under consideration with vertex $V = \{v_i,\ i = 1, \cdots, N\}$ and edge set E. Let $A = \left(\omega_{ij}\right)_{N\times N}$ denote the adjacency matrix of the network. ω_{ij} represents the edge weight from nodes v_i and v_j. If $(v_k, v_i) \in E$ and $(v_i, v_j) \in E$, v_k and v_j have second-order neighborhood relationship. If there is no node connected to nodes v_k and v_j at the same time, then second-order proximity of the two nodes is 0.

A. SECOND-ORDER PROXIMITY CALCULATION

The observed first-order proximity in the real world data is not sufficient for preserving the global network structures. As a complement, we explore the second-order proximity between the vector, which is not determined through the

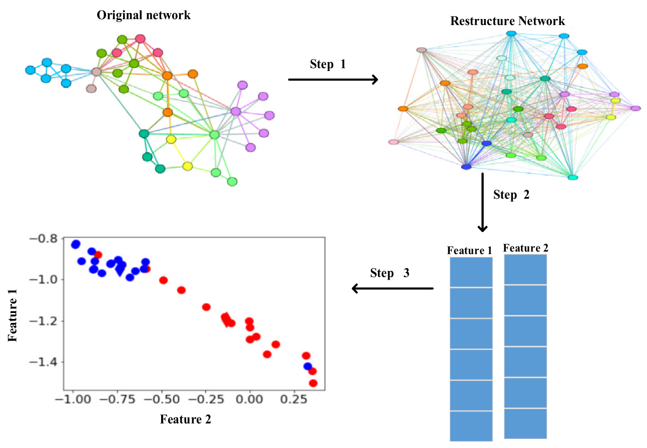

FIGURE 2. Framework of SINE algorithm.

observed tie strength but through the shared neighborhood structures of the vector. The general notion of the second-order proximity can be interpreted as nodes with shared neighbors being likely to be similar. Two nodes have a high second-order proximity and therefore should also be represented closely to each other. We expect that the consideration of the second-order proximity effectively complements the sparsity of the first-order proximity and better preserves the global structure of the network.

SINE algorithm expands the adjacency of a node by adding higher-order nodes. Such as, neighbors and neighbors. In this paper, we only consider adding second-order neighbors, that is, neighbors of neighbors, to each vertex. Two methods are proposed to calculate the second-order proximity between nodes as following:

a. Second-order proximity calculation (denoted by SINE-1)

The weight between vertex v_k and its second-order neighbor v_j is measured as:

$$\omega_{kj} = \sum_{i \in N(K)} \frac{\omega_{ki} \cdot \omega_{ij}}{\omega_{ki} + \omega_{ij}} \tag{1}$$

where $N(K)$ represents the neighbor set of node $v_k \cdot \omega_{ki} \cdot \omega_{ij}$ represents the product of node v_k and node v_j and the weight of two nodes' common neighbor v_i. $\omega_{ki} + \omega_{ij}$ represents the sum of nodes iv weights. According to the formula, adding a new second-order weight to the original network to complete the network reconstruction.

b. Based on Topological overlap matrix (TOM) [17] method (denoted by SINE-TOM)

$$T_{\omega_{\mathrm{kj}}} = \frac{\sum \omega_{ki}\omega_{ij} + \omega_{kj}}{\min\left\{m_k, m_j\right\} + 1} \tag{2}$$

where:

$$m_j = \sum\nolimits_{i \neq j} \omega_{ji} \quad m_k = \sum\nolimits_{i \neq k} \omega_{ki}$$

$\sum \omega_{ki}\omega_{ij} + \omega_{kj}$ represents the value of the adjacency matrix associated by node v_k with node v_j through any node.

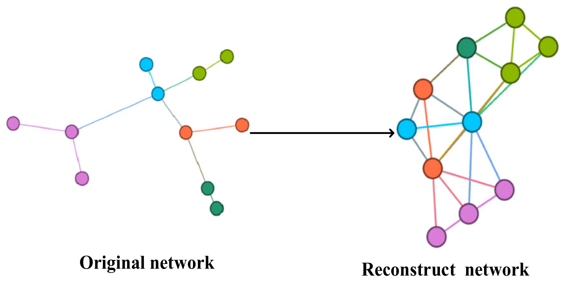

FIGURE 3. Network flow chart of SINE algorithm reconstruction.

To compare the performance of different second-order proximity, another method is proposed to measure whether there is second-order proximity between two unconnected nodes through constructing TOM matrix to measure the correlation. In the network, the correlation of any two nodes is not only determined by whether they are directly connected or not, but also includes the node v_k in the TOM matrix value of the node v_i and the node v_j through the connection correlation of the node v_k to the node v_j.

We expect that the consideration of the second-order proximity effectively complements the sparsity of the first-order proximity and better preserves the global structure of the network. Therefore, we propose two different second-order proximity measures to reconstruct network Fig 3. Firstly, SINE-1 finds the nodes with the second-order neighborhood by traversing the network, then filter out the common neighbors. Secondly, according to the formula calculate the new weight of the second-order nodes. Finally, update the adjacency matrix. The time complexity of the SINE-1 method is large. In order to improve the operation efficiency, we propose SINE-TOM method, which is base on matrix operation. SINE-TOM method can improve the calculation efficiency.

B. NETWORK NODES LOW-DIMENSIONAL EMBEDDING

In the content of section 2.1, our work is mainly to reconstruct the original network with SINE. This section will introduce the use of the reconstructed network into the existing network embedding algorithm to obtain the low-dimensional vector representation of nodes. The following is a brief introduce the network embedding algorithm for comparison in this paper, such as LINE, SDNE and Node2vec.

LINE define the second-order proximity, which complements the first-order proximity and preserves the network structure. LINE investigate both first-order and second-order proximity for network embedding. In the process of training, the first-order and second-order proximity are reserved respectively, and then the embedding vectors trained for each vertex by two methods are connected. In order to fuse the information of first-order and second-order proximity directly, we propose the SINE method to reconstruct the original network directly.

SDNE algorithm uses deep network to model the nonlinear between node representations. The whole model can be divided into two parts: The module of modeling the first level proximity supervised by Laplace matrix on the one hand, on the other hand, the unsupervised deep self encoder models the second level proximity relationship. Finally, SDNE algorithm takes the deep self encoder middle layer as the network embedding of nodes.

Node2vec algorithm's contribution is define a flexible notion of a node's network neighborhood. By choosing an appropriate notion of a neighborhood, Node2vec can learn representations that organize nodes based on their network roles and/or communities they belong to. Achieving this by developing a family of biased random walks, which efficiently explore diverse neighborhoods of a given node. The resulting algorithm is flexible, giving us control over the search space through two adjustable parameters, in contrast to rigid search procedures in prior work and in node classification task have significantly improves.

C. EVALUATION MEASURE

In this section, taking the low dimensional vector representation of nodes as the feature input clustering algorithm, the following evaluation indexes are selected to evaluate the quality of clustering results:

1) DAVIES-BOULDIN INDEX (DBI)

DBI, also known as classification accuracy index, is a kind of index to evaluate the advantages and disadvantages of clustering algorithm [18]. DBI is the maximum value of the ratio of the sum of the average distance within two classes to the center of mass distance. Molecules σ_i and σ_j represent the sum of the average distances from all points in any two clusters to the center of mass of the cluster.

$$DBI = \frac{1}{N}\sum_{i=1}^{N}\max_{j\neq i}\left(\frac{\overline{\sigma_i+\sigma_j}}{d\left(c_i,c_j\right)}\right) \tag{3}$$

N is the number of categories. $d\left(c_i, c_j\right)$ is the sum of the distance between the center points of category i and category j. If the distance within a class is smaller, the distance between classes is larger. Then DBI index will be smaller. In the clustering experiment, if the clustering results are low proximity between clusters and high proximity within clusters, then the clustering results will better.

2) SILHOUETTE COEFFICIENT (SC)

SC index is an evaluation method of clustering effect. It was first proposed by Peter [19] in 1986. The contour coefficient is applicable to cases where the actual category information is unknown. For a single node v_i, let a is the average distance from other nodes in the same category, and b is the average distance from nodes in different categories closest to it. The Silhouette Coefficient is defined as:

$$s = \frac{b-a}{\max(a,b)} \tag{4}$$

The value range of the Silhouette Coefficient is $[-1, 1]$. The more similar node are closer, the more different classes node are father, then Silhouette Coefficient is higher.

3) CALINSKI-HARABAZ INDEX (CH)

CH index [20] is also called variance ratio standard, which is defined as the ratio of the average value of dispersion in clusters to the dispersion between clusters. In the case of unknown real group label, CH value is used as an index of evaluation model.

Calculating the square sum of the distance between each point and the center of the class to measure the compactness within the class and the square sum of the distances between various kinds of center points and data set center points to measure the separation degree of data set. The CH index is obtained by the ratio of separation degree and compactness.

$$s\left(k\right) = \frac{T_r\left(B_K\right)}{T_r\left(w_k\right)} \times \frac{N-m}{k-1} \tag{5}$$

N is the number of samples in the training set, m is the number of categories, B_k is the covariance matrix between categories, w_k is the covariance matrix of data within categories, and T_r is the trace of the matrix.

The denominator w_k indicates that the covariance of the data within the category as smaller as better, the numerator B_k indicates that the covariance between the categories as larger as better. Therefore, CH value as higher as better.

III. EXPERIMENTAL RESULTS

A. DATA SOURCE

In order to evaluate the performance of SINE, we apply SINE to three real networks-the social network data set karate [21] (karate club network), the Dolphins network [22] and the football network [23]. Karate network is a social network constructed by observing a Karate club in an American university. The network consists of 34 nodes and 78 edges, and the number of community division is 2. The Dolphins network, which is an undirected network with 62 nodes and 159 edges. The nodes represent dolphins, while the edges represent frequent contact between dolphins. The Dolphins network is divided into two communities. The Football network consists of 115 nodes and 616 edges, the node in the network represents the football team, and the edge in the network represents a game between two teams. 115 college student teams are divided into 12 leagues, which can be expressed as the real community structure of the network. Refer to TABLE3 for data details.

B. NETWORK RECONSTRUCTION

Based on SINE-1 and SINE-TOM, we reconstructed the real network by calculating the second-order proximity of the vectors in the network Fig 4. From figure, it is shown that the nodes are arranged more closely, the network edges are obviously dense, and the community division results are more obvious after the reconstruction of the network by SINE.

C. CLUSTERING TASK

In this section, we make an experimental analysis of SINE algorithm and evaluate it with clustering algorithm. We compare SINE with several existing network representation

TABLE 3. Characteristics of three network data sets.

The original network	Number of nodes	Number of edges	Real community number	Restructured number of edges
karate	34	78	2	426
Dolphins	62	159	2	2919
football	115	613	12	607

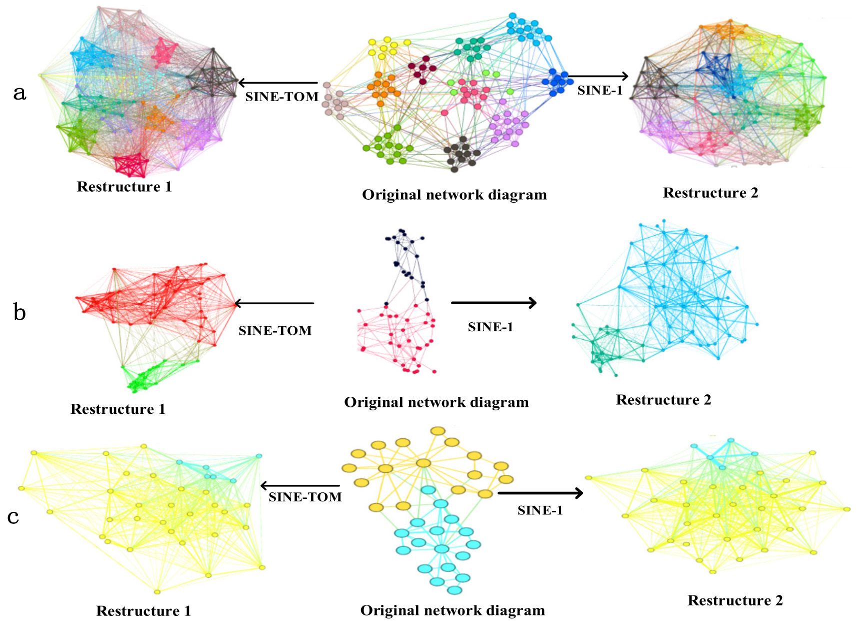

FIGURE 4. Reconstruction Network based on SINE algorithm. (a), (b) and (c) describe the original and reconstructed networks of football, dolphins and karate respectively.

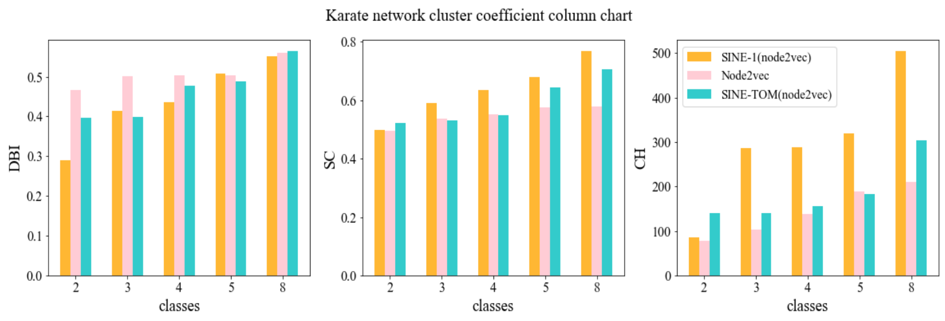

FIGURE 5. Histogram of cluster evaluation coefficient of karate network.

learning methods, Node2vec, LINE and SDNE. Firstly, we input the reconstructed real network into the existing network embedding algorithm. Secondly, use the multidimensional feature representation of nodes as the input parameters to cluster with k-Means method. Finally, the experimental results are analyzed according to the evaluation index of k-means algorithm. Due to the length of the thesis, we only give the karate network column chart in this paper, and other experimental network will be analyzed in supplementary materials (Supplementary materials 1).

The clustering effect of the reconstructed karate network brought into the node2vec algorithm (parameters are set as $p = 0.25$, $q = 0.25$), and compares it with the clustering effect of the original network brought into the node2vec algorithm to make a histogram Fig 5. In the experiment, the dimensions of vector is 2, and the number of clusters is 2, 3, 4, 5, 8 respectively. According to the three clustering evaluation indexes in section 2.3, we can clearly see that for the three clustering evaluation coefficients CH, DBI, SC, SINE is better than Node2vec algorithm.

The original karate network is brought into the LINE network embedding, and the reconstructed karate network is brought into the Node2vec (parameters are set as $p = 0.25$, $q = 0.25$) to obtain the node feature representation,

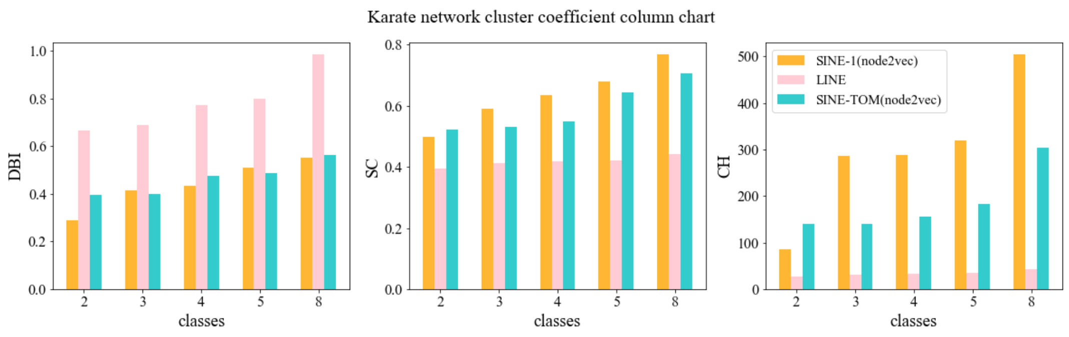

FIGURE 6. Comparison of clustering evaluation coefficients.

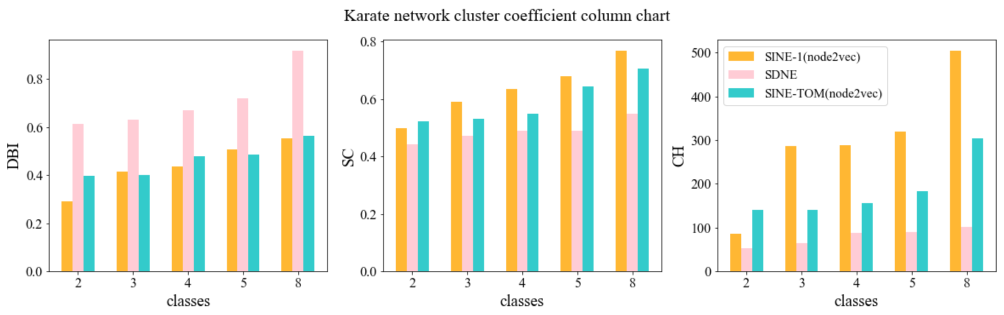

FIGURE 7. Comparison of SDNE clustering effect.

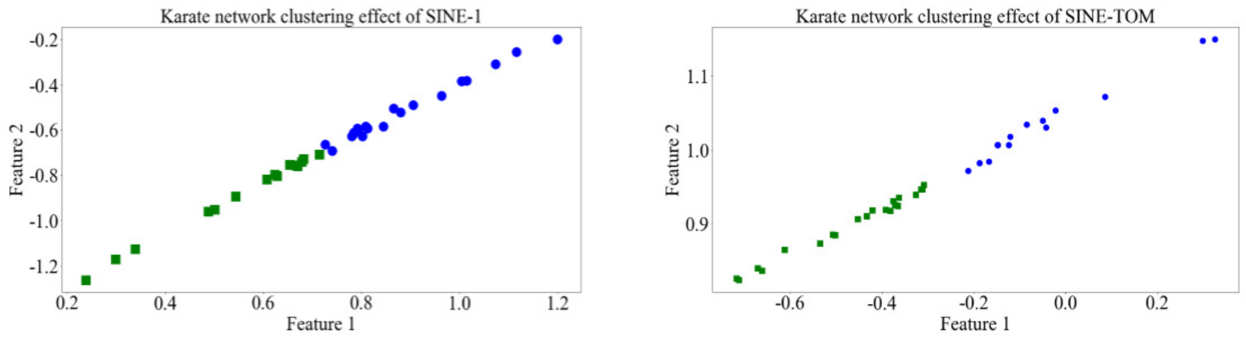

FIGURE 8. SINE clustering effect chart.

which is respectively applied to the clustering task. The comparison chart of clustering evaluation coefficient obtained Fig 6. LINE also considers the first-order proximity and second-order proximity of network nodes, but they are trained separately. From the three clustering evaluation coefficients, the effect of using SINE to reconstruct the network and then bringing into Node2vec to get the characteristic representation of nodes is better than that of LINE.

The original karate network is brought into Node2vec, the reconstructed karate network is brought into SDNE, and cluster the nodes based on these feature Fig 7. Comparing two clustering results, we can see clearly that the clustering effect of the node obtained by SINE algorithm is better than SDNE.

The original karate network is brought into Node2vec, the reconstructed karate network is brought into SDNE, and cluster the nodes based on these feature Fig 7. Comparing two clustering results, we can see clearly that the clustering effect of the node obtained by SINE algorithm is better than SDNE.

In order to visually observe the effect of the learned node vector on clustering, we reconstruct karate network bring into Node2vec to obtain two features representation of the nodes which are input k-means clustering to get the karate clustering rendering Fig 8. It is shown that Two types of nodes are obviously clustered into two types.

The points represent the nodes, and the horizontal and vertical coordinates represent the two-dimensional features respectively.

To show the impact of different embedded dimensions on the results, three experimental networks are input into the existing network representation learning method with different embedded dimensions, test the clustering effect of different network learning algorithms under the real community clustering numbers Fig 9. Parameter setting $d = 2, 4, 6, 8$. Node2vec: $p = 0.25$, $q = 0.25$, the cluster numbers of karate, dolphin and football are 2, 2 and 12 respectively.

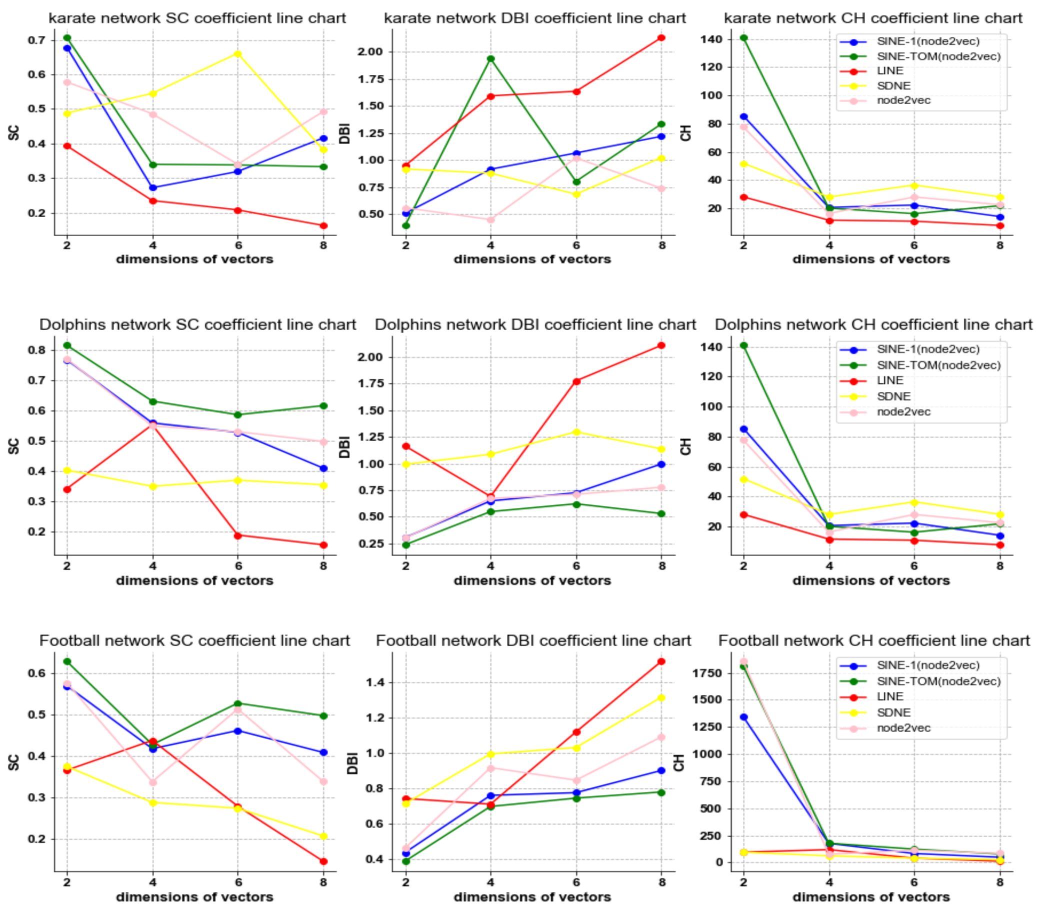

FIGURE 9. Clustering effect comparison with different embedded dimension.

TABLE 4. Comparison table of network clustering methods.

	Karate (m = 2)			Football(m = 12)			Dolphins (m = 2)		
methods	CH	DBI	SC	CH	DBI	SC	CH	DBI	SC
SINE(Node2vec)	20.610	0.915	0.272	176.579	0.759	0.417	**101.902**	0.649	0.560
SINE - TOM (Node2vec)	20.129	1.937	0.340	**177.711**	**0.696**	**0.447**	51.696	**0.549**	**0.632**
LINE	11.614	1.593	0.235	118.923	0.709	0.437	94.064	0.691	0.554
Node2vec	**28.035**	**0.451**	0.486	76.831	0.916	0.337	38.170	0.673	0.550
SDNE	16.289	0.879	**0.545**	61.931	0.994	0.287	95.130	1.086	0.350

Changing the dimension of node embedding vector, SINE algorithm has better performance than other methods for karate network when $d = 2$, while better in football and dolphin networks when $d = 4$.

In the above of clustering experiments, we set the dimension of network embedding as $d = 2$, and we can also get different experimental results by changing the dimension of network embedding to compare with other algorithms. Set $d = 4$, the number of karate and football network data clusters is 2 and football network data clusters is 12, the results is shown in TABLE 4.

In order to evaluate the effect of different node embedded dimension (d) on different network embedding algorithms applied to clustering tasks, we experimented with three data sets: karate, football and dolphins networks to change the network embedding dimension. In karate network, SINE-1 and SINE-TOM have the same performance, which was better than LINE and SDNE network embedding algorithm. In a word, for football network, the clustering effect of SINE algorithm is the best. For dolphin network, the clustering effect of SINE-TOM is better than other algorithms.

The experimental results show that the network embedding algorithm represents nodes as low-dimensional dense vectors in different dimensions, and the performance of clustering tasks also depends on the dimension of network embedding.

IV. DISCUSSION

In order to better integrate network structure for vector representation of nodes, we propose SINE algorithm. Firstly, the algorithm reconstructs the original network into combining the first-order proximity and the second-order proximity network, to preserves both the local and global network structures. Secondly, the reconstructed network graph is input into the existing network embedding algorithm to get the low-dimensional vector representation of network nodes. Finally, the input of k-Means clustering algorithm is node feature. The experimental results show that when $d = 2$, the clustering effect of SINE algorithm is better than other network embedding algorithms. It shows that use SINE algorithm to restructure network, apply to large-scale network compression dimension to retain the structural characteristics of the network as much as possible. However, from the experimental value of SINE-1 and SINE-TOM, the effect is slightly different. It may be that the two methods proposed to calculate the weights between the second-order proximity nodes are different, so the calculated weights are different.

This paper presented a novel network embedding method called the SINE. It proposes two methods to reconstruct the network that preserve both the first-order and second-order proximity, which are complementary to each other. In the future, we plan to investigate the application of this method in large scale network and the suitability of learning algorithms for different types of network embedding. We plan to study the application of SINE in large-scale network data, and complete the reconstruction of large-scale different types of network data with short time complexity.

REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[2] T. Kipf, M. Welling, "Semi-supervised classification with graph convolutional networks," Presented at the Int. Conf. Learn. Represent. [Online]. Available: http://openreview.net/pdf?id=SJU4ayYgl

[3] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node classification in social networks," in *Computer Science*. 2011, pp. 115–148.

[4] T. Y. Berger-Wolf, C. Tantipathananandh, and D. Kempe, "Community identification in dynamic social networks," Presented at the 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, San Jose, CA, USA, Aug. 2007.

[5] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 58, no. 7, pp. 1019–1031, May 2007.

[6] L. Maaten and G. Hinton, "Visualizing data using t-SNE," in *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[7] B.-Y. Sun, X.-M. Zhang, J. Li, and X.-M. Mao, "Feature fusion using locally linear embedding for classification," *IEEE Trans. Neural Netw.*, vol. 21, no. 1, pp. 163–168, Jan. 2010.

[8] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 585–591.

[9] S. C. , W. Lu, and Q. Xu, "Grarep: Learning graph representations with global structural information," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 891–900, doi: 10.1145/2806416.2806512.

[10] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," 2014, *arXiv:1403.6652*. [Online]. Available: http://arxiv.org/abs/1403.6652

[11] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," 2015, *arXiv:1503.03578*. [Online]. Available: http://arxiv.org/abs/1503.03578

[12] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," 2016, *arXiv:1607.00653*. [Online]. Available: http://arxiv.org/abs/1607.00653

[13] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1225–1234, doi: 10.1145/2939672.2939753.

[14] T. Cunchao, W. Zhang, Z. Liu, and M. Sun, "Max-margin deepwalk: Discriminative learning of network representation," Tech. Rep., doi: 10.5555/3061053.3061163.

[15] G. Sun and X. Zhang, "Graph embedding with rich information through heterogeneous network," 2017, *arXiv:1710.06879*. [Online]. Available: http://arxiv.org/abs/1710.06879

[16] C. Tu, X. Zeng, H. Wang, Z. Zhang, Z. Liu, M. Sun, B. Zhang, and L. Lin, "A unified framework for community detection and network representation learning," 2016, *arXiv:1611.06645*. [Online]. Available: http://arxiv.org/abs/1611.06645

[17] A. M. Yip and S. Horvath, "The generalized topological overlap matrix for detecting modules in gene networks," presented at the Int. Conf. Bioinf. Comput. Biol., Las Vegas, NV, USA, Jun. 2006.

[18] J. C. Rojas Thomas, M. S. Penas, and M. Mora, "New version of davies-bouldin index for clustering validation based on cylindrical distance," in *Proc. 32nd Int. Conf. Chilean Comput. Sci. Soc. (SCCC)*, Nov. 2013, pp. 49–53, doi: 10.1109/sccc.2013.29.

[19] I. B. Konovalov, M. Beekmann, A. Richter, J. P. Burrows, and A. Hilboll, "Multi-annual changes of NOx emissions in megacity regions: Nonlinear trend analysis of satellite measurement based estimates," *Atmos. Chem. Phys. Discuss.*, vol. 10, no. 4, pp. 10925–10968, Apr. 2010.

[20] L. I. Li, Y. Liu, Y. Yang, and S. Han, "Short-term wind speed forecasting based on CFD pre-calculated flow fieldss," *Proc. Chin. Soc. Elect. Eng.*, vol. 33, no. 7, pp. 27–32, 2013.

[21] K. Zhou, A. Martin, and Q. Pan, "Evidential communities for complex networks," 2015, *arXiv:1501.01780*. [Online]. Available: http://arxiv.org/abs/1501.01780

[22] D. Lusseau, "The emergent properties of a dolphin social network," *Proc. Roy. Soc. London. B, Biol. Sci.*, vol. 270, no. 2, pp. 186–188, Nov. 2003.

[23] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.

ZIQI WANG is currently pursuing the master's degree with the Qingdao University of Technology. Her current research interests include machine learning and network embedding.

YUANYUAN ZHANG received the B.S. and M.S. degrees from the Shandong University of Science and Technology, in 2008 and 2011, respectively, and the Ph.D. degree from Xidian University, Xi'an, China, in 2016. She is currently an Associate Professor at the School of Information and Control Engineering, Qingdao University of Technology. Her research interests include computational bioinformatics, complex networks, and network representation learning.

SHUDONG WANG received the Graduation degree from the Huazhong University of Science and Technology, Wuhan, in 2004. She is currently a Professor at the China University of Petroleum, Qingdao, China. Her current research interests include biological computing and software engineering.

JUNLIANG SHANG (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Xidian University, Xi'an, China, in 2007, 2010, and 2013, respectively. He is currently an Associate Professor at the School of Information Science and Engineering, Qufu Normal University, Rizhao, China. His research interests include bioinformatics and big data mining.

• • •