Local Structure and High-Order Feature Preserved Network Embedding Based on Non-negative Matrix Factorization

QIN TIAN¹, LIN PAN¹, XUAN GUO², XIAOMING LI³, WEI YU³, FAMING LI⁴

¹School of Marine Science and Technology, Tianjin University, Tianjin, 300072, P. R. China
²College of intelligence and computing, Tianjin University, Tianjin, 300350, P. R. China
³School of International Business, Zhejiang Yuexiu University, Shaoxing, 312069, P.R. China
⁴Research Institute for Chemical Defense, Beijing 102205, P.R. China

Corresponding author: Lin Pan (e-mail: linpan@tju.edu.cn); Faming Li (e-mail: lifamingming@126.com).

This work was supported by the National Natural Science Foundation of China (61902278).

ABSTRACT  Network embedding, as an effective method of learning the low-dimensional representations of nodes, has been widely applied to various complex network analysis tasks, such as node classification, community detection, link prediction and evolution analysis. The existing embedding methods usually focus on the local structure of the network by capturing community structure, first-order or second-order proximity, etc. Some methods have been proposed to model the high-order proximity of networks to capture more effective information. However, they are incapable of preserving the similarity among nodes that are not very close to each other in network but have similar structures. For instance, the nodes with similar local topology structure should be similar in embedding space even if they are not in the same community. Herein, we regard these structure characteristics as the high-order features, which reveals that the structure similarity between nodes is spatially unrelated. In light of above the limitations of existing methods, we construct the high-order feature matrix for mutually reinforcing the embedding which preserves the local structure. To integrate these features effectively, we propose LHO-NMF, which fuses the high-order features into non-negative matrix factorization framework while capturing the local structure. The proposed LHO-NMF could effectively learn the node representations via preserving the local structure and high-order feature information. In specific, the high-order features are learned based on random walk algorithm. The experimental results show that the proposed LHO-NMF method is very effective and outperforms other state-of-the-art methods among multiple downstream tasks.

INDEX TERMS  Network embedding, Local structure, High-order feature, Non-negative matrix factorization

I. INTRODUCTION

Large-scale networks with complex topology structure, node content, node labels and other side information are becoming ubiquitous, such as social networks, biological networks, protein interaction networks, citation networks and telecommunication networks, etc [1]–[6]. In recent years, complex network analysis has attracted increasing attention. Network embedding is a fundamental method in complex network analysis, which refers to the approach of embedding the nodes and links in the network into a low-dimensional space while preserving internal structure and attribute affinity [7]. Similar nodes in the network are also similar in lower-dimensional space and should be kept closer. To this way, a variety of network analysis tasks such as node classification [8], [9], community detection [10]–[12], visualization [13] and link prediction [14], [15], could be well supported by vector-based machine learning algorithms.

Recently, many network embedding methods mainly pay attention to neighborhood structure, community structure, as well as high order proximity, all of are typical network structure. By treating nodes as words and generating short random walks as sentences, DeepWalk [7] based on random walk applies a natural language processing model to learn the representations of nodes preserving the neighbor structure. To enable capturing the diversity of connectivity patterns of the network, Node2vec [16] improves the random walk...
strategy utilizing breadth-first sampling and depth-first sampling to preserve the neighbor structures of different levels. And LGSFA [17] proposed by Hong Huang et al. captures the local geometric structure by reconstructing current nodes with neighbors and enhancing intra-class compactness and inter-class separability. In addition, to improve the classification performance, LNSPE [18] captures the neighbors of samples and defines adjustable loss function to learn the optimal adjacency graph.

Furthermore, it is necessary to capture more abundant neighbor structures to learn more connotative and discriminative node representations. At present, a bunch of methods have been proposed to preserve the first-order, second-order even higher-order node proximity of the network. DeepWalk learns the high-order proximity of nodes by a hyper-parameter, i.e., the window size. LINE [19] integrates the first-order and second-order proximity between nodes into the embedding learning phase. Nevertheless, LINE defines two loss functions so that it is only able to capture the local and global network structure information, respectively. Then, SDNE [20] preserves the local and global structure simultaneously using deep learning algorithm. And it maps the network to a non-linear latent space to capture the highly non-linear network structure. Besides, to capture more global structure information, a number of methods are proposed to capture the higher-order proximity of nodes [20], [21]. The methods mentioned above can merely preserve fixed-order proximity, which can not perform well on all networks or tasks. Hence, Zhang et al. [22] propose AROPE based SVD framework. AROPE obtains arbitrary order proximity of nodes, meanwhile revealing the intrinsic relationship between proximities of different orders.

The above methods have achieved excellent performance on a variety of network analysis tasks. Even so, they are incapable of capturing the similarity of among nodes that are not very close to each other in the network but have similar structures. For example, the nodes which are not in the same community may have similar surrounding topology structure. In that case, they should be similar in embedding space too. In this paper, we regard this feature revealing the structure similarity of node spatially unrelated as high-order features. In addition, they also ignore the mutual interaction between local structural similarity and higher-order features. In fact, it’s essential to explore the relationships between them while preserving these characteristics and how they affect the performance of network embedding. The methods based on matrix factorization by decomposing the adjacency matrix to learn node embedding could only capture the local structure of the network. In addition, most networks are very sparse so that node embeddings learned from adjacency matrix factorization will lose a lot of structural information. The algorithms which define a polynomial function of the adjacency matrix to capture high-order proximity make the network denser and increase computational complexity and noise. And it is also not accurate enough because for an unweighted network, the k-order power of adjacency matrices may include plenty of repetitive edges. In light of the above limitations, we devote to capture the local structure and high-order features at the same time and exploit their consensus relationship on how to impact node embedding. One of the main obstacles we face is how to effectively fuse the high-order features and local structure into embedding space so as to learn more informative and discriminative representations for each node.

Towards this goal, we propose LHO-NMF (Local structure and High-Order features preserved network embedding based on Non-negative Matrix Factorization), a novel approach to preserve the high-order features learned from random walk algorithm and local structure proximity for node representations. NMF enables to map multiple components to a low-dimensional space effectively while maintaining internal correlation between the components due to its interpretability and additivity. Therefore, we construct the high-order feature matrix and local structure extraction matrix, then through NMF framework learn the node low-dimensional embedding preserving multiple structures. In specific, the high-order features are captured by factorizing a random matrix, which preserves the global network structure. Eventually, we optimize it via an iterative multiplicative updating algorithm. Comprehensive and extensive experiments on various real networks demonstrate the effectiveness of our proposed model.

In summary, our major contributions are summarized as follows:

- We propose LHO-NMF, an efficient and scalable algorithm for network embedding, which effectively integrates the high-order feature of nodes into a non-negative matrix factorization framework while capturing the local structure.
- We construct the high-order feature matrix based on random walk and adopt the combined NMF framework to permit users to control the loss weight between different structure features.
- We conduct extensive experiments, including multi-label classification, clustering and link prediction on several real-world datasets to evaluate the performance of the model. The results show that our method achieves significant accuracy improvements than baseline methods.

The rest of this article is organized as follows. In Section 2, we introduce related works. In Section 3, we present the proposed framework LHO-NMF in detail. In Section 4, we provide the cogent experiment results and analysis to evaluate our method. Finally, this article is concluded in Section 5.

II. RELATION WORK

In this section, firstly, we give a simple description of the related work about network embedding. Then we formally introduce the related work about non-negative matrix factorization to help to understand our model better.
A. NETWORK EMBEDDING.

Our work aims to learn representations of nodes in the network, which also is called network embedding. Recently, network embedding has been an active area of frontier research [23]–[25]. From the perspective of the research method, network embedding algorithms can be roughly divided into following categories, including random walk, deep learning and matrix factorization [26].

Early works such as IsoMAP [27] and Local Linear Embedding (LLE) [28], decompose the constructed affinity matrix to obtain the eigenvectors as low dimensional representations of the network. With the achievement of random walk in the field of natural language processing, some researchers propose to apply this method on the graph. For example, DeepWalk [7] generates random walk path sequences for each node of the network and treats path sequences as sentences in word2vec [29]. Then it feeds this path sequences into Skip-Gram [30] to learn the embedding of nodes. To make up the arbitrariness, node2vec [16] defines two hyper-parameters to balance depth-first sampling (DFS) and breadth-first sampling (BFS) to capture more comprehensive neighborhoods structure and global information of the network. More recently, Tang et al. [19] define two loss functions to capture first-order proximity and second-order proximity respectively to preserve the local and global structure of the network. Here, the first-order proximity is defined if there is an edge that is directly connected between two nodes and the second-order proximity is defined by shared neighbors. The limitation of LINE is that it learns the local and global information independently. SDNE [20] defines a semi-supervised neural network model to capture jointly the first-order and second-order proximity, which increases the ability of model to capture nonlinear structure. The second-order proximity is learned by the unsupervised part and capture local information by using first-order proximity as a monitoring message. Further, in addition to integrating the first- and second-order jointly via a deep variational model, DVNE [31] maps the network into a Wasserstein space to capture more uncertainties of the nodes.

Recently, Graph Neural Network which encodes nodes by aggregating neighborhood information in spacial or spectral domains have become popular trends of network embedding. The aggregation mechanism of GCN [32] is defined as localized first-order approximation of spectral graph convolutions. GAT [33] firstly applies self-attention mechanism on neighborhood aggregation. Both of GCN and GAT are trained in semi-supervised manner. GAE [34] has the same convolutional architecture with GCN while it is trained by reconstructing the adjacent relationships. GraphSAGE [35] aggregates neighborhood information with sum-pooling, mean-pooling or LSTMs on sampled neighbors and is trained by reconstructing both edges and sampled non-edges.

Moreover, to capture more comprehensive internal structure information, many researchers extend the works by utilizing high-order information [20]–[22], [36]. For example, GraRep [21] defines different loss function to capture k-hop relational information. Then, they combine k-proximity learned from different models to learn the global representations of nodes. As mentioned before, the above methods only preserve fixed-order proximities, which could not perform well on all networks tasks. Hence, Zhang et al. [22] propose AROPE (ARbitrary-Order Proximity preserved Embedding) based SVD framework. AROPE enables to select arbitrary order proximity of nodes according to the network characteristics, meanwhile, reveal the intrinsic relationship between proximities of different orders.

Furthermore, matrix factorization is also applied in network embedding widely. The common methods such as Singular Value Decomposition (SVD) and Principal components analysis (PCA) are used in the series of matrix factorization models [15], [37]. Non-negative Matrix Factorization (NMF) [38], as a variant of the standard matrix factorization, adds non-negative constraints on matrices. A number of studies indicate that NMF has better additivity and interpretability so that it is often used to learn the embedding. However, the matrix factorization is unable to address with higher order recovery problem and WATITF [39] is proposed to overcome this issues. More about network embedding for a recent review of Cui et al. [26].

B. NON-NEGATIVE MATRIX FACTORIZATION.

Given a nonnegative vector \( x \in \mathbb{R}^n_+ \) and \( n \) observations defined as \( x_{i} = 1, 2, ..., n \), we can obtain a data matrix \( X = [x_1, x_2, ..., x_n] \in \mathbb{R}^{m \times n} \). Non-negative Matrix Factorization(NMF) seeks to find a \( m \times d \) basis matrix \( U = [u_1, u_2, ..., u_d] \in \mathbb{R}^{m \times d} \) and a \( d \times n \) coefficient matrix \( V = [v_1, v_2, ..., v_n] \in \mathbb{R}^{d \times n} \), so that their product approximate to the original data matrix \( X \). \( UV \approx X \). Where \( d \ll \min(m, n) \) is the dimension of the latent space or the rank of the underlying data.

As a popular dimension reduction model, NMF has been used in diverse network mining tasks such as community detection, link prediction and network embedding. A number of studies have confirmed the effectiveness of NMF on these tasks [40]–[48]. Recently, PPNMF (Proximity Preserving Non-negative Matrix Factorization) [49] combines the first-order and the second-order proximity into NMF framework to learn node representations. In addition to integrating first-order and second-order similarity, Wang et al. [10] propose a Modularized Non-negative Matrix Factorization (M-NMF) framework to add community constraints through defining a modularity constraint term. Then M-NMF utilizes the NMF to incorporate the similarity information and community structure into embedding space.

Further, in consideration of the significance of higher-order proximity for grasping the global characteristics of the network, some studies take higher-order proximity into account to capture more comprehensive information. GraRep [21] applies matrix factorization to preserve the high-order proximity and concatenate the representations of nodes learned from different orders. To be more effective and flexible, AROPE (ARbitrary-Order Proximity preserved
Embedding) [22] based SVD framework could take the similarity of any order according to the network characteristics while retaining the correlation between different orders.

However, the above NMF-based methods are shallow model so that they are unable to reveal the nonlinearity correlation between the original network and embedding space. In light of these issues, Ye et al. [50] propose the Deep Autoencoder-like NMF (DANMF) based on NMF framework. DANMF combines an encoder and decoder component with NMF to preserve the hidden structure characteristic.

It is obvious that almost all of the above network embedding frameworks only consider the local neighbor structure, community property, low order or high order proximity between nodes independently. How they affect network embedding performance is needed to explore. In our paper, we construct a NMF framework to exploit the consensus relationship between neighbor structure, community attribute and k-order structure information impact on network embedding.

III. THE PROPOSED MODEL

In this section, we present our proposed method LHO-NMF in details. Firstly, we report the notations used in this paper. Then, we introduce how to capture high-order feature and local structure respectively. Finally, we integrate them with NMF to learn the low-dimensional representations of nodes following the optimization algorithm.

LHO-NMF includes two vital parts, a local structure information encoder and a high-order feature encoder. As we know, adjacency matrix \( A \) preserves the most of the network topology structure characteristic and directly represents the first-order proximity. Our model first maps the original network into a low-dimensional embedding space to obtain \( \mathbf{X} \) by factorizing the adjacency matrix \( A \). Because network is very sparse, \( \mathbf{X} \) loses a lot of pivotal information when preserving the local structure. At the same time, we generate random walk sequences for each node and learn the k-order feature by controlling the window size. Further, we integrate the high-order features learned from random walk into NMF framework, capturing the local structure to learn node representations preserving high-order feature information and local structure jointly. The overall architecture of our model is shown in Fig. 1.

A. NOTATIONS

Suppose there is an undirected network \( G = (V, E) \) with \( n \) nodes and \( e \) edges, \( V = [v_1, v_2, ..., v_n] \) denotes the set of nodes and \( E \) denotes the set of edges among the nodes. \( G \) is represented by the adjacency matrix \( A \in \mathbb{R}^{n \times n} \). \( A(i,:) \) represents the connections between the node \( i \) and other nodes in \( G \). For unweighted networks, if there exists an edge between node \( i \) and node \( j \), \( A_{ij} = 1 \), otherwise \( A_{ij} = 0 \). Since network \( G \) is undirected, \( A \) is a symmetric matrix, i.e \( A_{ij} = A_{ji} \). In this paper, we utilize bold uppercase characters to exhibit matrices. The matrix \( B \in \mathbb{R}^{n \times d} \) preserves the high-order structure features of nodes in the network, where \( d \) is the dimension of features. \( B(i,:) \) describes the structure feature of the node \( i \). Our purpose is to learn the representations of nodes \( \mathbf{W} \in \mathbb{R}^{n \times k} \) \((k \leq n)\), where \( k \) is the dimension of representations. The terms and notations used later are shown in Table 1.

| Notation | Definition |
|----------|------------|
| \( G = (V, E) \) | The graph \( G \) with node set \( V \) and edge set \( E \) |
| \( A \in \mathbb{R}^{n \times n} \) | The adjacency matrix of \( G \) |
| \( B \in \mathbb{R}^{n \times d} \) | The high-order structure feature matrix of nodes |
| \( \mathbf{W} \in \mathbb{R}^{n \times m} \) | The representations of nodes |
| \( \mathbf{X} \in \mathbb{R}^{n \times m} \) | The local structure matrix of nodes |
| \( \text{vol}(G) = \sum_{i=1}^{T} \sum_{j=1}^{T} A_{ij} \) | The volume of a graph \( G \) |
| \( \text{size of window} \) | The size of window |

B. LOCAL STRUCTURE EXTRACTION

In the real world, information network often has a lot of information missing, many nodes without direct connection in the adjacency matrix also have high similarity in essence. In addition, the adjacency matrix preserves directly connected edges which characterize the first-order proximity of nodes. They particularly reveal that adjacency matrix preserves the topology structure of the network. Hence, we factorize the adjacency matrix \( A \) to capture local topology structure of the network. To preserve the local structure in a low-dimensional space, we adopt the NMF method, which minimizes the following objective function:

\[
\| \mathbf{A} - \mathbf{X} \mathbf{X}^T \|_F^2 \quad \text{s.t.,} \quad \mathbf{X} \geq 0, \tag{1}
\]

where \( \mathbf{X} \in \mathbb{R}^{n \times m} \) is local feature matrix and \( m \) is the dimension of the space.

As mentioned before, embedding matrix \( \mathbf{X} \) loses a lot of pivotal information actually. Therefore, we integrate higher-order features to mutually enhance the learning of \( \mathbf{X} \). At the same time, we obtain the low dimensional representations of nodes preserving local and community structures by decomposing the local feature matrix \( \mathbf{X} \), which minimizes the following objective function:

\[
\| \mathbf{X} - \mathbf{WU} \|_F^2 \quad \text{s.t.,} \quad \mathbf{X} \geq 0, \mathbf{W} \geq 0, \mathbf{U} \geq 0, \tag{2}
\]

where \( \mathbf{W} \in \mathbb{R}^{n \times k} \) is the embedding matrix, \( \mathbf{U} \in \mathbb{R}^{k \times m} \) is an auxiliary matrix and \( k \) is the embedding dimensionality.

C. HIGH-ORDER STRUCTURE EXTRACTION

In this paper, we extract high-order structure of nodes based on random walk. Qiu et al. [51] prove that the models based on random walk and skip-gram are regarded as factorizing the matrix with a closed form and verify their effectiveness for conventional network mining tasks. The matrices that are implicitly approximated and factorized is denoted as the following:

\[
\mathbf{M} = \log(\text{vol}(G)\left(\frac{1}{T} \sum_{t=1}^{T} (\mathbf{D}^{-1} \mathbf{A})^t (\mathbf{D}^{-1})\right)) - \log(b) \tag{3}
\]
where \( \text{vol}(G) \) denote the volume of a graph \( G \), \( \text{vol}(G) = \sum_i \sum_j A_{i,j} \). \( D \) is the degree matrix of network \( G \), \( D = \text{diag}(d_1, d_2, \ldots, d_{|V|}) \). \( T \) denotes the context window size and \( b \) denotes the number of negative sampling in skip-gram. Then, we perform truncated singular value decomposition (SVD) on the constructed \( M \) to capture the feature matrix \( B \) of node in low-dimensional space, i.e., \( M = BV \). Higher-order structures can be obtained as the window size \( T \) increases. Inspired by this, we extract the high-order structure from the matrix \( B \).

\[
\|WH - B\|_F^2 \quad \text{s.t.,} \quad W \geq 0, H \geq 0, \quad (4)
\]

where \( H \in \mathbb{R}^{t \times d} \) is an auxiliary matrix.

**D. UNITED FRAMEWORK**

Together with the objective function (1), (2), (4), the final loss function of our method can be denoted as:

\[
\mathcal{L} = \min_{X, W, H, U} \| A - XX^T \|_F^2 + \alpha \| X - WU \|_F^2 + \beta \| WH - B \|_F^2 + \gamma (\| U \|_F^2 + \| H \|_F^2) \quad (5)
\]

\[
\text{s.t.,} \quad X \geq 0, W \geq 0, H \geq 0, U \geq 0,
\]

where \( \alpha, \beta \) and \( \gamma \) are positive parameters for adjusting the contribution of the corresponding terms.

**E. OPTIMIZATION**

Because the loss function in (5) is not convex, it is not feasible to use derivation to calculate the optimal solution. In our paper, we divide the loss function into four subproblems to separately optimize four parameter matrices \((X, W, H, U)\). Then we use Majorization-Minimization framework [52] to compute the local optimal solution of each problem. The update strategy we adopted is alternation optimization, i.e., fixing the other three matrices when updating one. Algorithm 1 shows the pseudo code of the optimization process. The specific formulas are shown as the following.

**X-subproblem:** Updating \( X \) with other parameters \( W, U, H \) fixed, denoted as the following suboptimization problem:

\[
\min_X \| A - XX^T \|_F^2 + \alpha \| X - WU \|_F^2 \quad \text{s.t.,} \quad X \geq 0. \quad (6)
\]

Herein, \( X \) is a nonnegative matrix so we introduce the Lagrange multiplier matrix \( \Theta \), which results in the following equivalent function:

\[
L(X) = tr(AA^T - AXX^T - XX^TA^T + XX^TXX^T) + \alpha tr(XX^T - XU^TW^T - WUX^T + WUU^TW^T) + tr(\Theta X^T). \quad (7)
\]

Further, we set the value of (7) to 0, i.e., \( L(X) = 0 \), we have:

\[
\Theta = 4AX - 4XX^TX - 2\alpha X + 2\alpha WU. \quad (8)
\]

Following the Karush-Kuhn-Tucker (KKT) condition for the non-negativity of \( X \), resulting in the following equation:

\[
(4AX - 4XX^TX - 2\alpha X + 2\alpha WU) \cdot X = \Theta X = 0. \quad (9)
\]
Initialize the value of $X$, and the updating rule of $X$ is:

$$X \leftarrow X \odot \left( \frac{2AX + \alpha WU}{2XX'X + \alpha X} \right),$$

(10)

where $\odot$ denotes the matrix multiplication.

**W-subproblem:** When updating $W$ with other parameters $X$, $H$, $U$ fixed, we denote to solve the following objective function:

$$\min_{W} \alpha \|X - WU\|_F^2 + \beta\|WH - B\|_F^2 \ s.t., \ W \geq 0.$$  

(11)

Similar to the optimization computation of $W$, we define the following updating rule for $W$:

$$W \leftarrow W \odot \left( \frac{\alpha XU' + \beta BH'}{\alpha WU'U + \beta WH'H} \right).$$  

(12)

**H-subproblem:** Updating $H$ with other parameters $X$, $W$, $U$ fixed, which results in the following function:

$$\min_{H} \alpha \|WH - B\|_F^2 + \gamma \|H\|_F^2 \ s.t., \ H \geq 0.$$  

(13)

In the same way, the updating rule of $H$ is then given as follows:

$$H \leftarrow H \odot \left( \frac{WB'}{\beta WHW + \gamma H} \right).$$  

(14)

**U-subproblem:** Updating $U$ with other parameters $X$, $W$, $H$ fixed leads to solve the following problem is needed:

$$\min_{U} \beta \|X - WU\|_F^2 + \gamma \|U\|_F^2 \ s.t., \ U \geq 0.$$  

(15)

Likewise, optimize $U$ by the same optimization method as $X$, we use the following rule to update:

$$U \leftarrow U \odot \left( \frac{\alpha W'X}{\alpha W'UW + \gamma U} \right).$$  

(16)

**Algorithm 1:** Optimization of the LHO-NMF

**Input:** The network $G$, the high-order structure features matrix $B$, the embedding dimension $d$, convergence coefficient $\delta$, and balance parameter $\alpha, \beta, \gamma$.

**Output:** $X \in \mathbb{R}^{n \times m}$, $W \in \mathbb{R}^{n \times k}$, $U \in \mathbb{R}^{k \times m}$, $H \in \mathbb{R}^{k \times d}$.

1. Initialize $X$, $W$, $U$, $H$;
2. while not conv do:
3. Update $X$ according to (10);
4. Update $W$ according to (12);
5. Update $H$ according to (14);
6. Update $U$ according to (16);
7. compute loss function $L$ using (5);
8. $i \leftarrow i + 1$;
9. until $\frac{e_i}{e_{i-1}} < \delta$
10. end while
11. return $X$, $W$, $U$, $H$.

The optimization flowchart of our method is described in Algorithm 1. The input data of optimization include the network $G$, the high-order feature matrix $B$, the embedding dimension $k$, convergence coefficient $\delta$, and balance parameter $\alpha, \beta, \gamma$. First of all, randomly initialize $X, U, W, H$ with a uniform distribution. Then, we update $X, U, W, H$ iteratively until convergence (Alg. 1, Line 2 – 10). The output is the embedding matrix $W$ for all nodes in network. The representations of nodes learned by LHO-NMF could preserve both the high-order feature and local structure.

**F. COMPUTATIONAL COMPLEXITY ANALYSIS**

The whole computation complexity of our model depends on the matrix multiplication in the updating rules. As we know, given two matrices $A$ and $B$, where $A \in \mathbb{R}^{m \times r}$ and $B \in \mathbb{R}^{r \times n}$, the computational complexity of $AB$ is $O(mrn)$. Based on this, the computational complexity of four updating equations in Algorithm 1 (Line 6 - 9) is $O(n^2m + nkm + nm^2), O(nmk + nk + 2nk^2 + mk^2 + dk^2), O(nkd + nk^2 + dk^2), O(nkm + nkm + mk^2)$ respectively. Since $m$, $k$, $d$ can be regard as the input constant and $m \leq n$, $k \ll n$, $d \ll n$, the computational complexity is $O(n^2 + nm^2 + nkm + nk)$. In practice, most networks are very sparse, hence only the non-zero values are computed in matrix multiplication. Based on this, the computation is reduced to $O(ne + nm^2 + nkm + nk)$, where $e$ is the edge number in networks. In addition, in our model the matrix $X \in \mathbb{R}^{n \times m}$, $W \in \mathbb{R}^{n \times k}$, $U \in \mathbb{R}^{k \times m}$, $H \in \mathbb{R}^{k \times d}$ are parameter matrix, so the space complexity is $O(nmk + km + nk)$. Because $k$, $m$, and $d$ are smaller than $n$, the space computation is reduced to $O(n)$. We can find that the complexity of our model is the same order of magnitude with most NMF-based algorithms.

In order to effectively illustrate the time complexity of our model, we learn the low-dimensional embeddings of nodes for several different size datasets and compute the time then show them in Table 2. The results show that the scale of the network increases in multiples, and the computing time also increases in multiples.

**IV. EXPERIMENTS**

In this section, we conduct extensive experiments on multi-label node classification, clustering, link prediction and visualization tasks to evaluate the effectiveness of our model. Firstly, we report the datasets and experimental settings used in this paper. To demonstrate the effectiveness of our model, we compare it to those of state-of-the-art methods on extensive experiments. In the end, we analyze its sensitivity across used parameters.

| TABLE 2: Computing time on different datasets. |
|-----------------------------------------------|
| **Datasets** | **PPI** | **Wikipedia** | **BlogCatalog** | **Flickr** |
| **N(nodes)** |      |              |                |           |
| Time          | 0.615s | 0.660s       | 10.012s        | 70.230s   |

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3045532, IEEE Access
TABLE 3: Network Statistics.

| Datasets               | N(nodes) | E(edges)  | K  | Multi-label |
|------------------------|----------|-----------|----|-------------|
| BlogCatalog            | 10,312   | 333,983   | 39 | yes         |
| PPI                    | 3,890    | 76,584    | 50 | yes         |
| Wikipedia              | 4,777    | 184,812   | 40 | yes         |
| Flickr                 | 80,513   | 5,899,882 | 195| yes         |
| Polblog                | 1,490    | 16,627    | 2  | no          |
| Livejournal            | 11,118   | 396,461   | 26 | no          |
| Orkut                  | 998      | 23,050    | 6  | no          |

### A. DATASETS AND EXPERIMENTAL SETTINGS

We employ four widely-used network datasets for multi-label node classification task and three real network with ground truth datasets for clustering and link prediction. The statistical characteristics of these datasets are shown in Table 3 and the detailed introduction is as follows.

- **BlogCatalog** [53]: It is an online blogging network. The nodes represent the bloggers and the edges represent the relationships between bloggers. Labels indicate categories of interest to the blogger. A blogger has multiple interests so a node has multiple labels.
- **Wikipedia** [1]: It is a co-occurrence network of words on Wikipedia. The class labels are the Part-of-Speech (POS) tags inferred by Stanford POS-Tagger [55].
- **Flickr** [53]: It is also an online community platform. User could subscribe to a diversity of interest groups. Class labels represent the interest groups.
- **Polblog** [56]: Polblog is a social network whose nodes represent the American politicians’ blogs and an edge exists if their blogs have web links. Labels indicate categories of politicians.
- **Livejournal** [57]: Livejournal is also an online social network dataset, whose nodes represent the bloggers and an edge exits between two nodes if they are friends. Divide bloggers into groups according to their friendship and treat the groups as labels.
- **Orkut** [57]: Orkut is an online making friends network with nodes as users and they have an edge if they are friends. The groups that organized by users are ground-truth.

### Experimental settings

The parameters of our model LHO-NMF include three hyperparameters $\alpha$, $\beta$, and $\gamma$, the local structure dimension $m$ and embedding dimension $k$. In the experiment, we set $\alpha$, $\beta$, and $\gamma \in [1, 101]$ or $\in [0, 1]$, $m \in [100, 200, 300, 400, 500]$ to find the optimal parameters of the model. And $k$ changes according to the number of labels. The experiment results show that the performance is better when $\alpha$, $\beta$, and $\gamma \in [1, 101]$. In the process of extracting the high-order structure matrix $B$, we set the dimension $d = 128$. When $m$ is greater than 200, the clustering performance gradually decreases, so that we set the dimension $m = 200$. The details of parameter analysis are shown in Fig. 4.

The effects of two parameters on the experimental results are analyzed in the following. In general, we set the value of $\alpha$, $\beta$, and $\gamma$ depending on the data set in the experiment below.

### B. BASELINE METHODS

We compare LHO-NMF with three NMF-based and four state-of-the-art network embedding methods. Their details are listed as follows.

- **M-NMF** [10]: M-NMF incorporates community structure and first two order proximities into NMF framework to learn the embedding of nodes. In experiment, we set the embedding dimension as 128, and utilize the default values in paper for other parameters.
- **NetMF** [51]: NetMF proves that the models with negative sampling such as DeepWalk, LINE, node2vec and PTE are regarded as factorizing the matrix with a closed form and demonstrate proposed model outperforms DeepWalk and LINE for conventional network mining tasks.
- **AROPE** [22]: AROPE based SVD framework shifts embedding vectors across arbitrary orders and reveals the intrinsic relationship between them to learn arbitrary high-order proximity of nodes.
- **DeepWalk** [7]: Deepwalk generates the random walk paths for each nodes and treats these paths of nodes as sentences in language model. Then it uses the skipgram [29] to learn the embedding vectors. In experiment, we set the parameters suggested by the paper.
- **Node2vec** [16]: Node2vec extends the DeepWalk by using a biased random walk. It introduces two bias parameters $p, q$ to optimize the random walk. The all parameters are set by default settings. We adopt the default settings for all parameters.
- **LINE** [19]: LINE preserves the first and second-order proximities separately by defining two loss functions to learn embeddings of nodes. We adopt the default parameter settings except for the negative ratio to 5.
- **SDNE** [20]: SDNE utilizes a deep auto-encoder to optimize first-order and second-order proximity simultaneously. We set the parameters as in the article of the authors.
- **GAE** [34]: GAE is based on the variational autoencoder and has the same convolutional architecture with GCN while it is trained by reconstructing the adjacent relationships.

### C. MULTI-LABEL CLASSIFICATION

In this subsection, we evaluate the performance for multi-label node classification in terms of metrics Micro-F1 and Macro-F1. In order to reduce the occasionality of experimental results, we repeat the classification procedure 10 times and take the average as the results. Table 4 shows the node classification performance of our method and baselines on...
The reason is that Wikipedia is a dense word co-occurrence performance than our method in terms of both Micro-F1 numbers represent the best results.

$T$ variations from $2$ to $10$, in Table 5. In Blogcatalog, PPI and Flickr, as window size $T$ increases, the proposed LHO-NMF achieves significantly better classification performance as measured by both Micro-F1 and Macro-F1. But the performance tends to be stable after $T$ reaches $4$ gradually. In Wiki, when $T$ is greater than $3$, the classification performance gradually degrades. This phenomenon indicates that the size of window is never the bigger, the better, we should set it in terms of the network sparsity dynamically to reduce computation. Our method can dynamically adjust the window size according to network sparsity to learn better node representations.

D. NODE CLUSTERING

In this subsection, we evaluate the performance for node clustering based on typical metrics, Normalized Mutual Information (NMI). We evaluate the clustering performance on real-world datasets with ground-truth that include Polblog, Livejournal and Orkut. NMI varies from $0$ to $1$ and larger value indicates better clustering performance. In our experiment, we apply the standard k-means algorithm to obtain the clustering results of other network embedding methods. Since the initial value has a great influence on the clustering result, we repeat the clustering $10$ times and compute the mean of them as the results.

Table 6 shows the node clustering performance about NMI. Similarly, bold numbers represent the best results in the table. The table results show LHO-NMF achieves the best performance over all network datasets on NMI. Especially, on the Polblog dataset, our method LHO-NMF achieves $24$
percent improvement on NMI compared with the second-best method. This benefits from that our method integrates both the high-order structure, neighbor and community attribute, capturing diverse and comprehensive structure characteristic of networks. DeepWalk and node2vec based on random walk can capture the second even higher-order proximities. However, they ignore the community structure. In addition, SDNE and LINE only preserve the proximity between nodes of network, which is also incapable to effectively preserve community structure. AROPE could capture the similarity of different orders. Although more global structural information is captured as the order increases, module information is still ignored by AROPE. And the M-NMF adds modularity term to learn the embedding of nodes. However, for the networks that are very sparse and community structure of which are not obvious, the modularity constraint of NMF makes the representation of nodes similar to each other, so it shows relatively low in performance. In specific, we show the relative performance of LHO-NMF(X) where X as embedding matrix and LHO-NMF where W as embedding matrix, respectively. The results show that the methods only based on matrix factorization show poor performance on clustering tasks. The above results demonstrate the superior power of fusing high-order features into embedding when preserving local structure.

E. LINK PREDICTION

Given a network that is removed a portion of the edges proportionally, link prediction aims to predict which pairs nodes are likely to form edges. In our experiments, we randomly hide 50%, 40%, 30%, 20%, 10% edges as test data for evaluation while ensuring that the remaining network is connected, and use the remaining edges to train the node embedding vectors respectively. A typical metric Area Under Curve (AUC) score is utilized to evaluate the performance of LHO and other baseline methods.

First, we show the results of removing 10% edges on all network datasets to verify the performance of LHO-NMF. As we can see in Table 7, our model achieves 19.6%, 7.5% and 9.5% improvements on Polblog, Orkut and Livejournal respectively. We notice that M-NMF is next only to our model on all datasets in predictive power. In the same way, we also show the relative performance of LHO-NMF(X) where X is embedding matrix, demonstrating the methods only based on matrix factorization perform poor on link prediction.

In specific, we take the results on Livejournal and Orkut as examples to explore the effect of the ratio of training data. The results in Fig. 3 show that our method achieves significant and consistent superiority than all the baselines in the two datasets under different portions of removing edges. Due to the difference in network structure, some datasets reach the optimal level at 80, while others reach the optimal level at 90. In general, the results demonstrate that our method could achieve superior performance in link prediction, indicating the effectiveness of preserving the high-order feature and local structure information for network embedding.
In this subsection, we analyze the effect of parameters \(\alpha\), \(\beta\), and \(m\) of LHO-NMF for clustering performance on the real networks, where \(m\) is the dimension of local structure embedding space. Experiment results reveal that with the change of parameters the clustering performance presents similar trends on different datasets. For brevity, we use PPI as examples to explore the effects of different parameters. We explore the effect of each parameter by changing two of them and fix the others at the same time. For example, we vary \(\alpha\), \(\beta\) and fix \(\gamma\), \(m\) to observe the effect of \(\alpha\), \(\beta\). And so on in a similar fashion. Specifically, we vary \(m\) from 100, 200, 300, 400, 500. Fig. 4(a) - (c) and Fig. 6 show the performance of NMI with the change of these parameters respectively. In Fig. 4(a), clustering performance is worst when both \(\alpha\) and \(\beta\) are less than 10. Within a certain range, the value of NMI tends to be stable with the increase of \(\alpha\) and \(\beta\). When \(\alpha\) is greater than 50 and \(\beta\) is less than 30, clustering performance achieves best. As we can see in Fig. 4(b), from the horizontal-level, NMI does not vary too much when \(\alpha\) is in [1, 20], which suggests the clustering performance is relatively stable as \(\gamma\) increases when \(\alpha\) is within a certain range. In Fig. 4(c), we notice that within a certain range, NMI tends to be stable when \(\beta\) and \(\gamma\) are linearly correlated, and NMI reaches its maximum when \(\gamma\) and \(\beta\) are in range [20, 101]. Fig. 6 shows the effect of the dimensionality of local structure embedding space \(m\) on three datasets. NMI is maximum when \(m\) is 200, then the NMI decreases as \(m\) increases, which suggests when \(m = 200\), low-dimensional space is sufficient to capture the local structure of the most network.

### F. PARAMETER ANALYSIS

In this subsection, we analyze the effect of parameters \(\alpha\), \(\beta\), \(\gamma\) and \(m\) of LHO-NMF for clustering performance on the real networks, where \(m\) is the dimension of local structure embedding space. Experiment results reveal that with the change of parameters the clustering performance presents similar trends on different datasets. For brevity, we use PPI as examples to explore the effects of different parameters. We explore the effect of each parameter by changing two of them and fix the others at the same time. For example, we vary \(\alpha\), \(\beta\) and fix \(\gamma\), \(m\) to observe the effect of \(\alpha\), \(\beta\). And so on in a similar fashion. Specifically, we vary \(m\) from 100, 200, 300, 400, 500. Fig. 4(a) - (c) and Fig. 6 show the performance of NMI with the change of these parameters respectively. In Fig. 4(a), clustering performance is worst when both \(\alpha\) and \(\beta\) are less than 10. Within a certain range, the value of NMI tends to be stable with the increase of \(\alpha\) and \(\beta\). When \(\alpha\) is greater than 50 and \(\beta\) is less than 30, clustering performance achieves best. As we can see in Fig. 4(b), from the horizontal-level, NMI does not vary too much when \(\alpha\) is in [1, 20], which suggests the clustering performance is relatively stable as \(\gamma\) increases when \(\alpha\) is within a certain range. In Fig. 4(c), we notice that within a certain range, NMI tends to be stable when \(\beta\) and \(\gamma\) are linearly correlated, and NMI reaches its maximum when \(\gamma\) and \(\beta\) are in range [20, 101]. Fig. 6 shows the effect of the dimensionality of local structure embedding space \(m\) on three datasets. NMI is maximum when \(m\) is 200, then the NMI decreases as \(m\) increases, which suggests when \(m = 200\), low-dimensional space is sufficient to capture the local structure of the most network.

### G. VISUALIZATION

In order to show the great performance of our model more intuitively, we show the embeddings generated by our method and baselines on Orkut graph. We map the node representations learned from the models into a two-dimensional space, where nodes with the same color have the same label. The closer the nodes with the same color are and the farther away the nodes with different colors are, the better effectiveness the models have. Here, the label is the community ground truth.

The detailed results of visualization are exhibited in Fig. 5. We have the following observations and insights:

- It is intuitive that our proposed model achieves a significantly better clustering performance. The observation implies that preserving high-order proximity, neighbor structure and community structure could learn more distinguish representations for nodes.

- AROPE captures the different order similarity of nodes and more information can be captured as the order increases. However, the module information is ignored, which results in representations of nodes are too similar to distinguish for Orkut network with close relationship between the different community.

### V. CONCLUSION

In this paper, we propose an efficient scalable algorithm LHO-NMF to learn low-dimensional node representations. To mutually enhance the embedding learning of nodes, LHO-NMF effectively integrates the high-order features and local...
structure into non-negative matrix factorization framework. Specifically, high-order features are captured based on random walk. We further introduce hyper-parameters for control the loss weight between different structure features. Our extensive experiments including multi-label classification, clustering and link prediction on several real-world datasets demonstrate the effectiveness of our model for network embedding. But if the network has a million or a hundred million nodes the efficiency may be reduced because the model is based on matrix factorization.

In the future, we will focus on designing better algorithm for different structure preserving and improve computational efficiency while maintaining high accuracy.

REFERENCES

[1] T. Kuhn, M. Perc, and D. Helbing, “Inheritance patterns in citation networks reveal scientific memes,” Physical Review X, vol. 4, no. 4, p. 041036, 2014.
[2] T. Nepusz, H. Yu, and A. Paccanaro, “Detecting overlapping protein complexes in protein-protein interaction networks,” Nature methods, vol. 9, no. 5, p. 471, 2012.
[3] J. Kim and M. Hastak, “Social network analysis: Characteristics of online social networks after a disaster,” International Journal of Information Management, vol. 38, no. 1, pp. 86–96, 2018.
FIGURE 6: The effect of the dimensionality of local structure embedding space $m$.

4. D. M. Camacho, K. M. Collins, R. K. Powers, J. C. Costello, and J. J. Collins, “Next-generation machine learning for biological networks,” Cell, vol. 173, no. 7, pp. 1581–1592, 2018.

5. H.-J. Li and L. Wang, “Multi-scale asynchronous belief percolation model on multiplex networks,” New Journal of Physics, vol. 21, no. 1, p. 015005, 2019.

6. H.-J. Li, Q. Wang, S. Liu, and J. Hu, “Exploring the trust management mechanism in self-organizing complex network based on game theory,” Physica A: Statistical Mechanics and its Applications, vol. 542, p. 123514, 2020.

7. B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2014, pp. 701–710.

8. P. Sen, G. Namata, M. Bilgic, L. Getoor, B. galligher, and T. Eliassi-Rad, “Collective classification in network data,” AI magazine, vol. 29, no. 3, pp. 93–93, 2008.

9. S. Bhagat, G. Cormode, and S. Muthukrishnan, “Node classification in social networks,” in Social network data analytics. Springer, 2011, pp. 115–148.

10. X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, “Community preserving network embedding,” in Thirty-First AAAI Conference on Artificial Intelligence, 2017.

11. H.-J. Li, Z. Bu, Z. Wang, and J. Cao, “Dynamical clustering in electronic commerce systems via optimization and leadership expansion,” IEEE Transactions on Industrial Informatics, vol. 16, no. 8, pp. 5327–5334, 2019.

12. H.-J. Li, L. Wang, Y. Zhang, and M. Perc, “Optimization of identifiability for efficient community detection,” New Journal of Physics, vol. 22, no. 6, p. 063035, 2020.

13. I. Herman, G. Melançon, and M. S. Marshall, “Graph visualization and navigation in information visualization: A survey,” IEEE Transactions on visualization and computer graphics, vol. 6, no. 1, pp. 24–43, 2000.

14. D. Liben-Nowell and J. Kleinberg, “The link-prediction problem for social networks,” Journal of the American society for information science and technology, vol. 58, no. 7, pp. 1019–1031, 2007.

15. M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, “Asymmetric transitivity preserving graph embedding,” in Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2016, pp. 1105–1114.

16. 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. ACM, 2016, pp. 855–864.

17. F. Luo, H. Huang, Y. Duan, J. Liu, and Y. Liao, “Local geometric structure feature for dimensionality reduction of hyperspectral imagery,” Remote Sensing, vol. 9, no. 8, p. 790, 2017.

18. G. Shi, H. Huang, and L. Wang, “Unsupervised dimensionality reduction for hyperspectral imagery via local geometric structure feature learning,” IEEE Geoscience and Remote Sensing Letters, 2019.

19. 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. International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.

20. D. Wang, P. Cui, and W. Zhu, “Structural deep network embedding,” in Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2016, pp. 1225–1234.

21. S. Cao, W. Lu, and Q. Xu, “GraRep: Learning graph representations with global structural information,” in Proceedings of the 24th ACM international conference on information and knowledge management. ACM, 2015, pp. 891–900.

22. Z. Zhang, P. Cui, X. Wang, J. Pei, X. Yao, and W. Zhu, “Arbitrary-order proximity preserved network embedding,” in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2018, pp. 2778–2786.

23. J. Zheng, H. Qiu, X. Xu, W. Wang, and Q. Huang, “Fast discriminative stochastic neighbor embedding analysis,” Computational and mathematical methods in medicine, vol. 2013, 2013.

24. C. Li, X. Xian, X. Ai, and Z. Cui, “Representation learning of knowledge graphs with embedding subspaces,” Scientific Programming, vol. 2020, 2020.

25. A. Tijhuis, A. Franchois, and J.-M. Geoffrin, “Embedding approach to modeling electromagnetic fields in a complex two-dimensional environment,” International Journal of Antennas and Propagation, vol. 2018, 2018.

26. P. Cui, X. Wang, J. Pei, and W. Zhu, “A survey on network embedding,” IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 5, pp. 833–852, 2018.

27. J. B. Tenenbaum, V. De Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” science, vol. 290, no. 5500, pp. 2319–2323, 2000.

28. S. T. Roweis and K. L. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” science, vol. 290, no. 5500, pp. 2323–2326, 2000.

29. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in Advances in neural information processing systems, 2013, pp. 3111–3119.

30. Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” Journal of machine learning research, vol. 3, no. Feb, pp. 1137–1155, 2003.

31. D. Zhu, P. Cui, D. Wang, and W. Zhu, “Deep variational network embedding in wasserstein space,” in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2018, pp. 2827–2836.

32. T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in International Conference on Learning Representations (ICLR), 2017.

33. P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph Attention Networks,” International Conference on Learning Representations, 2018.

34. T. N. Kipf and M. Welling, “Variational graph auto-encoders,” NIPS Workshop on Bayesian Deep Learning, 2016.

35. W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in NIPS, 2017.

36. C. Yang, M. Sun, Z. Liu, and C. Tu, “Fast network embedding enhancement via high order proximity approximation,” in IJCAI, 2017, pp. 3894–3900.

37. D. Lian, K. Zheng, Y. W. Zheng, Y. Ge, L. Cao, I. W. Tsang, and X. Xie, “High-order proximity preserving information network hashing,” in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1744–1753.

38. D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” Nature, vol. 401, no. 6755, pp. 788, 1999.

39. X. Shi, H. Lu, Y. He, and S. He, “Community detection framework based on non-negative factorization using maxflow ct technique based on weight adaptive total-variation and image-spectral tensor factorization for small animals imaging,” IEEE Transactions on Instrumentation and Measurement, 2020.

40. Z. Wang, W. Wang, G. Xue, P. Jiao, and X. Li, “Semi-supervised community detection framework based on non-negative factorization using individual labels,” in International conference in swarm intelligence. Springer, 2015, pp. 349–359.

41. X. Shi, H. Lu, Y. He, and S. He, “Community detection in social network with pairwise constrained symmetric non-negative matrix factorization,” in 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, 2015, pp. 541–546.
K. Tu, P. Cui, X. Wang, P. S. Yu, and W. Zhu, "Deep recursive network embedding with regular equivalence," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2018, pp. 2357–2366.

K. Henderson, B. Gallagher, T. Eliaissi-Rad, H. Tong, S. Basu, L. Akoglu, D. R. Kumburu, C. Faloutsos, and L. Li, "Robs: structural role extraction and mining in large graphs," in Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012, pp. 1231–1239.

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 & Data Mining. ACM, 2018, pp. 1320–1329.

N. K. Ahmed, R. Rossi, J. B. Lee, T. L. Willke, R. Zhou, X. Kong, and H. Eldardiry, "Learning role-based graph embeddings," arXiv preprint arXiv:1802.02896, 2018.

T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:1609.02907, 2016.

N. K. Ahmed, R. A. Rossi, J. B. Lee, T. L. Willke, R. Zhou, X. Kong, and H. Eldardiry, "role2vec: Role-based network embeddings." L. D. Sailer, "Structural equivalence: Meaning and definition, computation and application," Social Networks, vol. 1, no. 1, pp. 73–90, 1978.

K. Nowicki and T. A. B. Snijders, "Estimation and prediction for stochastic blockstructures," Journal of the American statistical association, vol. 96, no. 455, pp. 1077–1087, 2001.

E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," Journal of machine learning research, vol. 9, no. Sep, pp. 1981–2014, 2008.

J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," Journal of the ACM (JACM), vol. 46, no. 5, pp. 604–632, 1999.

B. Mitra, S. Sural, J. Vaidya, and V. Atluri, "A survey of role mining," ACM Computing Surveys (CSUR), vol. 48, no. 4, p. 50, 2016.

J. Li, J. Zhu, and B. Zhang, "Discriminative deep random walk for network classification," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 1004–1013.

D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Collective classification via discriminative matrix factorization on sparsely labeled networks," in Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. ACM, 2016, pp. 1563–1572.

S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in Thirtieth AAAI Conference on Artificial Intelligence, 2016.

D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Collective classification via discriminative matrix factorization on sparsely labeled networks," in Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. ACM, 2016, pp. 1563–1572.

T. Li, J. Zhang, S. Y. Philip, Y. Zhang, and Y. Yan, "Deep dynamic network embedding for link prediction," IEEE Access, vol. 6, pp. 29219–29230, 2018.

R. L. Breiger, S. A. Boorman, and P. Arabie, "An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling," Journal of mathematical psychology, vol. 12, no. 3, pp. 328–383, 1975.

J. B. Krukal, "Nonmetric multidimensional scaling: a numerical method," Psychometrika, vol. 29, no. 2, pp. 115–129, 1964.

G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," in Linear Algebra. Springer, 1971, pp. 134–151.

R. S. Burt and M. J. Minor, Applied network analysis: A methodological introduction. Sage Publications, Inc., 1983.

S. P. Borgatti and M. G. Everett, "Two algorithms for computing regular equivalence," Social networks, vol. 15, no. 4, pp. 361–376, 1993.

E. A. Leicht, P. Holme, and M. E. Newman, "Vertex similarity in networks," Physical Review E, vol. 73, no. 2, p. 026120, 2006.

N. Pizarro, "Structural identity and equivalence of individuals in social networks: beyond duality," International Sociology, vol. 22, no. 6, pp. 767–792, 2007.

I. A. Zager and G. C. Verghese, "Graph similarity scoring and matching," Applied mathematics letters, vol. 21, no. 1, pp. 86–94, 2008.

P. D. Hoff, A. E. Raftery, and M. S. Handcock, "latent space approaches to social networks," Journal of the American Statistical association, vol. 97, no. 460, pp. 1090–1098, 2002.

B. Shi, C. Zhou, H. Qiu, X. Xu, and J. Liu, "Unifying structural proximity and equivalence for network embedding," IEEE Access, vol. 7, pp. 10624–10638, 2019.

M. D. Zeiler, "Adadelta: an adaptive learning rate method," arXiv preprint arXiv:1212.5701, 2012.

D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Network representation learning: A survey," IEEE transactions on Big Data, 2018.
[90] P. Goyal and E. Ferrara, “Graph embedding techniques, applications, and performance: A survey,” Knowledge-Based Systems, vol. 151, pp. 78–94, 2018.

[91] T. Chen, L.-A. Tang, Y. Sun, Z. Chen, H. Chen, and G. Jiang, “Integrating community and role detection in information networks,” in Proceedings of the 2016 SIAM International Conference on Data Mining. SIAM, 2016, pp. 72–80.

[92] L. He, C.-T. Lu, J. Ma, J. Cao, L. Shen, and P. S. Yu, “Joint community and structural hole spanner detection via harmonic modularity,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016, pp. 875–884.

[93] N. Atias and R. Sharan, “Comparative analysis of protein networks: hard problems, practical solutions,” Communications of the ACM, vol. 55, no. 5, pp. 88–97, 2012.

QIN TIAN received the Bachelor degree in School of Computer and Science, Tianjin University in 2018. She is currently pursuing the Master degree from School of Marine Science and Technology, Tianjin University, Tianjin, China. Her research interests include network embedding and data mining.

LIN PAN received the Ph.D. degrees in computer science from Tianjin University, Tianjin, China, in 2016. He is a lecturer with the School of Marine Science and Technology, Tianjin University, Tianjin, China. His current research interests include complex network analysis, social computing, and marine data mining.

XUAN GUO received the Bachelor degree in Computer Science from Tianjin University. Now he is currently pursuing a master’s degree at the School of Computer Science and Technology, Tianjin University, P. R. China. His current research interests include complex network analysis, network representation learning, graph neural networks and role discovery.

XIAOMING LI is a PhD student in School of Computer Science and Technology at Tianjin University, China. Before that, he received his Master’s degree from Xinjiang university in 2008. Currently, He is a lecturer with the School of International Business, Zhejiang Yuyue University of Foreign Languages, Shaoxing, China. His research interests include human behavior dynamics, Multi-layer network local community detection.

WEI YU received the Ph.D. degrees in computer science from Tianjin University, Tianjin, China, in 2020. He is a lecture with the School of International Business, Zhejiang Yuyue University of Foreign Languages, Shaoxing, China. His current research interests include complex network analysis and social computing.

FAMING LI received the Ph.D. degree in computer application technology from Beihang University, Beijing, China, in 2014. He works in the Research Institute for Chemical Defense now and interest in the Knowledge Graph research and its application.