Lower order information preserved network embedding based on non-negative matrix decomposition

Qiang Tian [a,f], Lin Pan [b], Wang Zhang [a], Tianpeng Li [a], Huaming Wu [c], Pengfei Jiao [d,*], Wenjun Wang [a,e,f]

[a] College of Intelligence and Computing, Tianjin University, Tianjin, China
[b] School of Marine Science and Technology, Tianjin University, Tianjin, China
[c] Center for Applied Mathematics, Tianjin University, Tianjin, China
[d] Center of Biosafety Research and Strategy, Tianjin University, Tianjin, China
[e] College of Information Science and Technology, Shihezi University, Shihezi, China
[f] School of Computer and Information Engineering, Tianjin Normal University, Tianjin, China

ARTICLE INFO

Article history:
Received 10 February 2020
Received in revised form 17 February 2021
Accepted 29 April 2021
Available online 4 May 2021

Keywords:
Network embedding
Structural feature
Lower-order information
Non-negative matrix decomposition

ABSTRACT

Network embedding has been successfully used for a variety of tasks, e.g., node clustering, community detection, link prediction and evolution analysis on complex networks. For a given network, embedding methods are usually designed based on first-order proximity, second-order proximity, community constraints, etc. However, they are incapable of capturing the structural similarity of nodes. The bridge nodes with small proximity and located in different communities, should be similar in embedding space since they have the same surrounding structure. In this paper, these structural features are referred to as lower-order information, which could reveal and modify the structural similarity of nodes in the embedding space. Specifically, we propose to construct the feature matrix with the lower-order information of the network. In order to effectively fuse the structural features of nodes into embedding space, an intuitive, interpretable and feasible method named LONE-NMF is proposed, which adopts the representation learning framework based on non-negative matrix factorization. It can effectively learn the representation vectors of nodes in the network via preserving the proximity and lower-order information. Moreover, an optimization algorithm is designed for LONE-NMF. Extensive experiments based on clustering and link prediction show that the proposed method achieves significant performance improvement comparing with some baselines. Finally, we validate the principle and advantage of LONE-NMF through a case study.

1. Introduction

Complex network analysis has been attracting increasing attention and applied to many real-world scenarios, e.g., social networks, citation networks, biological networks and protein interaction networks [1–4]. Network embedding is an important representation technology rising in complex networks, which aims to learn the low-dimensional representation of nodes in the network while preserving its internal structures and characteristics [5]. Benefited from this, various network analysis tasks such as node classification [6,7], node clustering [8], node visualization [9] and link prediction [10,11] can be well supported intuitively in the low-dimensional embedding space based on the off-the-shelf machine learning methods.

At present, most researchers are committed to preserving the neighborhood structural information (first-order), second-order even higher-order proximity and community constraint into embedding space to learn the representations of nodes, and a number of methods have been reported. For example, DeepWalk [5] based on random walk preserves the neighborhood structural information into network embedding. LINE [12] integrates the first-order and second-order proximity between nodes into the embedding learning phase. To preserve more structural information, some methods have been proposed to capture higher-order proximity of nodes [13,14]. In addition, some studies utilize community structural information to enhance the node embedding. Wang et al. [8] proposed a Modularized Non-negative Matrix Factorization (M-NMF) framework that incorporates community structure and pairwise node proximity into a low-dimensional vector space. However, to guarantee the effectiveness, for these community-based methods, the number of communities K should be specified in advance, while in most real-world networks, K is usually unavailable or difficult to indicate. Furthermore, for networks with weak community structure (intra-community edges nearly close to inter-community edges), it may result in an over smoothing problem, i.e., the representations of nodes in the network are too similar to distinguish.

Although these methods have achieved good performance on different network tasks, they often ignore the interaction between the nodes and their surrounding nodes (structural similarity), i.e., the connection pattern between a node and its neighbors. It is usually denoted as structural similarity which represents structural patterns such as star-centers, star-edge, near-cliques or bridge nodes acting as bridges of different subgraphs, and these patterns can reflect the identity or "behavior" of nodes [15]. Two nodes belonging to the same structural category need not be connected by an edge. In fact, nodes in the same community may have different structural functions, however, the nodes in different communities may have the same structural similarity. Herein, we define these structure features as the lower-order information of the network. More recently, a number of studies [15–17] have reported that this lower-order information can enhance the process of role detection which is different from the notion of communities. Therefore, it is necessary to preserve the lower-order information for learning network embeddings comprehensively. One of the main obstacles we face is how to effectively fuse the proximity and lower-order information into embedding space so as to learn more informative and discriminative representations for each node.

In this paper, we propose a Lower-Order information preserved Network Embedding method based on Non-negative Matrix Factorization (LONE-NMF). Considering the interpretability and additivity of Non-negative Matrix Factorization (NMF) [18], whose matrices have no negative elements for dimension reduction and clustering, NMF can effectively map the multiple information into a low-dimensional space while maintaining the intrinsic correlation between data. LONE-NMF integrates structural similarity into NMF to preserve both the lower-order information and proximity. Firstly, we incorporate local structure features and neighbor structure features of nodes recursively to obtain the structure feature matrix, which can indicate the lower-order information of networks. Afterwards, considering that the matrix decomposition of the adjacency matrix can capture the local information of nodes (e.g., the connection closeness [19]), we design a unified loss function elaborately under the NMF by fusing the structure feature matrix and adjacency matrix. Finally, we adopt an iterative multiplicative updating algorithm to optimize the loss function to learn the comprehensive embeddings of nodes in networks. Extensive experiments on various real networks demonstrate the advantages of the proposed model over the state-of-the-art embedding methods. Additionally, we analyze some sub-networks and conduct a case study, which can effectively reveal the principles and advantages of our method.

In summary, our main contributions are listed as follows:

- We propose LONE-NMF, an efficient and scalable algorithm for network embedding, which enhances the effect of structural similarity constraints in embedding space to break through the above-mentioned limitations.
- We incorporate both the proximity and lower-order information into NMF framework. To capture the lower-order information, we recursively combine the local structural features and neighborhood structural features of nodes in the network to construct a structure feature matrix.
- In order to effectively evaluate the proposed model, we conduct extensive experiments, including multi-label classification, clustering and link prediction on several real-world datasets. The results show that our method achieves superior performance than baseline methods.

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

2. Related work

In this section, we give a simple description of the network embedding and the non-negative matrix factorization, which helps to understand our model.

2.1. Network embedding

Network embedding, as an effective and efficient way of network representation, has recently become a very active area and a number of methods have been proposed. As mentioned above, random walk, matrix factorization and deep learning have been widely used in network representation learning [20].

Motivated by word2vec [21], the models based on random walk capture the neighborhood information by generating random paths for nodes and learn representation by treating random paths as sentences. For example, DeepWalk [5] generates random walk sequences for each node of the network and regards the node sequences as sentences in word2vec. Then, these sequences are fed into Skip-Gram [21], which is a classical language model of natural language processing, to learn the network representation. Nevertheless, DeepWalk is not expressive enough to capture the diversity of connectivity patterns in a network. To overcome this limitation, node2vec [22] modifies the strategy of random walk by defining two parameters to balance Depth-First Sampling (DFS) and Breadth-First Sampling (BFS), accordingly it can capture more comprehensive neighborhoods information of nodes. Tang et al. [12] proposed LINE for large-scale network embedding, which integrates first-order proximity and second-order proximity to learn the representation of nodes. The first-order proximity is learned from directly connected nodes and the second-order proximity is learned from the shared neighborhood structures of the nodes. Nevertheless, LINE studies the local and global information of the network separately, and finally simply connects the two representation vectors. Moreover, to capture more internal characteristics, a series of models that preserve high-order proximity are proposed [14,23].

In order to increase nonlinear expression ability and address the sub-optimal network embedding problem, SDNE [13] optimizes the first-order and second-order proximity jointly by a semi-supervised neural network model. The unsupervised part reconstructs the second-order proximity to maintain the global network structure, and the supervised part utilizes the first-order proximity as the monitoring information to preserve the local structure of the network. Furthermore, DVNE [24] integrates a deep variational model and maps the original data to the Wasserstein space to learn the latent representations of nodes. DVNE preserves the first-order and second-order proximity as well as the uncertainties of the nodes, which is beneficial to real-world networks with full uncertainty.

As an effective dimension reduction method, matrix factorization is also used in network embedding. Singular Value Decomposition (SVD) is commonly used in the series of matrix factorization models [11,25]. As a variant of the standard Matrix Factorization (MF), NMF [18] enhances the interpretability and additivity by adding non-negative constraints on matrices, and is usually used to obtain the embedding of nodes.

A recent study [26] indicated that the skip-gram with negative-sampling was implicitly a word-context matrix factorization. The entries of this matrix represent the Positive Point-wise Mutual Information (PPMI) of the corresponding word and its context. Specifically, they factorized the PPMI matrix through SVD to obtain the low-dimension representation of words. Building upon this theoretical foundation, Qiu et al. [27] proved that the models with negative sampling, e.g., DeepWalk, LINE, node2vec and PTE, were regarded as factorizing the matrix with a closed-form and verified their model outperformed DeepWalk and LINE for conventional network mining tasks.

2.2. Non-negative matrix factorization

Given a data matrix $\mathbf{X} = [x_1, x_2, \ldots, x_n] \in \mathbf{R}_+^{m\times n}$, where n is the number of data point and m is the dimension of data, non-negative matrix factorization (NMF) aims to find two non-negative matrices $\mathbf{U} = [u_1, u_2, \ldots, u_d] \in \mathbf{R}_+^{m\times d}$ and $\mathbf{V} = [v_1, v_2, \ldots, v_n] \in \mathbf{R}_+^{d\times n}$, so that their product approximate to the original matrix $\mathbf{X}$: $\mathbf{X} \approx \mathbf{UV}$, where $d \ll \min(m, n)$ is the dimension of the latent space or the rank of the underlying data. $\mathbf{U}$ is the basis matrix and $\mathbf{V}$ is the coefficient matrix. Here, each data point x_i can also be written as $x_i \approx \sum_i^d u_i v_{ij}$, where v_i represents the basis vectors or the latent feature vectors of X, which means the observation x_i is decomposed into the additive linear combination of the basis vectors.

As a popular low-rank matrix decomposition model, NMF has been applied to various data mining tasks such as community detection, link prediction and network embedding. There have been a number of studies that demonstrate the effectiveness of NMF in community detection [28–31]. In general, the community detection methods based on NMF framework obtain the community relationships of nodes by factorizing the adjacent matrix of a network into low-rank factor matrices [32,31]. Other related methods of adding various constraints could refer to [8,33] and so on. In addition to community detection, NMF is widely used in link prediction due to the excellent capacity of describing network properties by matrix factorization [34–38].

For network embedding, recently, Proximity Preserving Non-negative Matrix Factorization (PPNMF) [39] preserves the first-order proximity and the second-order proximity independently in its loss function. Wang et al. [8] proposed a Modularized Non-negative Matrix Factorization (M-NMF) framework, which simultaneously investigates the community structure and first-order and second-order similarity in a low-dimensional vector space. In particular, they defined a modularity constraint term for community structure and preserved first-order similarity and second-order similarity by factorizing the pairwise node similarity matrix. To be more effective, higher-order proximity between nodes is taken into account, which is very crucial to grasp the global characteristics of the network. However, the proximities of different orders are often desired for distinguishing networks and target applications. Hence, Zhang et al. [40] proposed ARbitrary-Order Proximity preserved Embedding (AROPE) based SVD framework to shift embedding vectors across arbitrary orders and demonstrate the intrinsic relationship between them. The aforementioned NMF-based algorithms mainly adopt shallow methods that are not incapable of reflecting the nonlinearity relationship between the original network and embedding space. Ye et al. [41] incorporated an encoder component and decoder component into NMF to capture the hidden characteristic. It is obvious that almost all of the above network embedding frameworks only consider proximity between nodes or community property, which loss

the interaction information of nodes and their neighborhoods. Hence, in this paper, we incorporate the structural similarity which reveals the interaction between nodes and surroundings into an NMF framework.

3. Methods

In this section, we present our proposed method LONE-NMF in detail. LONE-NMF preserves both the structural similarity information (connection pattern) and proximity information (connection closeness) as shown in Fig. 1. We first report the notations used in this paper. Then, we introduce how to extract lower-order information and integrate it with NMF, following the optimization algorithm to learn the low-dimensional representations of nodes.

3.1. Notations

Given an undirected network $G = (V, E)$ with n nodes and e edges, $V = [v_1, v_2, \ldots, v_n]$ denotes the set of nodes and E denotes the set of edges among the nodes. G is represented by the adjacency matrix $\mathbf{A} \in \mathbf{R}^{n\times n}$, $\mathbf{A_{ij}} = 1$ if there exists an edge between node i and node j, otherwise $\mathbf{A_{ij}} = 0$. Since the network G is undirected, $\mathbf{A}$ is a symmetric matrix. For each node v, the set of its neighborhood is denoted as $\mathcal{N}(v) = \{u|(v,u) \in E\}$. In addition, egonet is widely used in social network analysis, which can depict the neighbor information of nodes. The set of nodes in a specific node's ego network includes the node (ego) and its neighbours, the set of edges includes any edges in the subgraph on these nodes. The egonet of node v is denoted as $\widehat{G}(v) = \left(V_{\widehat{G(v)}}, E_{\widehat{G(v)}}\right)$. Likewise, $V_{\widehat{G(v)}}$ and $E_{\widehat{G(v)}}$ denote the set of nodes and edges in the egonet, respectively, $V_{\widehat{G(v)}} = \{v\} \cup \mathcal{N}(v)$and $E_{\widehat{G(v)}} = \left\{((i,j)|(i,j) \in E) \wedge \left(i,j \in V_{\widehat{G(v)}}\right)\right\}$. $\Gamma(v) = \{\{v,i,j\}|(v,i),(v,j),(i,j) \in E\}$ denotes the set of triangles containing node v and $|\Gamma(v)|$ indicates the size, respectively. The matrix $\mathbf{S} \in \mathbf{R}^{n\times m}$ preserves the structural similarity feature of nodes in the network, where m is the dimension of the feature. The i-th row of $\mathbf{S}$ represents the structural feature of node i. Our purpose is to learn the representations of nodes $\mathbf{X} \in \mathbf{R}^{n\times d}$ $(d \leqslant n)$, where d is the dimension of representations. The terms and notations are listed in Table 1.

3.2. LONE-NMF

The proposed LONE-NMF is under the NMF framework, and takes lower-order information and proximity into consideration simultaneously. We define the structural similarity features, including local and neighbor structural features, as the lower-order information. In this section, we first present how to encode the lower-order information, and then integrate them into a unified model based on NMF.

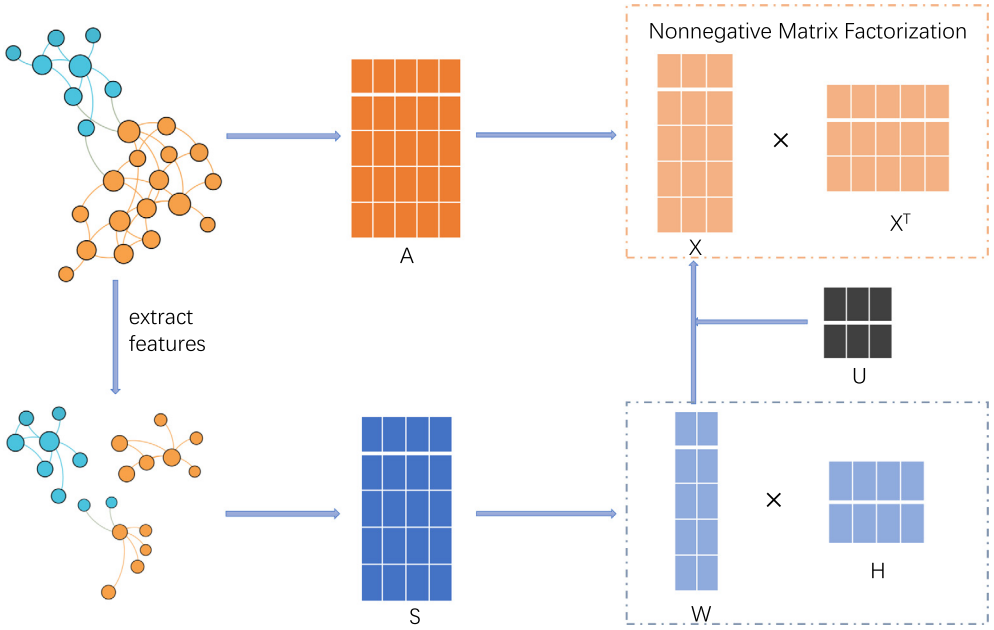


Fig. 1. The framework of LONE-NMF.

Table 1
Notations and their definitions.

Notation	Definition
$G = (V, E)$	The graph G with node set V and edge set E
$\mathbf{A} \in \mathbf{R}^{n \times n}$	The adjacency matrix of G
$\mathbf{S} \in \mathbf{R}^{n \times m}$	The structural feature matrix of nodes
$\mathbf{X} \in \mathbf{R}^{n \times d}$	The representations of nodes
$\mathcal{N}(v) = \{u \mid (v, u) \in E\}$	The set of node v's neighbors
$\widehat{G}(v) = \left(V_{\widehat{G}(v)}, E_{\widehat{G}(v)}\right)$	The egonet of node v
$\Gamma(v)$	The set of triangles containing node v

Lower-Order Information Encoder. In this paper, we extract the structural features of nodes to capture the lower-order information. The overall process is shown in Algorithm 1. Specifically, inspired by ReFeX [42], we adopt two types of features, i.e., local and egonet features of nodes as the initial input. We extract six types of features of nodes in all and summarize them in Table 2.

We construct the primary structural feature matrix $\mathbf{F}$ whose row is a 24-dimensional vector corresponding to the six features, each of which are encoded by a four-dimensional one-hot vector following [42]. Based on $\mathbf{F}$, we aggregate the features by computing the sum and the mean of node features in each egonet: $\widetilde{\mathbf{F}} = (\mathbf{AF}) \circ \left(\mathbf{D}^{-1}\mathbf{AF}\right)$, where $\mathbf{D}$ is a diagonal matrix whose diagonal elements are the degree of nodes, i.e., $\mathbf{D_{ii}} = \sum_j \mathbf{A_{ij}}$, $\mathbf{A}$ is the adjacency matrix of graph G and $\circ$ means the concatenation operator. Then, we continue to aggregate the features recursively (Alg. 1, line 5–9). As the times of the recursion increases, we can learn the more abundant structural characteristics gradually.

Algorithm 1: Process of extracting structural similarity feature

Input: The network $G = (V, E)$, the adjacency matrix $\mathbf{A}$, the number of recursions γ.
Output: structural similarity feature matrix $\mathbf{S}$.

1: Extract the egonet $\widehat{G}(v)$ for each node v;
2: Compute the f_i, $i = 1, 2, \ldots, 6$ for each node;
3: Construct the initial feature matrix $\mathbf{F}$ by converting each feature f_i to a one-hot vector with four dimensions;
4: $\widetilde{\mathbf{F}} = (\mathbf{AF}) \circ \left(\mathbf{D}^{-1}\mathbf{AF}\right)$;
5: $\mathbf{S} = \mathbf{F} \circ \widetilde{\mathbf{F}}$
6: **for** $i = 1$ to $\gamma - 1$ do:
7: $\quad \widetilde{\mathbf{F}} = \left(\mathbf{A}\widetilde{\mathbf{F}}\right) \circ \left(\mathbf{D}^{-1}\mathbf{A}\widetilde{\mathbf{F}}\right)$;
8: $\quad \mathbf{S} = \mathbf{S} \circ \widetilde{\mathbf{F}}$
9: **end for**
10: **return S**.

The Unified Network Embedding Model. In this section, we aim to integrate the above structural similarity feature matrix $\mathbf{S}$ into the NMF framework to guide the learning process of final representation matrix $\mathbf{X}$. In NMF, we introduce a non-negative basis matrix $\mathbf{W} \in \mathbf{R}^{n \times r}$ and a non-negative representation matrix $\mathbf{H} \in \mathbf{R}^{r \times m}$, where r is the number of structure types in $\mathbf{S}$ and $\mathbf{W_i}$ is the relationship between nodes i and structural similarity categories. Then, we make use of these two matrices to approximate the structural similarity feature matrix $\mathbf{S}$, which gives rise to the following objective function:

$$\min \|\mathbf{S} - \mathbf{WH}\|_F^2, \quad s.t., \quad \mathbf{W} \geqslant 0, \mathbf{H} \geqslant 0, \tag{1}$$

Table 2
Features and their definitions.

Feature	Definition	Formulation
f_1	The degree of v	$f_1 = \lvert\mathcal{N}(v)\rvert$
f_2	The number of edges in the egonet of v	$f_2 = \lvert E_{\widehat{G}(v)}\rvert$
f_3	The sum of node's degrees in the egonet of v	$f_3 = \sum_{\widehat{G}(v)} \lvert\mathcal{N}(u)\rvert$
f_4	The estimated ratio of within-egonet edges in the egonet of v	$f_4 = f_2/f_3$
f_5	The estimated ratio of non-egonet edges in the egonet of v	$f_5 = 1 - f_2/f_3$
f_6	The clustering coefficient of v	$f_6 = \lvert\Gamma(v)\rvert/(f_1(f_1 - 1))$

where $\|\cdot\|_F$is the Frobenius norm of the matrix.

Here, we introduce an auxiliary matrix $\mathbf{U} \in \mathbf{R}^{r\times d}$, where d is the dimension of the node embeddings to be learned and each row of $\mathbf{U}$ represents the mapping vector of the structure type i. The matrix $\mathbf{W}$ preserves the propensity between nodes and structure categories, and it can provide effective guidance to correct the embeddings of all nodes. Hence, we expect $\mathbf{WU}$ to reconstruct the embedding matrix $\mathbf{X}$. Together with the objective function (Eq. (1)), the final loss function of our method can be denoted as:

$$\mathcal{L} = \min_{\mathbf{X},\mathbf{W},\mathbf{H},\mathbf{U}} \left\|\mathbf{A} - \mathbf{X}\mathbf{X}^T\right\|_F^2 + \alpha\|\mathbf{S} - \mathbf{W}\mathbf{H}\|_F^2 + \beta\|\mathbf{X} - \mathbf{W}\mathbf{U}\|_F^2, \tag{2}$$

$$s.t., \quad \mathbf{X} \geqslant 0, \mathbf{W} \geqslant 0, \mathbf{H} \geqslant 0, \mathbf{U} \geqslant 0, \tag{3}$$

where α and β are positive parameters for adjusting the contribution of the corresponding terms.

Optimization. Since the objective function in (Eq. (2)) is not convex, it is impractical to compute the optimal solution by derivative. Herein, considering that there are four parameter matrices to optimize ($\mathbf{W}, \mathbf{H}, \mathbf{S}, \mathbf{U}$), we divide our objective function into four subproblems, so that we can compute the local minimum of each problem by Majorization-Minimization framework [43]. The update strategy we adopted is alternating optimization, i.e., fixing the other three matrices when updating one. The specific formulas are shown as the following.

X-subproblem: Updating $\mathbf{X}$ with other fixed parameters $\mathbf{W}, \mathbf{H}$ and $\mathbf{V}$, which can be expressed as the following suboptimization problem:

$$\min_{\mathbf{X}} \left\|\mathbf{A} - \mathbf{X}\mathbf{X}^T\right\|_F^2 + \beta\|\mathbf{X} - \mathbf{W}\mathbf{U}\|_F^2, \quad s.t., \quad \mathbf{X} \geqslant 0. \tag{4}$$

Since $\mathbf{X}$ satisfies the non-negative constraint, the Lagrange multiplier matrix Θ is introduced, resulting in the following equivalent objective function:

$$L(\mathbf{X}) = tr\left(\mathbf{A}\mathbf{A}^T - \mathbf{A}\mathbf{X}\mathbf{X}^T - \mathbf{X}\mathbf{X}^\mathbf{T}\mathbf{A}^T + \mathbf{X}\mathbf{X}^T\mathbf{X}\mathbf{X}^T\right) + \beta \cdot tr\left(\mathbf{X}\mathbf{X}^T - \mathbf{X}\mathbf{U}^T\mathbf{W}^T - \mathbf{W}\mathbf{U}\mathbf{X}^T + \mathbf{W}\mathbf{U}\mathbf{U}^T\mathbf{W}^T\right) + tr\left(\Theta\mathbf{X}^T\right). \tag{5}$$

Through setting the value of the objective function in (5) to 0, i.e., $L(\mathbf{X}) = 0$, we have:

$$\Theta = 4\mathbf{A}\mathbf{X} - 4\mathbf{X}\mathbf{X}^T\mathbf{X} - 2\beta\mathbf{X} + 2\beta\mathbf{W}\mathbf{U}. \tag{6}$$

Following the Karush–Kuhn–Tucker (KKT) condition for the non-negativity of $\mathbf{X}$, we have the following equation:

$$\left(4\mathbf{A}\mathbf{X} - 4\mathbf{X}\mathbf{X}^T\mathbf{X} - 2\beta\mathbf{X} + 2\beta\mathbf{W}\mathbf{U}\right) \cdot \mathbf{X} = \Theta\mathbf{X} = 0. \tag{7}$$

Given an initial value of $\mathbf{X}$, the updating rule for $\mathbf{X}$ is:

$$\mathbf{X} \leftarrow \mathbf{X} \odot \left(\frac{2\mathbf{A}\mathbf{X} + \beta\mathbf{W}\mathbf{U}}{2\mathbf{X}\mathbf{X}^T\mathbf{X} + \beta\mathbf{X}}\right), \tag{8}$$

where $\odot$ denotes the matrix multiplication.

W-subproblem: Updating $\mathbf{W}$ with other parameters $\mathbf{X}, \mathbf{H}$ and $\mathbf{U}$ fixed:

$$\min_{\mathbf{W}} \alpha\|\mathbf{S} - \mathbf{W}\mathbf{H}\|_F^2 + \beta\|\mathbf{X} - \mathbf{W}\mathbf{U}\|_F^2, \quad s.t., \quad W \geqslant 0. \tag{9}$$

Similar to the optimization process of $\mathbf{X}$, we have the following updating rule:

$$\mathbf{W} \leftarrow \mathbf{W} \odot \left(\frac{\alpha\mathbf{S}\mathbf{H}^T + \beta\mathbf{X}\mathbf{U}^T}{\alpha\mathbf{W}\mathbf{H}\mathbf{H}^T + \beta\mathbf{W}\mathbf{U}\mathbf{U}^T}\right). \tag{10}$$

H-subproblem: Updating $\mathbf{H}$ with other fixed parameters $\mathbf{X}, \mathbf{W}$ and $\mathbf{U}$, we need to solve the following problem:

$$\min_{\mathbf{H}} \alpha\|\mathbf{S} - \mathbf{W}\mathbf{H}\|_F^2, \quad s.t., \quad \mathbf{H} \geqslant 0. \tag{11}$$

In the same way with $\mathbf{X}$, the updating rule for $\mathbf{H}$ is then given as follows:

$$\mathbf{H} \leftarrow \mathbf{H} \odot \left(\frac{\mathbf{W}^T\mathbf{S}}{2\mathbf{W}^T\mathbf{W}\mathbf{H}}\right). \tag{12}$$

U-subproblem: Updating $\mathbf{U}$ with other parameters $\mathbf{X}, \mathbf{W}, \mathbf{H}$ fixed leads to the following problem:

$$\min_{\mathbf{U}} \beta\|\mathbf{X} - \mathbf{W}\mathbf{U}\|_F^2, \quad s.t., \quad \mathbf{U} \geqslant 0. \tag{13}$$

Similarly, optimize $\mathbf{U}$ according to the process of optimizing $\mathbf{X}$, we adopt the following rule to update:

$$\mathbf{U} \leftarrow \mathbf{U} \odot \left(\frac{\mathbf{W}^T \mathbf{X}}{\mathbf{W}^T \mathbf{W} \mathbf{U}} \right). \tag{14}$$

The optimization workflow of our method is shown in Algorithm 2. The input of our model is the network G, the structural similarity feature matrix $\mathbf{S}$, the embedding dimension d, the structure categories r, the convergence coefficient δ, and the balance parameters α and β. Firstly, randomly initialize $\mathbf{X}, \mathbf{U}, \mathbf{W}$ and $\mathbf{H}$ with a uniform distribution. Then, we update $\mathbf{X}, \mathbf{U}, \mathbf{W}$ and $\mathbf{H}$ iteratively until convergence (Algorithm 2, Line 2–12). The output is the embedding matrix $\mathbf{X}$ for all nodes, the structural similarity matrix $\mathbf{S}$ and auxiliary matrix $\mathbf{U}$. Our model can learn representations of nodes by integrating the proximity and structural features.

3.3. Complexity analysis

The whole computational complexity of our model depends on the matrix multiplication in the updating rules. As far as we know, the computation of the matrix product $\mathbf{AB}$ is $O(xyz)$, where $\mathbf{A} \in \mathbf{R}_+^{x \times y}$ and $\mathbf{B} \in \mathbf{R}_+^{y \times z}$. The computation of updating rules in (8), (10), (12) and (14) run in $O\left(n^2 d + nrd + nd^2\right), O\left(nmr + ndr + nr^2 + mr^2 + dr^2\right), O\left(mnr + nr^2 + mr^2\right)$ and $O\left(ndr + nr^2 + dr^2\right)$, respectively. Since generally $m, r, d \leqslant n$, the overall computation is $O(n^2 m)$. 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 $O(nem)$, where e is the edge number in the network. We can find that the complexity of our model is the same order of magnitude as most NMF-based algorithms.

Algorithm 2: Optimization of the objective function of LONE-NMF

Input: The network G, the structural similarity features matrix $\mathbf{S}$, the embedding dimension d, the structural similarity categories r, the convergence coefficient δ, and the balance parameters α and β.
Output: $\mathbf{X}$ of size $n \times d$, $\mathbf{W}$ of size $n \times r$, $\mathbf{U}$ of size $r \times d$,.

1: Initialize $\mathbf{X}, \mathbf{W}, \mathbf{U}$;
2: **while not** $conv$ do:
3: **if** $\frac{\mathcal{L}^i - \mathcal{L}^{i-1}}{\mathcal{L}^{i-1}} < \delta$ **then**
4: $conv \leftarrow$ **true**;
5: **end if**
6: $\mathbf{X} \leftarrow \mathbf{X} \odot \left(\frac{2\mathbf{AX} + \beta \mathbf{WU}}{2\mathbf{XX}^T\mathbf{X} + \beta \mathbf{X}} \right)$;
7: $\mathbf{W} \leftarrow \mathbf{W} \odot \left(\frac{\alpha \mathbf{SH}^T + \beta \mathbf{XU}^T}{\alpha \mathbf{WHH}^T + \beta \mathbf{WUU}^T} \right)$;
8: $\mathbf{H} \leftarrow \mathbf{H} \odot \left(\frac{\mathbf{W}^T \mathbf{S}}{2\mathbf{W}^T \mathbf{WH}} \right)$;
9: $\mathbf{U} \leftarrow \mathbf{U} \odot \left(\frac{\mathbf{W}^T \mathbf{X}}{\mathbf{W}^T \mathbf{WU}} \right)$;
10: compute loss function $\mathcal{L}$ using Eq. (2);
11: **end while**
12: **return** $\mathbf{X}, \mathbf{W}, \mathbf{U}$.

4. Experiments

In this section, we conduct extensive experiments to evaluate our model. We first report the datasets and experimental settings in this paper and then compare our method with popular competitive baselines to indicate the effectiveness of our model. Finally, parameter sensitivity analysis and some case studies are given.

4.1. Datasets and experimental settings

We employ six real-world network datasets for sensitivity analysis with ground truth in our experiments. The statistical characteristics of these networks are shown in Table 3 and the detailed information is introduced as follows.

- Polblog [44]: It is a political blogging network. The nodes represent the blogs of US politicians and the edges represent the web links between blogs. According to the politicians to which these blogs belong, blogs are divided into two factions.
- Livejournal & Livejournal-L [45]: They come from the same network that is also a free online blogging network. The difference between them is the size. The nodes and edges represent the users who have registered and friendship between them, respectively. The user-defined groups are treated as ground-truth communities.

Table 3
The statistical characteristics of different networks.

Dataset	Nodes	Edges	Classes	Density
Polblog	1490	16627	2	0.022
Livejournal	516	15020	5	0.113
Livejournal-L	11118	396461	26	0.006
Orkut	998	23050	6	0.046
Amazon	641	2091	10	0.010
Cornell	183	280	5	0.015

- Orkut [45]: It is an online making friends network. The nodes represent the users and edges represent they are friends. We regard groups formed spontaneously by users as ground-truth communities.
- Amazon [45]: It is an e-commerce network. The nodes are the products and if a product i is frequently co-purchased with product j, there is an edge between them. The communities are product categories provided by Amazon.
- Cornell [46]: It is a citation network. The nodes represent webpages gathered from Cornell universities and the edges represent citation relationships between them. These webpages are classified into five classes.

The parameters of our model LONE-NMF include two hyperparameters α and β, the structure type r and embedding dimension d. In the experiment, we set α and $\beta \in [1, 101], r \in [1, 10], d = 128$. The details of parameter analysis are shown in Section 4.5. In the process of extracting the structural feature matrix $\mathbf{S}$ recursively, we set the number of recursion to 2. Hence, the number of columns in the matrix $\mathbf{S}$ representing the features is $m = 168$.

4.2. Baseline methods

To validate that our model can learn better representations of nodes, we compare it with two recent NMF-based methods and four popular network embedding methods. We set the parameters in these methods to default values, which are listed as follows.

- **MNMF** [8]: MNMF incorporates community structure by a modularity constraint term, at the same time, preserve the first-order and second-order proximities to learn node embeddings. We set the dimension as 128, other parameters of MNMF use the default values.
- **DANMF** [41]: DANMF integrates an encoder and a decoder component into NMF to learn node representations. All parameters of DANMF are the default values.
- **DeepWalk** [5]: Deepwalk, based on random walks, treats the random walk sequences of nodes as sentences in the language model. And then adopts skip-gram [21] to learn the representations of nodes. We use the default settings in the experiments. The number of random walks for each node is 10, walk-length is 80 and the window size of the skip-gram model is 8.
- **Node2vec** [22]: Node2vec is an extension of DeepWalk. The difference is that Node2vec uses a biased random walk, which introduces two hyperparameters to control the random walk strategy. We set two hyperparameters $p = 1.0$ and $q = 1.0$, respectively.
- **LINE** [12]: LINE preserves the first-order proximity and second-order proximity separately to obtain representations of nodes. We set the negative ratio to 5.
- **SDNE** [13]: SDNE uses deep neural networks to optimize first-order and second-order proximity simultaneously. We also adopt default settings, where the number of the neurons at each encoder layer is 1000 and the dimension of the output node representations is 128.
- **VGAE** [47]: VGAE leverages GCN to construct a variational auto-encoder, and it reconstruct the adjacency matrix to capture proximity. We set the epoch to 200 and the dimension of node embedding is 128.

4.3. Node clustering

In this subsection, we evaluate the performance for node clustering based on typical metrics, namely, Normalized Mutual Information (NMI) and accuracy. The range of these metrics is from 0 to 1 and a larger value indicates better performance. To explore the influence of different structural features, we replace the feature matrix $\mathbf{S}$ with another method RoleSim [48] (LONE-NMF-R), and degree matrix(LONE-NMF-D). For network embedding methods, we apply the standard k-means algorithm to obtain the clustering results. 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. Tables 4 and 5 show the node clustering performance about NMI and accuracy, respectively. In these tables, bold numbers represent the best results.

As we can see, LONE-NMF achieves the best performance among all network datasets on NMI and accuracy. Especially, on the Orkut dataset, our method LONE-NMF achieves 36 percent improvement on NMI compared with the second-best method. This benefits from the integration of structural features in our method, which can reveal the lower-order informa-

tion of networks. DeepWalk and node2vec based on random walks can capture the second even higher-order proximities. However, for bridge nodes that directly connected but not belong to the same community, they can not distinguish effectively. In additive, SDNE and LINE only preserve the microscopic structure of the network, which is incapable to effectively capture community structure. DANMF only takes the adjacent matrix as input, which results in the absence of other constraints that affect the clustering effect. MNMF adds modularity term to learn the embedding of nodes. However, for the networks that are very sparse and their community structure is not obvious, the modularity constraint of NMF makes the representation of nodes similar to each other, so it shows relatively low performance. The above results demonstrate the superior power of providing structural similarity information to learn node embedding. Besides, LONE-NMF-D and LONE-NMF-R have batter performances than baselines on node clustering, but not as good as LONE-NMF. This proves the effectiveness of our feature matrix S.

4.4. Link prediction

In the link prediction task, given a network that is removed a portion of the edges proportionally, and we would like to predict these missing edges. We randomly remove $50\%, 40\%, 30\%, 20\%, 10\%$ edges as a test set for evaluation while ensuring that the remaining network is connected, and use the remaining edges to learn the node representations respectively. In our experiment, a typical metric named Area Under Curve (AUC) score is used to evaluate the performance of our model for link prediction. Specifically, we take the results on Orkut and Amazon as examples under different portions of removing edges. From Fig. 2, it can be found that our proposed method outperforms consistently all the baseline methods in the two datasets in regardless of the removing portions of edges.

In addition, we take the results of 10% on all six datasets to further explore the performance of our method. From the results in Table 6, we can conclude that our model achieves $13.6\%, 12\%$ and 26.6% improvements on Livejournal, Orkut and Amazon, respectively. We notice that our method on Polblog dataset is next only to MNMF with superior predictive performance, hence it is still competitive. The reason may be that Polblog network has a high degree of community closeness, the MNMF which integrates the community structure will perform better than our model without community information. In general, our method could achieve superior performance in link prediction, which indicates the effectiveness of our LONE-NMF.

4.5. Parameter analysis

In this subsection, we analyze the effect of parameters α, β and r of LONE-NMF for clustering performance on the real-world networks, where r is the number of structure categories. Experiments show that different datasets show similar trends, so we just utilize two datasets as examples.

We first evaluate the effects of α and β on Orkut and Cornell by fixing r at 5. As depicted in Fig. 3, we notice that from the vertical-level, NMI does not change too much, which suggests the performances are relatively stable as α increases. However, β has a heavy effect on the performance of clustering.

When testing the effect of r, we randomly select $\alpha = 1, \beta = 90$ and vary r from 1 to 10 with an increment of 1. The results of Orkut and Amazon are shown in Fig. 4(a). Since $r = 1$ represents no structural difference, the performance is the worst. It can be seen that NMI has a biggish increase when r from 3 to 4, which reflects four typical structures that are common for most social networks. As the increment of r, the value of NMI is also increasing and gradually stable, which indicates a better capacity of expression. We choose different r for different datasets to achieve high performance.

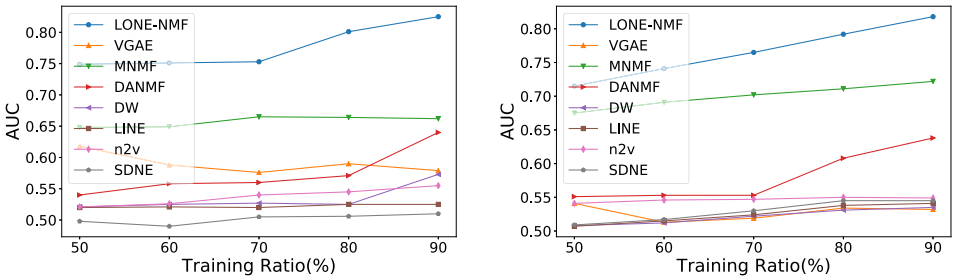
In addition, the effect of embedding dimensions d is shown in Fig. 4(b). Here, on Amazon and Livejournal, we fix other parameters $\alpha = 1, \beta = 80, r = 9$ and vary d from 2 to 256. We notice that the NMI of both two datasets tends to be stable when $k > 32$, which indicates the robustness of LONE-NMF to the embedding dimensions d.

Table 4
Performance evaluation based on NMI.

Dataset	Polblog	Livejournal	Livejournal-L	Orkut	Amazon	Cornell
MNMF	0.005	0.477	0.534	0.212	0.045	0.008
DANMF	0.510	0.396	0.391	0.217	0.153	0.081
DeepWalk	0.473	0.647	0.103	0.020	0.024	0.035
Node2vec	0.453	0.723	0.028	0.031	0.035	0.064
LINE	0.220	0.732	0.505	0.014	0.006	0.087
SDNE	0.067	0.740	0.240	0.013	0.022	0.066
VGAE	0.431	0.490	0.341	0.223	0.151	0.044
LONE-NMF-D	0.529	0.536	0.281	0.277	0.195	0.089
LONE-NMF-R	0.508	0.753	0.434	0.374	0.303	0.096
LONE-NMF	**0.545**	**0.818**	**0.592**	**0.578**	**0.313**	**0.138**

Table 5
Accuracy of node clustering.

Dataset	Polblog	Livejournal	Livejournal-L	Orkut	Amazon	Cornell
MNMF	0.828	0.269	0.126	0.088	0.156	0.431
DANMF	0.076	0.269	0.231	0.102	0.166	0.062
DeepWalk	0.527	0.200	0.196	0.153	0.160	0.328
Node2vec	0.848	0.198	0.327	0.139	0.129	0.344
LINE	0.574	0.399	0.432	0.146	0.137	0.353
SDNE	0.426	0.387	0.145	0.172	0.156	0.031
VGAE	0.165	0.297	0.112	0.142	0.136	0.205
LONE-NMF-D	0.873	0.485	0.331	0.274	0.221	0.388
LONE-NMF-R	0.825	0.593	0.502	0.325	0.303	0.437
LONE-NMF	**0.953**	**0.691**	**0.531**	**0.371**	**0.289**	**0.477**

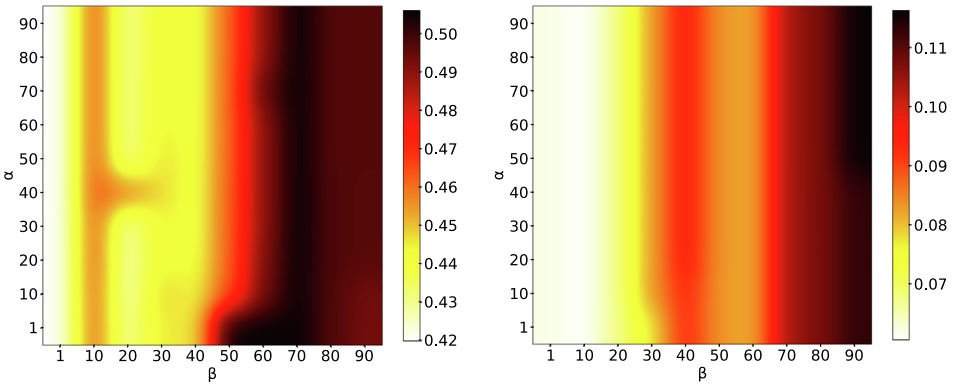


(a) Orkut (b) Amazon

Fig. 2. Predictive performance w.r.t the ratio of training data.

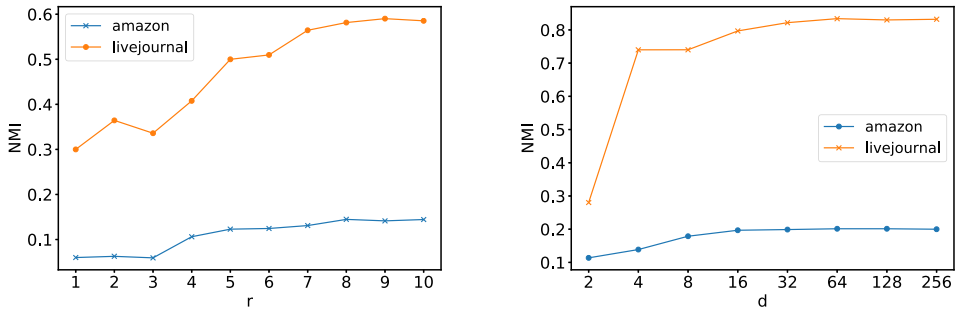
Table 6
The AUC scores of different methods.

Dataset	Polblog	Livejournal	Livejournal-L	Orkut	Amazon	Cornell
MNMF	0.664	0.683	0.812	0.679	0.726	0.542
DANMF	0.510	0.657	0.646	0.655	0.632	0.502
DeepWalk	0.499	0.591	0.512	0.592	0.567	0.525
Node2vec	0.490	0.586	0.498	0.579	0.531	0.538
LINE	0.462	0.495	0.518	0.534	0.553	0.507
SDNE	0.464	0.501	0.509	0.521	0.560	0.484
VGAE	0.505	0.687	0.656	0.579	0.532	0.509
LONE-NMF	0.515	**0.815**	**0.838**	**0.821**	**0.824**	**0.568**



(a) Orkut (b) Cornell

Fig. 3. The effect of α and β.



(a) The effect of r (b) The effect of d

Fig. 4. Parameter sensitivity analysis.

4.6. Visualization

Furthermore, we show the embeddings generated by our method and baselines in Fig. 5. The representations learned from the models are mapped into a two-dimensional space, where the colors of nodes represent their labels. The closer the nodes with the same color are, the better effect the models have. Here, the label is the community ground truth. It is intuitive that our model has the best clustering performance.

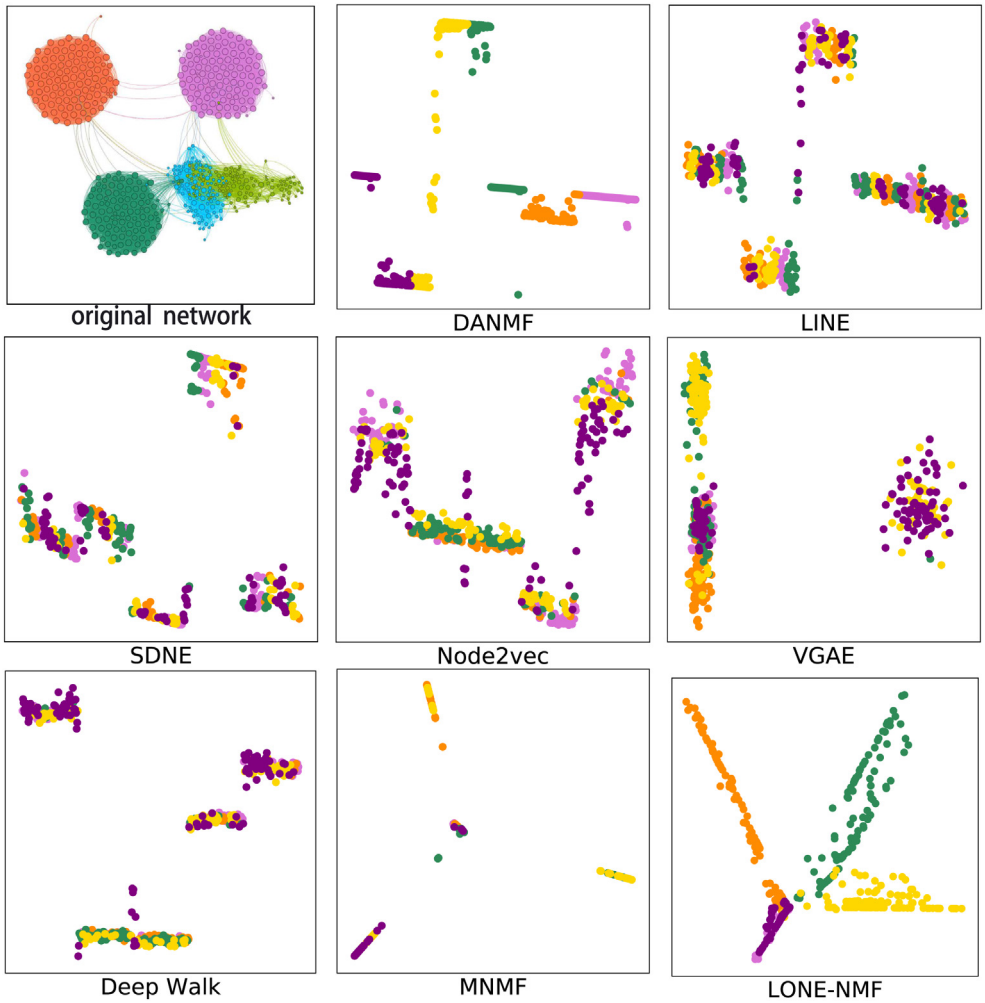
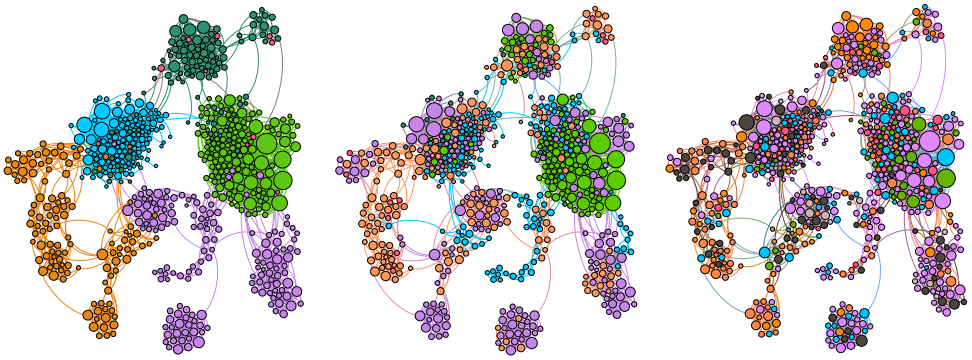
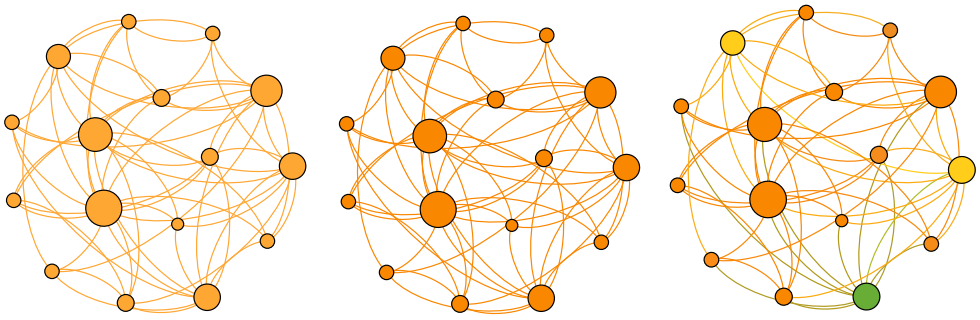


Fig. 5. Visualization of node representations on the Livejournal network.



(a) Ground truth (b) LONE-NMF (c) DANMF

Fig. 6. Clustering results on the Amazon network. The color represents the community and the node size represents its degree.



(a) Ground truth (b) LONE-NMF (c) DANMF

Fig. 7. Clustering results on part of the Amazon network.

4.7. Case study

In this subsection, we analyze a real-world network to verify the validity of LONE-NMF and the reason why our method has better performance.

Fig. 6 shows the clustering results of LONE-NMF and DANMF on the Amazon network. As shown before, the clustering performance of DANMF is the best among baseline methods, hence, we only compare LONE-NMF with DANMF.

To be more clearly understood, we select the part of Amazon as shown in Fig. 7. We notice that DANMF divides nodes that should be the same class into multiple classes. DANMF adds an extra abstraction layer of the similarity between nodes in the matrix factoring process. In essence, it merely captures the first-order topology of nodes, which can not fully represent the community characteristics. LONE-NMF not only preserves this information, but also integrates additional lower-order information of nodes. For example, as depicted in Fig. 7, the degree of misclassified nodes (green and yellow nodes) are similar and are all connected to the orange nodes with the same structural similarity category. By adjusting the parameters, LONE-NMF can balance the importance of first-order topology and structural similarity information, thus realizing the correct classification of them.

Although there are differences between our model and ground truth, on the whole, the co-purchased relationship for E-commerce platforms is heavily influenced by attribute information. In the real world, getting attribute information is more difficult than structure information. Results show that our model can learn better network embeddings for such imperfect data.

5. Conclusion and future work

In this paper, we propose LONE-NMF, a novel NMF-based model for learning the embeddings of nodes in complex networks. We define the lower-order information and the model can preserve the structural similarity and structural features simultaneously, which can model the nodes in a more unified and comprehensive manner. In the optimization phase, we propose to adopt the alternating optimization algorithm to train the parameters in our model effectively. The extensive

experimental results on node clustering, link prediction, visualization tasks and case study demonstrate that our method could learn more effective embedding vectors of nodes.

In this work, we only focus on undirected and unweighted networks, but it would be interesting to explore preserving more inherent characteristics in directed and weighted networks in the future. In addition, how to improve computational efficiency while maintaining accuracy is an interesting topic.

CRediT authorship contribution statement

Qiang Tian: Validation, Writing - review & editing. **Lin Pan:** Data curation, Writing - original draft. **Wang Zhang:** Visualization, Writing - review & editing. **Tianpneg Li:** Data curation, Writing - original draft, Software. **Huaming Wu:** Writing - review & editing, Conceptualization. **Pengfei Jiao:** Writing - review & editing, Supervision. **Wenjun Wang:** Conceptualization & Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the National Key R&D Program of China (2018YFC0831005) and the National Natural Science Foundation of China (61902278).

References

[1] T. Kuhn, M. Perc, D. Helbing, Inheritance patterns in citation networks reveal scientific memes, Physical Review X 4 (4) (2014) 041036.
[2] T. Nepusz, H. Yu, A. Paccanaro, Detecting overlapping protein complexes in protein-protein interaction networks, Nature Methods 9 (5) (2012) 471.
[3] J. Kim, M. Hastak, Social network analysis: Characteristics of online social networks after a disaster, International Journal of Information Management 38 (1) (2018) 86–96.
[4] D.M. Camacho, K.M. Collins, R.K. Powers, J.C. Costello, J.J. Collins, Next-generation machine learning for biological networks, Cell 173 (7) (2018) 1581–1592.
[5] B. Perozzi, R. Al-Rfou, 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.
[6] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, T. Eliassi-Rad, Collective classification in network data, AI Magazine 29 (3) (2008) 93.
[7] S. Bhagat, G. Cormode, S. Muthukrishnan, Node classification in social networks, in: Social Network Data Analytics, Springer, 2011, pp. 115–148.
[8] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, S. Yang, Community preserving network embedding, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI Press, 2017, pp. 203–209.
[9] I. Herman, G. Melançon, M.S. Marshall, Graph visualization and navigation in information visualization: A survey, IEEE Transactions on Visualization and Computer Graphics 6 (1) (2000) 24–43.
[10] D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks, Journal of the American Society for Information Science and Technology 58 (7) (2007) 1019–1031.
[11] M. Ou, P. Cui, J. Pei, Z. Zhang, 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.
[12] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, 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..
[13] D. Wang, P. Cui, 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.
[14] S. Cao, W. Lu, Q. Xu, Grarep, Learning graph representations with global structural information, in, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, 2015, pp. 891–900.
[15] R.A. Rossi, N.K. Ahmed, Role discovery in networks, IEEE Transactions on Knowledge and Data Engineering 27 (4) (2014) 1112–1131.
[16] K. Henderson, B. Gallagher, T. Eliassi-Rad, H. Tong, S. Basu, L. Akoglu, D. Koutra, C. Faloutsos, L. Li, Rolx: structural role extraction & mining in large graphs, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2012, pp. 1231–1239.
[17] N.K. Ahmed, R.A. Rossi, J.B. Lee, T.L. Willke, R. Zhou, X. Kong, H. Eldardiry, role2vec: Role-based network embeddings, in: Proc. DLG KDD, 2019, pp. 1–7..
[18] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788.
[19] C. Ding, X. He, H.D. Simon, On the equivalence of nonnegative matrix factorization and spectral clustering, in, in: Proceedings of the 2005 SIAM International Conference on Data Mining, SIAM, 2005, pp. 606–610.
[20] P. Cui, X. Wang, J. Pei, W. Zhu, A survey on network embedding, IEEE Transactions on Knowledge and Data Engineering 31 (5) (2018) 833–852.
[21] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119..
[22] A. Grover, 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.
[23] C. Yang, M. Sun, Z. Liu, C. Tu, Fast network embedding enhancement via high order proximity approximation, IJCAI (2017) 3894–3900.
[24] D. Zhu, P. Cui, D. Wang, W. Zhu, Deep variational network embedding in wasserstein space, in, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2018, pp. 2827–2836.
[25] D. Lian, K. Zheng, V.W. Zheng, Y. Ge, L. Cao, I.W. Tsang, X. Xie, High-order proximity preserving information network hashing, in, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1744–1753.
[26] O. Levy, Y. Goldberg, Neural word embedding as implicit matrix factorization, in: Advances in Neural Information Processing Systems, 2014, pp. 2177–2185..
[27] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, J. Tang, Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec, in, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 459–467.
[28] Z. Wang, W. Wang, G. Xue, P. Jiao, 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.

[29] X. Shi, H. Lu, Y. He, S. He, Community detection in social network with pairwisely 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.
[30] H. Jin, W. Yu, S. Li, Graph regularized nonnegative matrix tri-factorization for overlapping community detection, Physica A: Statistical Mechanics and its Applications 515 (2019) 376–387.
[31] B.-J. Sun, H. Shen, J. Gao, W. Ouyang, X. Cheng, A non-negative symmetric encoder-decoder approach for community detection, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 597–606.
[32] F. Wang, T. Li, X. Wang, S. Zhu, C. Ding, Community discovery using nonnegative matrix factorization, Data Mining and Knowledge Discovery 22 (3) (2011) 493–521.
[33] X. Shi, H. Lu, G. Jia, Adaptive overlapping community detection with bayesian nonnegative matrix factorization, in: International Conference on Database Systems for Advanced Applications, Springer, 2017, pp. 339–353.
[34] B. Chen, F. Li, S. Chen, R. Hu, L. Chen, Link prediction based on non-negative matrix factorization, PloS one 12 (8) (2017) e0182968.
[35] S. Gao, L. Denoyer, P. Gallinari, Temporal link prediction by integrating content and structure information, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, ACM, 2011, pp. 1169–1174.
[36] G. Chen, C. Xu, J. Wang, J. Feng, J. Feng, Graph regularization weighted nonnegative matrix factorization for link prediction in weighted complex network, Neurocomputing 369 (2019) 50–60.
[37] N.M. Ahmed, L. Chen, Y. Wang, B. Li, Y. Li, W. Liu, Deepeye: link prediction in dynamic networks based on non-negative matrix factorization, Big Data Mining and Analytics 1 (1) (2018) 19–33.
[38] X. Ma, P. Sun, Y. Wang, Graph regularized nonnegative matrix factorization for temporal link prediction in dynamic networks, Physica A: Statistical Mechanics and its Applications 496 (2018) 121–136.
[39] Y.-A. Lai, C.-C. Hsu, W.H. Chen, M.-Y. Yeh, S.-D. Lin, Prune: Preserving proximity and global ranking for network embedding, in: Advances in Neural Information Processing Systems, 2017, pp. 5257–5266..
[40] Z. Zhang, P. Cui, X. Wang, J. Pei, X. Yao, 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.
[41] F. Ye, C. Chen, Z. Zheng, Deep autoencoder-like nonnegative matrix factorization for community detection, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, ACM, 2018, pp. 1393–1402.
[42] K. Henderson, B. Gallagher, L. Li, L. Akoglu, T. Eliassi-Rad, H. Tong, C. Faloutsos, It's who you know: graph mining using recursive structural features, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2011, pp. 663–671.
[43] D.R. Hunter, K. Lange, A tutorial on mm algorithms, The American Statistician 58 (1) (2004) 30–37.
[44] A. Lada, G. Natalie, The political blogosphere and the 2004 us election, in: Proceedings of the 3rd International Workshop on Link Discovery, vol. 1, 2005, pp. 36–43..
[45] J. Yang, J. Leskovec, Defining and evaluating network communities based on ground-truth, Knowledge and Information Systems 42 (1) (2015) 181–213.
[46] Q. Lu, L. Getoor, Link-based text classification, in: Proceedings of the 20th International Conference on Machine Learning, 2003, pp. 496–503.
[47] T.N. Kipf, M. Welling, Variational graph auto-encoders, arXiv preprint arXiv:1611.07308..
[48] R. Jin, V.E. Lee, H. Hong, Axiomatic ranking of network role similarity, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011, pp. 922–930.