Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Community Preserving Network Embedding

Xiao Wang,[1] **Peng Cui,**[1] **Jing Wang,**[2] **Jian Pei,**[3] **Wenwu Zhu,**[1] **Shiqiang Yang**[1]
[1]Department of Computer Science and Technology, Tsinghua University, China
[2]Faculty of Science and Technology, Bournemouth University, UK
[3]School of Computing Science, Simon Fraser University, Canada
wangxiao007@mail.tsinghua.edu.cn, cuip@tsinghua.edu.cn, jwang@bournemouth.ac.uk,
jpei@cs.sfu.ca, wwzhu@tsinghua.edu.cn, yangshq@tsinghua.edu.cn

Abstract

Network embedding, aiming to learn the low-dimensional representations of nodes in networks, is of paramount importance in many real applications. One basic requirement of network embedding is to preserve the structure and inherent properties of the networks. While previous network embedding methods primarily preserve the microscopic structure, such as the first- and second-order proximities of nodes, the mesoscopic community structure, which is one of the most prominent feature of networks, is largely ignored. In this paper, we propose a novel Modularized Nonnegative Matrix Factorization (M-NMF) model to incorporate the community structure into network embedding. We exploit the consensus relationship between the representations of nodes and community structure, and then jointly optimize NMF based representation learning model and modularity based community detection model in a unified framework, which enables the learned representations of nodes to preserve both of the microscopic and community structures. We also provide efficient updating rules to infer the parameters of our model, together with the correctness and convergence guarantees. Extensive experimental results on a variety of real-world networks show the superior performance of the proposed method over the state-of-the-arts.

Introduction

Network analysis has attracted considerable attention as networks exist in various complex systems, such as biological and social systems. Network analysis heavily relies on the network representation, which is traditionally represented as discrete adjacency matrix. However, this straightforward representation usually cannot well reflect the underlying distinct structural characteristics of networks and suffers from the data sparsity issue (Perozzi, Al-Rfou, and Skiena 2014). In recent years, network embedding, i.e., learning an effective low-dimensional vector representations of nodes while preserving the network structure, has aroused considerable research interest in network analysis (Perozzi, Al-Rfou, and Skiena 2014). Benefited from this, a variety of applications on networks, such as node classification (Bhagat, Cormode, and Muthukrishnan 2011), can be directly conducted by the off-the-shelf machine learning methods in the low-dimensional vector space.

One basic requirement of network embedding is that the learned representations of nodes should preserve the network structure and its inherent properties (Ou et al. 2016). Along with this direction, some network embedding methods, e.g., IsoMap (Tenenbaum, De Silva, and Langford 2000), are proposed to preserve the first-order proximity between nodes; the second-order proximity between nodes are then considered in (Tang et al. 2015; Wang, Cui, and Zhu 2016), and (Cao, Lu, and Xu 2015) further extends to capture higher-order proximity.

Essentially, these methods mainly focus on the microscopic structure of network, i.e., the pairwise relationship or similarity between nodes. Nevertheless, the community structure, one important mesoscopic description of network structure, is largely ignored. Many networks consists of different communities with dense connections within communities but sparser connections between them (Girvan and Newman 2002). It is well recognized that community structure is one of the most prominent features of networks, which reveals the organizational structures and functional components of networks (Wang et al. 2016). Therefore, whether the learned embedding space can well reflect the community structures in the original network is a critical requirement for network embedding methods.

Moreover, different from the microscopic structure, the mesoscopic community structure imposes constraints in a higher structural level on the node representations. The representations of nodes within a community should be more similar than those belonging to different communities. Also, for two nodes within a community, even if they only have weak relationship in microscopic structure due to the data sparsity issue, their similarities will also be strengthened by the community structure constraint. Thus, the incorporation of community structure in network embedding can provide effective and rich information to solve data sparsity issues in microscopic structures and also make the learned node representations more discriminative.

In this paper, we propose a novel Modularized Nonnegative Matrix Factorization (M-NMF) model which preserves both the microscopic structure (pairwise node similarity) and mesoscopic structure (community) for network embedding. In particular, for microscopic structure, we incorporate first- and second-order proximities of nodes to learn the representations using matrix factorization; for mesoscopic

structure, the communities are detected by a modularity constraint term. Then these two terms are connected by exploiting the consensus relationship between the representations of nodes and community structure of network with an auxiliary community representation matrix, and thus they can be jointly optimized. We provide the multiplicative updating rules, as well as their correctness and convergence guarantees, to infer the parameters of M-NMF. Extensive experiments on various real networks, in comparison with several state-of-the-arts, are conducted on two network analysis tasks (node clustering and classification) to assess the performance of M-NMF.

To summarize, we make the following contributions:

- We proposed a novel Modularized Nonnegative Matrix Factorization (M-NMF) model for network embedding, which preserves both the microscopic structure (first- and second-order proximities) and mesoscopic community structure.
- We derived efficient updating rules to learn the parameters of M-NMF, and provided the theoretical analysis on their correctness and convergence.
- M-NMF was extensively evaluated on nine real networks and two network analysis tasks, which demonstrated its effectiveness and robustness to the model parameters.

Related Work

Network embedding. Several network embedding methods have been proposed recently. For example, DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) shows that the distribution of nodes appearing in short random walks is similar as the distribution of words in natural language, so it employes Skip-Gram, a word representation learning model (Mikolov et al. 2013), to learn the representations of nodes. LINE (Tang et al. 2015) first learns the representations of nodes which preserve the first- and second-order proximities, respectively, and then concatenates them as the final representations. Thereafter, GraRep defines different loss functions to capture different k-order proximities and combines the representations learned from each function (Cao, Lu, and Xu 2015). By proving DeepWalk is equivalent to matrix factorization, TADW incorporates the text information associated with each node to network embedding (Yang et al. 2015). Further, the labeling information is considered by combining the matrix factorization and the max-margin classifier (Tu et al. 2016). In order to capture the non-linear network structure, (Wang, Cui, and Zhu 2016) proposes a deep model with non-linear functions, also, the first- and second-order proximities are preserved. By using matrix factorization to approximate high-order proximity based on asymmetric transitivity, (Ou et al. 2016) preserves the asymmetric transitivity property of directed network. (Grover and Leskovec 2016) defines a flexible notion of a node's neighborhood and designs a biased random walk procedure, and then learns the representations of nodes by maximizing the likelihood of preserving network neighborhoods of nodes. All the methods above mainly focus on preserving the microscopic structure of network, while the mesoscopic community structure is ignored.

Community detection. A number of community detection methods have been proposed from different perspectives. For example, one direction is to carefully design a metric to describe the quality of community structure, such as modularity (Newman 2006b). By optimizing this metric, the community structure can be uncovered. Another idea is to utilize a generative model to describe the generation process of network. By fitting an empirical network to this model, the underlying community structure can be inferred (Karrer and Newman 2011; Jin et al. 2016). To cover all community detection methods is beyond the scope of this paper, and an elaborate review can be found in (Fortunato and Hric 2016). However, investigating the community structure in a low-dimensional vector space and establishing the cooperation between community structure and network embedding together have not been fully considered.

M-NMF Model

Consider an undirected network $G = (V, E)$ with n nodes and e edges, represented by a binary adjacency matrix $\mathbf{A} = [A_{ij}] \in \mathbb{R}^{n \times n}$, here we aim to learn the representations of nodes $\mathbf{U} \in \mathbb{R}^{n \times m}$, where m $(m \leq n)$ is the dimension of representation.

Modeling community structure. The modularity maximization based community detection method, one of the most widely used algorithms (Newman 2006a), is adopted to model the community structure. Specifically, given a network $\mathbf{A}$ with two communities, the modularity is defined as follows (Newman 2006b):

$$Q = \frac{1}{4e} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2e}) h_i h_j, \tag{1}$$

where k_i is the degree of node i and $h_i = 1$ if node i belongs to the first community, otherwise, $h_i = -1$. Notice that $\frac{k_i k_j}{2e}$ is the expected number of edges between nodes i and j if edges are placed at random, so intuitively, the modularity measures the difference between the number of edges falling within communities and the expected number in an equivalent network with edges place at random. By defining the modularity matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ whose element $B_{ij} = A_{ij} - \frac{k_i k_j}{2e}$, we have $Q = \frac{1}{4e} \mathbf{h}^T \mathbf{B} \mathbf{h}$, where $\mathbf{h} = [h_i] \in \mathbb{R}^n$ is the community membership indicator.

To extend this to $k > 2$ communities, we can generalize the community membership indicator as $\mathbf{H} \in \mathbb{R}^{n \times k}$ with one column for each community. In each row of $\mathbf{H}$, only one element is 1 and all the others are 0, so we have the constraint $tr(\mathbf{H}^T \mathbf{H}) = n$. After suppressing the constant which has no effect on the maximum of the modularity, we have

$$Q = tr(\mathbf{H}^T \mathbf{B} \mathbf{H}), \quad s.t. \quad tr(\mathbf{H}^T \mathbf{H}) = n, \tag{2}$$

where $tr(\mathbf{X})$ is the trace of matrix $\mathbf{X}$.

Modeling microscopic structure. Specifically, the first-order proximity is defined as follows (Tang et al. 2015):

Definition 1. (First-order proximity $\mathbf{S}^{(1)} = [S_{ij}^{(1)}] \in \mathbb{R}^{n \times n}$) The first-order proximity is the observed pairwise proximity between two nodes, i.e., if $A_{ij} > 0$, there exists

positive first-order proximity between nodes i and j, otherwise, the first-order proximity is 0.

Here, we consider the adjacency matrix $\mathbf{A}$ as the first-order proximity. Basically, because the first proximity is the the most direct expression of network, it is necessary to preserve the first-order proximity. It demonstrates that if two nodes have an edge, then these two nodes should be similar in the low-dimensional vector space. However, the observed edges in a real network are usually sparse. For two nodes with no edge, it does not imply these two nodes have no similarity. So it is oversimplified to compute the similarity between two nodes by taking the first-order proximity into account alone. A complementary solution is to consider their common neighbors. Intuitively, if two nodes share many neighbors, even if they do not have a direct link, they are still similar, which gives rise to the second-order proximity as follows:

Definition 2. (Second-order Proximity $\mathbf{S}^{(2)} = [S_{ij}^{(2)}] \in \mathbb{R}^{n\times n}$) Let $\mathcal{N}_i = (S_{i,1}^{(1)}, ..., S_{i,n}^{(1)})$ be the first-order proximity between node i and other nodes. Then the second-order proximity is determined by the similarity of $\mathcal{N}_i$ and $\mathcal{N}_j$.

Here we consider the cosine similarity as the second-order proximity, i.e., for nodes i and j, $S_{ij}^{(2)} = \frac{\mathcal{N}_i\mathcal{N}_j}{\|\mathcal{N}_i\|\|\mathcal{N}_j\|}$, where $\|\mathbf{X}\|$ is the norm of vector $\mathbf{X}$. In this way, the second-order proximity is between $[0, 1]$.

To preserve both of the first- and second-order proximities, we obtain the final similarity matrix using $\mathbf{S} = \mathbf{S}^{(1)} + \eta\mathbf{S}^{(2)}$, where $\eta > 0$ is the weight of the second-order proximity and we set $\eta = 5$ uniformly here. Then in the framework of NMF, we introduce a nonnegative basis matrix $\mathbf{M} \in \mathbb{R}^{n\times m}$ and a nonnegative representation matrix $\mathbf{U} \in \mathbb{R}^{n\times m}$, where m is the dimension of representation and the i-th row of $\mathbf{U}$ ($\mathbf{U}_i$) is the representation of node i. With these two matrices, we expect to approximate the similarity matrix $\mathbf{S}$, which gives rise to the following objective function:

$$\min \|\mathbf{S} - \mathbf{M}\mathbf{U}^T\|_F^2 \quad s.t. \quad \mathbf{M} \geq 0, \quad \mathbf{U} \geq 0. \tag{3}$$

Please note that our model is not limited to preserve the first- and second-order proximities of nodes. By adding additional higher-order proximity, such as third- and fourth-order proximities (Cao, Lu, and Xu 2015), to $\mathbf{S}$ in the same way, our model is able to preserve these proximities simultaneously.

The unified network embedding model. In this section, we aim to combine the above two models together so that we can incorporate the community structure to guide the learning process of representation matrix $\mathbf{U}$. To this end, we introduce an auxiliary nonnegative matrix $\mathbf{C} \in \mathbb{R}^{k\times m}$, named community representation matrix, where the r-th row ($\mathbf{C}_r$) is the representation of community r. If the representation of a node is highly similar to that of a community, the node may have a high propensity to be in this community. Formally, the propensity of node i belonging to community r can be formulated as $\mathbf{U}_i\mathbf{C}_r^T$. So if the representation of node i is orthogonal to that of community r, i.e., their representations are totally different, then this node must not be in this community. As the community indicator matrix $\mathbf{H}$ offers a guidance for all the nodes, we expect $\mathbf{U}\mathbf{C}^T$ to be as closely consistent as possible with $\mathbf{H}$. Finally, together with the objective function (2) and (3), we have the following overall objective function:

$$\begin{aligned} &\min_{\mathbf{M},\mathbf{U},\mathbf{H},\mathbf{C}} \|\mathbf{S} - \mathbf{M}\mathbf{U}^T\|_F^2 + \alpha\|\mathbf{H} - \mathbf{U}\mathbf{C}^T\|_F^2 - \beta tr(\mathbf{H}^T\mathbf{B}\mathbf{H}) \\ &s.t., \mathbf{M} \geqslant 0, \mathbf{U} \geqslant 0, \mathbf{H} \geqslant 0, \mathbf{C} \geqslant 0, tr(\mathbf{H}^T\mathbf{H}) = n, \end{aligned} \tag{4}$$

where α and β are positive parameters for adjusting the contribution of corresponding terms. As we can see, with the community representation matrix $\mathbf{C}$, we project the node representation matrix $\mathbf{U}$ into the community indicator $\mathbf{H}$. In this way, we establish the consensus relationship between them. The representations of nodes $\mathbf{U}$ are constrained by both the microscopic structure (reflected by $\mathbf{S}$ in the first term) and mesoscopic community structure (reflected by $\mathbf{H}$ obtained from the third term), so that $\mathbf{U}$ contains more structural information and becomes more discriminative.

Optimization

The objective function (4) is not convex, and we separate the optimization of (4) to four subproblems and iteratively optimize them, which guarantees each subproblem converges to the local minima.

M-subproblem: Updating $\mathbf{M}$ with other parameters in (4) fixed leads to a standard NMF formulation (Lee and Seung 2001), so the updating rule for $\mathbf{M}$ is

$$\mathbf{M} \leftarrow \mathbf{M} \odot \frac{\mathbf{S}\mathbf{U}}{\mathbf{M}\mathbf{U}^T\mathbf{U}}. \tag{5}$$

U-subproblem: Updating $\mathbf{U}$ with other parameters in (4) fixed leads to a joint NMF problem (Akata, Thurau, and Bauckhage 2011), whose updating rule is

$$\mathbf{U} \leftarrow \mathbf{U} \odot \frac{\mathbf{S}^T\mathbf{M} + \alpha\mathbf{H}\mathbf{C}}{\mathbf{U}(\mathbf{M}^T\mathbf{M} + \alpha\mathbf{C}^T\mathbf{C})}. \tag{6}$$

C-subproblem: Updating $\mathbf{C}$ with other parameters in (4) fixed also leads to a standard NMF formulation, so the updating rule of $\mathbf{C}$ is

$$\mathbf{C} \leftarrow \mathbf{C} \odot \frac{\mathbf{H}^T\mathbf{U}}{\mathbf{C}\mathbf{U}^T\mathbf{U}}. \tag{7}$$

H-subproblem: when update $\mathbf{H}$ with other parameters in (4) fixed, we need to solve the following function:

$$\begin{aligned} \min_{\mathbf{H}\geq 0} L(\mathbf{H}) =& \alpha\|\mathbf{H} - \mathbf{U}\mathbf{C}^T\|_F^2 - \beta tr(\mathbf{H}^T(\mathbf{A} - \mathbf{B_1})\mathbf{H}), \\ & s.t. \quad tr(\mathbf{H}^T\mathbf{H}) = n, \end{aligned} \tag{8}$$

where the element in $\mathbf{B}_1$ is $\frac{k_i k_j}{2e}$. Recall that $\mathbf{H}$ is the community indicator matrix, and the constraint makes the optimization of (8) an NP-hard problem. Instead, we relax the constraint to $\mathbf{H}^T\mathbf{H} = \mathbf{I}$. Finally, by introducing a regularization coefficient λ for $\mathbf{H}^T\mathbf{H} = \mathbf{I}$, we transform (8) to the following function:

$$\begin{aligned} \min_{\mathbf{H}\geq 0} L(\mathbf{H}) =& -\beta tr(\mathbf{H}^T\mathbf{A}\mathbf{H}) + \beta tr(\mathbf{H}^T\mathbf{B_1}\mathbf{H}) \\ &+ \alpha\|\mathbf{H} - \mathbf{U}\mathbf{C}^T\|_F^2 + \lambda\|\mathbf{H}^T\mathbf{H} - \mathbf{I}\|_F^2, \end{aligned} \tag{9}$$

where $\lambda > 0$ should be large enough to insure the orthogonality satisfied and we fix it as 10^9 in our experiments. We then introduce a Lagrange multiplier matrix $\mathbf{\Theta} = [\Theta_{ij}]$ for the nonnegative constraint on $\mathbf{U}$ and utilize $\|\mathbf{X}\|_F^2 = tr(\mathbf{X}^T\mathbf{X})$, resulting the following function:

$$\begin{aligned} L'(\mathbf{H}) = &- \beta tr(\mathbf{H}^T\mathbf{A}\mathbf{H}) + \beta tr(\mathbf{H}^T\mathbf{B_1}\mathbf{H}) \\ &+ \alpha tr(\mathbf{H}\mathbf{H}^T - 2\mathbf{H}\mathbf{C}\mathbf{U}^T + \mathbf{U}\mathbf{C}^T\mathbf{C}\mathbf{U}^T) \\ &+ \lambda tr(\mathbf{H}^T\mathbf{H}\mathbf{H}^T\mathbf{H} - 2\mathbf{H}^T\mathbf{H} + \mathbf{I}) + tr(\mathbf{\Theta}\mathbf{H}^T). \end{aligned} \tag{10}$$

Set derivative of $L'(\mathbf{H})$ with respect to $\mathbf{H}$ to 0, we have:

$$\begin{aligned} \mathbf{\Theta} = &2\beta\mathbf{A}\mathbf{H} - 2\beta\mathbf{B_1}\mathbf{H} - 2\alpha\mathbf{H} + 2\alpha\mathbf{U}\mathbf{C}^T \\ &- 4\lambda\mathbf{H}\mathbf{H}^T\mathbf{H} + 4\lambda\mathbf{H}. \end{aligned} \tag{11}$$

Following the Karush-Kuhn-Tucker (KKT) condition for the nonnegativity of $\mathbf{H}$, we have the following equation:

$$\begin{aligned} &(2\beta\mathbf{A}\mathbf{H} - 2\beta\mathbf{B_1}\mathbf{H} - 2\alpha\mathbf{H} + 2\alpha\mathbf{U}\mathbf{C}^T \\ &- 4\lambda\mathbf{H}\mathbf{H}^T\mathbf{H} + 4\lambda\mathbf{H})_{ij}H_{ij} = \Theta_{ij}H_{ij} = 0. \end{aligned} \tag{12}$$

This is the fixed point equation that the solution must satisfy at convergence. Given an initial value of $\mathbf{H}$, the successive updating rule of $\mathbf{H}$ is:

$$\mathbf{H} \leftarrow \mathbf{H} \odot \sqrt{\frac{-2\beta\mathbf{B_1}\mathbf{H} + \sqrt{\Delta}}{8\lambda\mathbf{H}\mathbf{H}^T\mathbf{H}}}, \tag{13}$$

where $\Delta = 2\beta(\mathbf{B_1}\mathbf{H}) \odot 2\beta(\mathbf{B_1}\mathbf{H}) + 16\lambda(\mathbf{H}\mathbf{H}^T\mathbf{H}) \odot (2\beta\mathbf{A}\mathbf{H} + 2\alpha\mathbf{U}\mathbf{C}^T + (4\lambda - 2\alpha)\mathbf{H})$.

The correctness of the updating rule (13) can be guaranteed by the following theorem.

Theorem 1. If the updating rule of $\mathbf{H}$ converges, then the final solution satisfies the KKT optimality condition.

Proof. At convergence, $\mathbf{H}^{(\infty)} = \mathbf{H}^{(t+1)} = \mathbf{H}^{(t)} = \mathbf{H}$, where t is the t-th iteration, i.e.,

$$\mathbf{H} = \mathbf{H} \odot \sqrt{\frac{-2\beta\mathbf{B_1}\mathbf{H} + \sqrt{\Delta}}{8\lambda\mathbf{H}\mathbf{H}^T\mathbf{H}}}. \tag{14}$$

Then for each H_{ij}, it is easy to check

$$\sqrt{8\lambda(\mathbf{H}\mathbf{H}^T\mathbf{H})_{ij}} = \sqrt{-2\beta(\mathbf{B_1}\mathbf{H})_{ij} + \sqrt{(\Delta)_{ij}}}, \tag{15}$$

so we have

$$[2\beta(\mathbf{B_1}\mathbf{H})_{ij} + 8\lambda(\mathbf{H}\mathbf{H}^T\mathbf{H})_{ij}]^2 = (\Delta)_{ij}. \tag{16}$$

So by some algebraic operations, we can get the following equation:

$$\begin{aligned} &2\beta(\mathbf{A}\mathbf{H})_{ij} + 2\alpha(\mathbf{U}\mathbf{C}^T)_{ij} + (4\lambda - 2\alpha)\mathbf{H}_{ij} \\ &= 4\lambda(\mathbf{H}\mathbf{H}^T\mathbf{H})_{ij} + 2\beta(\mathbf{B_1}\mathbf{H})_{ij}, \end{aligned} \tag{17}$$

which satisfies (12).□

We now prove the convergence of the updating rule. To achieve this goal, we will make use of an auxiliary function as in (Lee and Seung 2001). The definition of the auxiliary function is as follows:

Definition 3. A function $V(\mathbf{H}, \mathbf{H}')$ is an auxiliary function of function $L(\mathbf{H})$ if $V(\mathbf{H}, \mathbf{H}') \geq L(\mathbf{H}), V(\mathbf{H}, \mathbf{H}) = L(\mathbf{H})$ for any $\mathbf{H}, \mathbf{H}'$.

The auxiliary function gives rise to the following lemma (Lee and Seung 2001):

Lemma 1. If V is an auxiliary function of L, then L is nonincreasing under the updating rule $\mathbf{H}^{(t+1)} = arg\min_{\mathbf{H}} V(\mathbf{H}, \mathbf{H}^{(t)})$.

Now we show the specific form of the auxiliary function $V(\mathbf{H}, \mathbf{H}')$ for the objective function $L(\mathbf{H})$ in (9) based on lemma 2.

Lemma 2. The function

$$\begin{aligned} V(\mathbf{H}, \mathbf{H}') = &- \beta tr(\mathbf{H}'^T\mathbf{A}\mathbf{Z}) - \beta tr(\mathbf{Z}^T\mathbf{A}\mathbf{H}') - \beta tr(\mathbf{H}'^T\mathbf{A}\mathbf{H}') \\ &+ \frac{1}{2}\beta tr(\mathbf{Y}^T\mathbf{B_1}\mathbf{H}') + \frac{1}{2}\beta tr(\mathbf{H}'^T\mathbf{B_1}\mathbf{Y}) \\ &- (2\lambda - \alpha)tr(\mathbf{H}'^T\mathbf{Z}) - (2\lambda - \alpha)tr(\mathbf{Z}^T\mathbf{H}') \\ &- (2\lambda - \alpha)tr(\mathbf{H}'^T\mathbf{H}') + \lambda tr(\mathbf{R}\mathbf{H}'^T\mathbf{H}'\mathbf{H}'^T) \\ &- 2\alpha tr(\mathbf{C}\mathbf{U}^T\mathbf{Z}) - 2\alpha tr(\mathbf{C}\mathbf{U}^T\mathbf{H}') \end{aligned} \tag{18}$$

is an auxiliary function for $L(\mathbf{H})$ in (9), where $R_{ij} = \frac{H_{ij}^4}{H_{ij}'^3}$, $Z_{ij} = H'_{ij} \ln \frac{H_{ij}}{H'_{ij}}$ and $Y_{ij} = \frac{H_{ij}^2}{H'_{ij}}$.

Proof. The function $L(\mathbf{H})$ in (9) is equivalent to the function $L'(\mathbf{H})$ in (10) without the last term.

By lemma 4 in (Wang et al. 2011), we have

$$\begin{aligned} -\beta tr(\mathbf{H}^T\mathbf{A}\mathbf{H}) \leq &- \beta tr(\mathbf{H}'^T\mathbf{A}\mathbf{Z}) - \beta tr(\mathbf{Z}^T\mathbf{A}\mathbf{H}') \\ &- \beta tr(\mathbf{H}'^T\mathbf{A}\mathbf{H}'), \end{aligned} \tag{19}$$

and

$$\begin{aligned} &- (2\lambda - \alpha)tr(\mathbf{H}^T\mathbf{H}) \leq -(2\lambda - \alpha)tr(\mathbf{H}'^T\mathbf{Z}) \\ &- (2\lambda - \alpha)tr(\mathbf{Z}^T\mathbf{H}') - (2\lambda - \alpha)tr(\mathbf{H}'^T\mathbf{H}'). \end{aligned} \tag{20}$$

By lemma 6 in (Wang et al. 2011), we have

$$\beta tr(\mathbf{H}^T\mathbf{B_1}\mathbf{H}) \leq \frac{1}{2}\beta tr(\mathbf{Y}^T\mathbf{B_1}\mathbf{H}') + \frac{1}{2}\beta tr(\mathbf{H}'^T\mathbf{B_1}\mathbf{Y}). \tag{21}$$

By lemma 2 in (Wang et al. 2011), we have

$$-2\alpha tr(\mathbf{H}\mathbf{C}\mathbf{U}^T) \leq -2\alpha tr(\mathbf{C}\mathbf{U}^T\mathbf{Z}) - 2\alpha tr(\mathbf{C}\mathbf{U}^T\mathbf{H}'). \tag{22}$$

By lemmas 6 and 7 in (Wang et al. 2011), we have

$$\lambda tr(\mathbf{H}^T\mathbf{H}\mathbf{H}^T\mathbf{H}) \leq \lambda tr(\mathbf{P}\mathbf{H}'^T\mathbf{H}') \leq \lambda tr(\mathbf{R}\mathbf{H}'^T\mathbf{H}'\mathbf{H}'^T), \tag{23}$$

where $P_{ij} = \frac{(\mathbf{H}^T\mathbf{H})_{ij}^2}{(\mathbf{H}'^T\mathbf{H}')_{ij}}$ and $R_{ij} = \frac{H_{ij}^4}{H_{ij}'^3}$.

By combining (19), (20), (21), (22) and (23), we have the final auxiliary function in lemma 2.□

Based on the lemmas 1 and 2, we can show the convergence of the updating rule for $\mathbf{H}$.

Theorem 2. The optimization problem (9) is nonincreasing under the iterative updating rule (13).

Proof. According to lemma 2, we have the specific form $V(\mathbf{H}, \mathbf{H}')$ of the auxiliary function for $L(\mathbf{H})$ in (9). We then can have the solution for $\min_{\mathbf{H}} V(\mathbf{H}, \mathbf{H}')$ by setting the derivative of $V(\mathbf{H}, \mathbf{H}')$ with respect to $\mathbf{H}$ to 0:

Table 1: Accuracy (%) of node clustering (bold numbers represent the best results).

Methods	DeepWalk	LINE1	LINE2	GraRep	Node2Vec	M-NMF0	M-NMF
Cornell	32.82	35.38	42.56	33.85	34.36	40.00	**43.05**
Texas	37.97	40.64	55.61	35.29	50.27	47.06	**63.10**
Washington	35.65	38.70	53.48	36.52	41.74	55.65	**59.57**
Wisconsin	34.34	35.09	43.77	36.60	35.47	42.64	**45.66**
Polblogs	52.68	57.38	63.88	53.42	**84.83**	72.75	82.82
Amherst	10.34	42.36	44.38	46.41	41.66	43.54	**47.25**
Hamilton	10.15	33.47	31.30	38.81	35.41	38.34	**42.49**
Mich	11.66	15.58	14.63	**35.12**	14.05	29.66	31.50
Rochester	7.94	17.88	16.86	33.80	18.00	30.35	**38.09**

Table 2: Accuracy (%) of node classification (bold numbers represent the best results).

Methods	DeepWalk	LINE1	LINE2	GraRep	Node2Vec	M-NMF0	M-NMF
Cornell	24.10	27.69	44.62	45.38	38.46	27.69	**47.18**
Texas	22.63	34.21	**73.16**	68.42	51.05	47.89	70.00
Washington	24.44	25.33	50.22	52.00	53.78	54.67	**63.56**
Wisconsin	26.15	28.46	51.54	59.62	44.62	39.62	**61.15**
Polblogs	64.77	83.02	80.87	89.60	84.03	80.20	**90.67**
Amherst	41.59	91.51	87.99	91.46	89.73	87.74	**92.00**
Hamilton	39.95	91.64	87.27	91.64	91.45	89.36	**92.92**
Mich	25.44	62.09	60.75	60.79	61.98	58.15	**62.26**
Rochester	34.78	87.04	84.23	85.47	83.65	84.28	**87.18**

$$\frac{\partial V(\mathbf{H},\mathbf{H}')}{\partial H_{ij}} = 4\lambda(\mathbf{H}'\mathbf{H}'^{T}\mathbf{H}')_{ij}H_{ij}^{4} + 2\beta(\mathbf{B_1}\mathbf{H}')_{ij}H_{ij}'^{2}H_{ij}^{2} - 2\beta(\mathbf{A}\mathbf{H}')_{ij}H_{ij}'^{4} - 2\alpha(\mathbf{U}\mathbf{C}^{T})_{ij}H_{ij}'^{4} - (4\lambda - 2\alpha)H_{ij}'^{5} = 0. \quad (24)$$

By using the root of quadratic equation and the nonnegative constraint, we can first get the updating rule for H_{ij}^2, and then the updating rule for H_{ij} is obtained, which is the same as (13). Following lemma 1, under this updating rule the objective function $L(\mathbf{H})$ of (9) will be nonincreasing.□

Complexity analysis. Bulk of the computation depends on the matrix multiplication in the updating rules. The computations of updating rules in (5), (6), (7) and (13) run in $\mathcal{O}(n^2m + nm^2), \mathcal{O}(nm^2 + n^2m + m^2k), \mathcal{O}(kmn)$ and $\mathcal{O}(n^2k + k^2n + mnk)$, respectively. Since usually $m, k \leq n$, consequently, the overall computation of M-NMF is $\mathcal{O}(n^2m + n^2k)$, which has the same order of magnitude as (5). That is to say, although we incorporate additional community information, the computation order of magnitude is not increased, compared with preserving the first- and second-order proximities only. Besides, many optimized libraries for matrix multiplication[1], such as OpenBLAS[2], are currently available to further speed up the computation.

[1]https://github.com/attractivechaos/matmul
[2]http://www.openblas.net/

Experimental evaluations

We employed the following real networks for the evaluations. The WebKB network[3] consists of 4 subnetworks with 877 webpages and 1608 edges. The subnetworks were gathered from 4 universities, i.e., Cornell, Texas, Washington and Wisconsin. Each subnetwork is divided into 5 communities. Political blog network (Polblogs)[4] (Adamic and Glance 2005) (1222 nodes, 16715 edges) is composed of blogs about US politics and the web links between them, recorded in 2005. The blogs are divided into 2 communities according to their political labels (liberal and conservative). Facebook networks (Traud, Mucha, and Porter 2012) are the facebook social networks at different universities in US. For each user, there are six pieces of metadata, and according to (Traud, Mucha, and Porter 2012), the class year is used as the ground-truth of community structure. Particularly, we used four social networks in four universities, i.e., Amherst (2021 nodes, 81492 edges, 15 communities), Hamilton (2118 nodes, 87486 edges, 15 communities), Mich (2933 nodes, 54903 edges, 13 communities) and Rochester (4145 nodes, 145305 edges, 19 communities).

We compared M-NMF against the following five network embedding algorithms: DeepWalk (Perozzi, Al-Rfou, and Skiena 2014), LINE (Tang et al. 2015), GraRep (Cao, Lu, and Xu 2015), Node2Vec (Grover and Leskovec 2016) and M-NMF0. Typically, we use LINE1 to represent the LINE preserving the first-order proximity and LINE2 to represent

[3]http://linqs.cs.umd.edu/projects/projects/lbc/
[4]http://www-personal.umich.edu/~mejn/netdata/

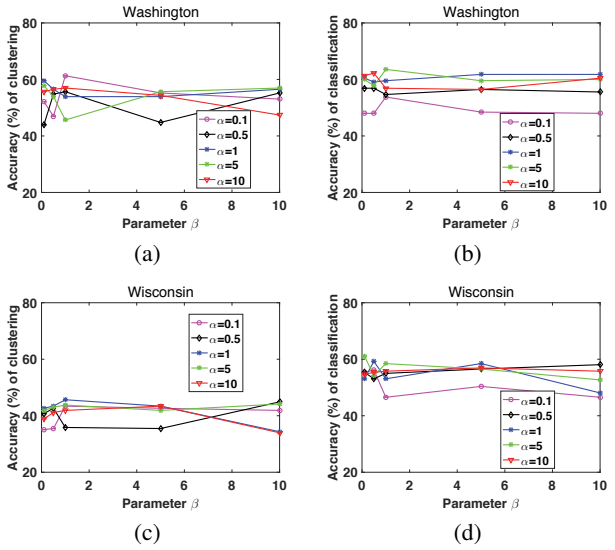

Figure 1: The effect of parameters α and β.

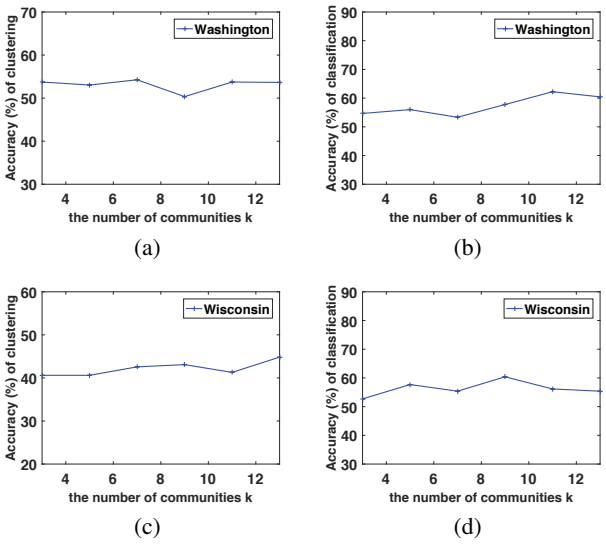

Figure 2: The effect of the number of communities k.

the LINE preserving the second-order proximity. M-NMF0 is our proposed M-NMF model which only preserves the first- and second-order proximities. M-NMF0 is used to verify the effectiveness of incorporating community structure. We uniformly set the representation dimension $m = 100$. For the M-NMF, we set α and $\beta \in \{0.1, 0.5, 1, 5, 10\}$.

Node clustering. In this section, we evaluated the performance of node clustering. We applied K-means to the learned representations of nodes and adopted accuracy (Cai et al. 2011) to assess the quality of the node clustering results. Due to the sensitivity of K-means on the initial values, we repeated the clustering 20 times, each with a new set of initial centroid, the average results were reported here, shown in Table 1. As we can see, M-NMF achieves the best results on seven of nine networks (except Polblogs and Mich). Especially on some networks, such as Texas, Hamilton and Rochester, compared with the second best results, M-NMF still achieves 4 percent to 8 percent improvement, demonstrating the superior performance of our model. Moreover, please note that M-NMF consistently shows better performance than M-NMF0 on all the tested networks, further suggesting the importance of incorporating the community structure to learn the representations of nodes.

Node classification. In this section, we verified the effectiveness of M-NMF on node classification task. The learned representations of nodes were used to classify these nodes into a set of labels. We used the LIBLINEAR package (Fan et al. 2008) to train the classifiers. For each class of a given network, we randomly selected 80% nodes as the training nodes and the rest as the testing nodes. We repeated the process 5 times and reported the average accuracy, shown in Table 2. As we can see, M-NMF outperforms the other methods on eight of nine networks (except Texas), which demonstrates the effectiveness of M-NMF on classification task. This is probably because that by utilizing the community indicator to guide the representation learning, the role of community indicator is similar as the pseudo label, so that the learned representations of nodes have more discriminative power. Besides, compared with M-NMF0, M-NMF significantly improves the accuracies, which again verifies the necessity of introducing mesoscopic community structure to network embedding.

Parameter analysis. We tested the effect of parameters α and β of M-NMF on the real networks. Because the results of different networks show similar tends, here we just used two networks (Washington and Wisconsin) as examples. We displayed the accuracies of clustering and classification with respect to α and β, respectively. As seen from Figure 1, the accuracies do not change too much and the performances are relatively stable. Also, we noticed that M-NMF on these two networks still shows competitive performances even when the accuracies are relatively low. For example, in Figure 1(a), the worst result is about 43%, better than most of the other methods.

We also tested the effect of the number of communities k, shown in Figure 2. Here we randomly selected $\alpha = 0.5$ and $\beta = 5$ and varied k from 3 to 13 with an increment of 2. As we can see, the curve of accuracies are relatively stable, indicating its robustness to the number of communities k. Overall, it is always important and an open question to accurately determine the number of communities k in a network, but our method is not very sensitive to it. In Figure 3, we can see the objective function values are nonincreasing and drop sharply within a small number of iterations (5 iterations). This empirically proves our convergence theory.

Conclusions

We developed a novel Modularized Nonnegative Matrix Factorization (M-NMF) model for network embedding, while preserving the microscopic structure (first- and second-order proximities of nodes) and the mesoscopic structure (community). NMF based learning model was

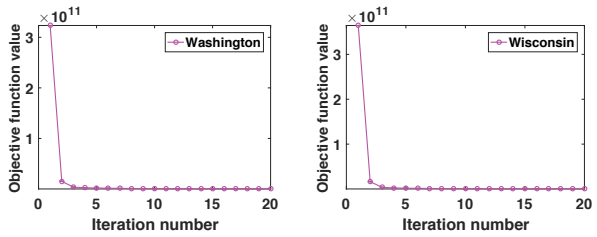

Figure 3: Convergence analysis.

used to incorporate the first- and second-order proximities, while the modularity based community detection model was adopted to detect communities. Their cooperation was established by exploiting the consensus relationship between the representations of nodes and the community structure, enabling us to jointly optimize them. The efficient updating rules with correctness and convergence guarantees were also provided. The extensive experimental results on node clustering and classification, as well as the parameter analysis, demonstrated the superiority of M-NMF.

Acknowledgments. This work was support by National Program on Key Basic Research Project, No. 2015CB352300; National Natural Science Foundation of China, No. 61370022, 61531006 and 61210008, an NSERC Discovery grant, the Canada Research Chair program, and a Yahoo! Faculty Research and Engagement Program (FREP) award. We are also grateful to the research fund of Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, and Beijing Key Laboratory of Networked Multimedia. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Adamic, L. A., and Glance, N. 2005. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, 36–43. ACM.

Akata, Z.; Thurau, C.; and Bauckhage, C. 2011. Non-negative matrix factorization in multimodality data for segmentation and label prediction. In *16th Computer vision winter workshop*.

Bhagat, S.; Cormode, G.; and Muthukrishnan, S. 2011. Node classification in social networks. In *Social network data analytics*. Springer. 115–148.

Cai, D.; He, X.; Han, J.; and Huang, T. S. 2011. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(8):1548–1560.

Cao, S.; Lu, W.; and Xu, Q. 2015. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 891–900. ACM.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *Journal of machine learning research* 9(Aug):1871–1874.

Fortunato, S., and Hric, D. 2016. Community detection in networks: A user guide. *arXiv preprint arXiv:1608.00163*.

Girvan, M., and Newman, M. E. 2002. Community structure in social and biological networks. *Proceedings of the national academy of sciences* 99(12):7821–7826.

Grover, A., and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 1225–1234. ACM.

Jin, D.; Wang, H.; Dang, J.; He, D.; and Zhang, W. 2016. Detect overlapping communities via ranking node popularities. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Karrer, B., and Newman, M. E. 2011. Stochastic blockmodels and community structure in networks. *Physical Review E* 83(1):016107.

Lee, D. D., and Seung, H. S. 2001. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, 556–562.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Newman, M. E. 2006a. Finding community structure in networks using the eigenvectors of matrices. *Physical review E* 74(3):036104.

Newman, M. E. 2006b. Modularity and community structure in networks. *Proceedings of the national academy of sciences* 103(23):8577–8582.

Ou, M.; Cui, P.; Pei, J.; Zhang, Z.; and Zhu, W. 2016. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 672–681. ACM.

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

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, 1067–1077. ACM.

Tenenbaum, J. B.; De Silva, V.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *science* 290(5500):2319–2323.

Traud, A. L.; Mucha, P. J.; and Porter, M. A. 2012. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications* 391(16):4165–4180.

Tu, C.; Zhang, W.; Liu, Z.; and Sun, M. 2016. Max-margin deepwalk: Discriminative learning of network representation. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Wang, F.; Li, T.; Wang, X.; Zhu, S.; and Ding, C. 2011. Community discovery using nonnegative matrix factorization. *Data Mining and Knowledge Discovery* 22(3):493–521.

Wang, X.; Jin, D.; Cao, X.; Yang, L.; and Zhang, W. 2016. Semantic community identification in large attribute networks. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Wang, D.; Cui, P.; and Zhu, W. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 1225–1234. ACM.

Yang, C.; Liu, Z.; Zhao, D.; Sun, M.; and Chang, E. Y. 2015. Network representation learning with rich text information. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina*, 2111–2117.