Attributed Network Embedding with Micro-Meso Structure

Juan-Hui Li[1], Chang-Dong Wang[1*], Ling Huang[1], Dong Huang[2], Jian-Huang Lai[1,3], and Pei Chen[1]

[1]School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China.
[2]College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China.
[3]XinHua College, Sun Yat-sen University, Guangzhou, China.
sysuLiJuanHui@163.com, changdongwang@hotmail.com, huanglinghl@hotmail.com, huangdonghere@gmail.com, stsljh@mail.sysu.edu.cn, chenpei@mail.sysu.edu.cn

Abstract. Recently, network embedding has received a large amount of attention in network analysis. Although some network embedding methods have been developed from different perspectives, on one hand, most of the existing methods only focus on leveraging the plain network structure, ignoring the abundant attribute information of nodes. On the other hand, for some methods integrating the attribute information, only the lower-order proximities (e.g. microscopic proximity structure) are taken into account, which may suffer if there exists the sparsity issue and the attribute information is noisy. To overcome this problem, the attribute information and mesoscopic community structure are utilized. In this paper, we propose a novel network embedding method termed Attributed Network Embedding with Micro-Meso structure (ANEM), which is capable of preserving both the attribute information and the structural information including the microscopic proximity structure and mesoscopic community structure. In particular, both the microscopic proximity structure and node attributes are factorized by Nonnegative Matrix Factorization (NMF), from which the low-dimensional node representations can be obtained. For the mesoscopic community structure, a community membership strength matrix is inferred by a generative model from the linkage structure, which is then factorized by NMF to obtain the low-dimensional node representations. The three components are jointly correlated by the low-dimensional node representations, from which an objective function can be defined. An efficient alternating optimization scheme is proposed to solve the optimization problem. Extensive experiments have been conducted to confirm the superior performance of the proposed model over the state-of-the-art network embedding methods.

Keywords: Network embedding, node attribute, microscopic proximity structure, mesoscopic community structure

[*] Corresponding author

1 Introduction

Network embedding aims to learn a low-dimensional node representation that reflects the inherent properties of a network, which plays a key role in many network analysis tasks such as visualization, node classification, link prediction, and entity retrieval [1–7]. In particular, it can well address the sparsity issue associated with network structure. Another benefit is that by transforming the topological linkage structure of network into the low-dimensional node representations, the node-interdependence is implicitly encoded into the node representations, from which both the large-scale distributed computing models (e.g., MapReduce) and off-the-shelf machine learning methods (e.g., node classification) can be directly applied.

Some recent methods propose to exploit different network structural properties to enhance network embedding, yet they mostly, if not all, ignore an important and inherent property of the network, i.e., the node attributes, which generally contains rich semantically meaningful information, such as user attributes in social network and paper titles in citation network [8, 9]. Although some attempts have been made to preserve node attributes in network embedding [10, 11], yet they attempt to infer low-dimensional node representations from the lower-order proximities, e.g., first- and second-order proximities [12–14], or the higher-order proximities [15, 16]. With only the the microscopic structure preserved, they may still suffer when the attribute is noisy and the network has the sparsity issue. To cope with this problem, Wang et al. [17] further proposed the modularized nonnegative matrix factorization (M-NMF) method to incorporate the mesoscopic community structure into network embedding, which encodes versatile organizational and functional properties of the network. However, the abundant attribute information is lost. It remains an open problem how to simultaneously incorporate the microscopic proximity structure, the mesoscopic community structure, and the node attributes into a unified and unsupervised network embedding framework.

In this paper, we propose a novel network embedding method termed Attributed Network Embedding with Micro-meso structure (ANEM) that triply preserves the microscopic proximity structure, the mesoscopic community structure and the node attributes. For the microscopic proximity structure, both the first- and second-order proximities of nodes are considered, which are summed together to form proximity matrix. To preserve the mesoscopic community structure, the recently proposed generative model termed BigCLAM [18] is used to infer the community membership strength matrix from the linkage structure. While the node attributes are characterized by a matrix with rows indicating the node ids and columns indicating the attribute dimensionality. By introducing other three matrices, these three components are factorized into the low-dimensional node representations under the framework of Nonnegative Matrix Factorization (NMF) [19], so that the learned results aggregate the information of the three components in a seamless way. NMF is used here since it generally models the generation of directly observable variables from the hidden variables [20], which coincides with our goal to learn the node representations. Under such a scheme, the three components are jointly correlated by the low-dimensional node representations, from which an objective function can be defined. An efficient alternating optimization scheme is proposed to solve the optimization problem. Extensive experiments conducted on six real-world datasets show that the proposed ANEM method outperforms most of the state-of-the-art network embedding methods in the tasks of node classification and clustering.

2 Related Work

Recently, some network embedding methods have been developed from different perspectives. Perozzi et al. [1] proposed a DeepWalk algorithm, where the truncated random walks are deployed to generate the node sequences, which are further fed into a neural language model (Skip-gram) [21] to produce the latent node embeddings. Thereafter, Tang et al. [13] proposed a large-scale information network embedding method called LINE that preserves both the first-order and second-order proximity. In [15], Cao et al. developed a GraRep model, which defines different loss functions to capture different k-step local relational information for different k, and then obtains the global representation by integrating the learned representations from each loss function. In [22], the proposed Node2Vec model generates the node sequence by balancing the breadth-first sampling and the depth-first sampling, and then learns the node representations through maximizing the likelihood of preserving network neighborhoods of nodes. To capture the highly non-linear network structure and preserve the global and local structures, Wang et al. [14] designed a Structural Deep Network Embedding (SDNE) model, where the multiple layers of non-linear functions are utilized. By preserving the asymmetric transitivity through approximating the higher-order proximity, Ou et al. [23] proposed a High-Order Proximity preserved Embedding (HOPE) model. Recently, Wang et al. [17] developed a M-NMF model which combines the mesoscopic community structure and the microscopic proximity structure simultaneously for learning low-dimensional node representations. However, most of the aforementioned methods only utilize the microscopic structure, with less methods combing both the microscopic structure and the mesoscopic community structure (e.g., [17]). In addition, rich information is lost by ignoring the valuable node attributes.

Some recent efforts have been made in integrating the node attributes information to learn the low-dimensional representations. In [24], the proposed TADW (Text-Associated DeepWalk) model is extended from DeepWalk [1] by means of combing the text features of each node via the matrix factorization. Thereafter, in [10], the proposed TriDNR model aggregates the inter-node relationships, node-content correlation, and label-content correspondence to learn the optimal node representation. In [25], the network embedding is aggregated from the ID embedding and the attribute embedding, both of which are learned through the multi-layer neural network. In [26], Li et al. proposed the property preserved algorithm through jointly optimizing the topology-derived objective function and the property-derived function. Unlike most of the existing unsupervised algorithms, Huang et al. [11] proposed the LANE framework by incorporating the label information into the attributed network embedding while preserving their correlations. However, most of these methods fail to capture the information of mesoscopic community structure.

Although the above methods work well in incorporating one or two of the three components (i.e., the microscopic proximity structure, mesoscopic community structure and node attributes), they fail to integrate all the information from these three components, resulting in less discriminative embeddings. To the best of our knowledge, there is still a lack of network embedding methods that can triply preserve microscopic proximity structure, the mesoscopic community structure and the node attributes in an unsupervised manner.

3 The Proposed Model

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{D})$ denote an attributed network consisting of n nodes, where $\mathcal{V}$ denotes the set of nodes, $\mathcal{E}$ denotes the edge set and $\mathbf{D} \in \mathbb{R}^{n \times m}$ denotes the node attribute matrix with m being the dimensionality of the node attributes. Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ denote the adjacency matrix. The goal is to learn the low-dimensional node representation denoted as $\mathbf{U} \in \mathbb{R}^{n \times d}$ that can comprehensively reflect the inherent properties of the attributed network, where d is the dimensionality of the representation vectors.

The main idea of ANEM is presented in Fig. 1. As shown in this figure, there is an attributed network consisting of 9 nodes associated with the node attributes. Similar nodes in this network belong to the same community. In the attributed network embedding, three inherent components, i.e., the microscopic proximity structure, the mesoscopic community structure and the node attributes, are preserved by NMF in a unified model, which is solved by the alternating optimization scheme to obtain the embedding representations $\mathbf{U}$. In Fig. 1, nodes 1 and 2 have similar properties in the original attributed network which results in similar node representation vectors [0.53 0.27 0.64 0.5] and [0.56 0.30 0.4 0.55]. In what follows, the proposed Attributed Network Embedding with Micro-meso structure (ANEM) model will be introduced in detail.

3.1 Modeling the Microscopic Proximity Structure

For the microscopic proximity structure, both the first-order and second-order proximities [13] are considered simultaneously. Specifically, the first-order proximity is the local pairwise proximity between two nodes. It describes the proximity of two linked nodes, i.e., $A_{ij} > 0$ indicates the first-order proximity between node i and j, otherwise, their first-order proximity is 0. However, the first-order proximity information observed in real world is only a small proportion, which leads to the sparse linkage structure. For the nodes without edges, the proximity information is lost even though they are intrinsically very similar to each other. A remedy is to considering their common neighbors, i.e., the nodes sharing similar neighbors tend to be similar to each other.

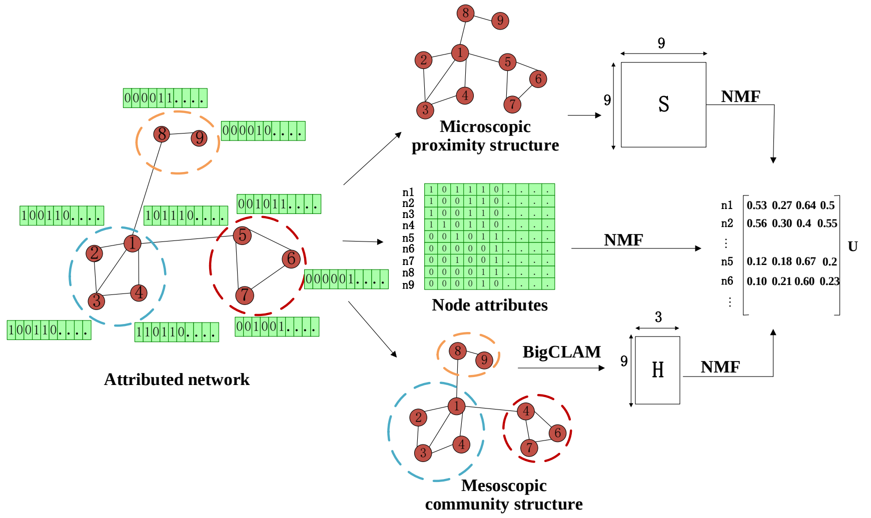


Fig. 1: Illustration of ANEM.

Formally, the first-order proximity is denoted by $\mathbf{S}^{(1)} \in \mathbb{R}^{n \times n}$, and as discussed above, we have $\mathbf{S}^{(1)} = \mathbf{A}$. Let $\mathcal{N}_i = [\mathbf{S}_{i1}^{(1)}, ..., \mathbf{S}_{in}^{(1)}]$ denote the first-order proximity of node i with other nodes. Then the second-order proximity is defined as [13]

$$\mathbf{S}_{ij}^{(2)} = \frac{\mathcal{N}_i \mathcal{N}_j}{||\mathcal{N}_i|| ||\mathcal{N}_j||}. \tag{1}$$

To preserve both the first-order proximity and the second-order proximity, the proximity matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ is defined as follows,

$$\mathbf{S} = \mathbf{S}^{(1)} + \eta \mathbf{S}^{(2)} \tag{2}$$

where η is a balancing parameter controlling the importance of the second-order proximity, and following [17], we set $\eta = 5$. Under the framework of Nonnegative Matrix Factorization (NMF) [19], the proximity matrix $\mathbf{S}$ can be decomposed into a nonnegative basis matrix $\mathbf{M} \in \mathbb{R}^{n \times d}$ and the nonnegative node representation matrix $\mathbf{U} \in \mathbb{R}^{n \times d}$ [19]:

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

3.2 Modeling the Mesoscopic Community Structure

For preserving the mesoscopic community structure, the BigCLAM [18] generative model is used to infer a community membership strength matrix, which is a recently proposed but popular method for community detection in networks. In the previous work [17], the classical modularity is used to model the community structure, which however leads to a very sparse binary membership indicator matrix. On the contrary, the BigCLAM model more generally allows generating the memberships with strengths, i.e., each community attracts its member nodes depending on the value of $\mathbf{H} \in \mathbb{R}^{n \times c}$, whose element $\mathbf{H}_{ur}$ in row $\mathbf{H}_u$ indicates the probability of node u belonging to community r. Parameter c is set according to the community number we have already known in the dataset. Specifically, to find the most likely membership matrix, the maximization of the likelihood function is designed as follows:

$$\max_{\mathbf{H}} \sum_{(u,v)\in\mathcal{E}} \log(1-\exp(-\mathbf{H}_u\mathbf{H}_v^T)) - \sum_{(u,v)\notin\mathcal{E}} \mathbf{H}_u\mathbf{H}_v^T \quad \text{s.t.} \quad \mathbf{H}\geq 0. \tag{4}$$

By introducing an auxiliary nonnegative matrix $\mathbf{C}\in\mathbb{R}^{c\times d}$, which is a community representation matrix, $\mathbf{U}_u\mathbf{C}_r^T$ can be considered as a description of the propensity that node u belongs to community r. As the membership matrix $\mathbf{H}$ encodes the probabilities of nodes belonging to the communities, $\mathbf{UC}^T$ should be as closely consistent as possible with $\mathbf{H}$. To this end, the following minimization is designed:

$$\min_{\mathbf{U},\mathbf{C}} ||\mathbf{H}-\mathbf{UC}^T||_F^2 \quad \text{s.t.} \quad \mathbf{U}\geq 0, \mathbf{C}\geq 0. \tag{5}$$

3.3 Modeling the Node Attributes

The nodes in the network contain rich semantically meaningful information, which plays a key role in network embedding [10, 11, 25, 27–29]. Let $\mathbf{D}\in\mathbb{R}^{n\times m}$ denote the node attributes, where the i-th row $\mathbf{D}_i$ is the attribute vector of node i. By introducing a nonnegative basis matrix $\mathbf{N}\in\mathbb{R}^{m\times d}$, we use the NMF framework to approximate the node attribute matrix $\mathbf{D}$, which gives rise to the following objective function:

$$\min_{\mathbf{N},\mathbf{U}} ||\mathbf{D}^T-\mathbf{NU}^T||_F^2 \quad \text{s.t.} \quad \mathbf{N}\geq 0, \mathbf{U}\geq 0. \tag{6}$$

3.4 The Overall Objective Function

As can be seen, each of the three components defined above involves the low-dimensional node representation $\mathbf{U}$. To make $\mathbf{U}$ contain the information from the microscopic proximity structure, the mesoscopic community structure and the node attributes, the consensus relationship among these three components should be established. To this end, the overall objective function is designed as follows,

$$\begin{aligned}\min_{\mathbf{M},\mathbf{U},\mathbf{H},\mathbf{C},\mathbf{N}} L &= ||\mathbf{S}-\mathbf{MU}^T||_F^2 + \alpha||\mathbf{H}-\mathbf{UC}^T||_F^2 \\ &- \beta\Bigg(\sum_{(u,v)\in\mathcal{E}} \log(1-\exp(-\mathbf{H}_u\mathbf{H}_v^T)) - \sum_{(u,v)\notin\mathcal{E}} \mathbf{H}_u\mathbf{H}_v^T\Bigg) + \gamma||\mathbf{D}^T-\mathbf{NU}^T||_F^2 \\ &\text{s.t.} \quad \mathbf{M}\geq 0, \mathbf{U}\geq 0, \mathbf{H}\geq 0, \mathbf{C}\geq 0, \mathbf{N}\geq 0\end{aligned} \tag{7}$$

where α, β, γ are the parameters that adjust the contributions of each component, the effect of which will be analyzed in our experiments. Notice that NMF is used for the three components since it generally models the generation of directly observable variables from the hidden variables [20], which coincides with our goal to learn the node representations.

4 Optimization

By using the alternating optimization scheme, the objective function Eq. (7) can be decomposed into five subproblems, i.e. updating one variable when fixing the remaining four variables.

Updating M: Updating $\mathbf{M}$ when fixing the other variables leads to the standard NMF optimization [19], which is implemented as follows,

$$\mathbf{M} \leftarrow \mathbf{M} \odot \frac{\mathbf{SU}}{\mathbf{MU}^T\mathbf{U}} \tag{8}$$

Updating U: Updating $\mathbf{U}$ when fixing the other variables leads to the joint NMF optimization [30], which is implemented as follows,

$$\mathbf{U} \leftarrow \mathbf{U} \odot \frac{\mathbf{S}^T\mathbf{M} + \alpha\mathbf{HC} + \gamma\mathbf{DN}}{\mathbf{U}(\mathbf{M}^T\mathbf{M} + \alpha\mathbf{C}^T\mathbf{C} + \gamma\mathbf{N}^T\mathbf{N})} \tag{9}$$

Updating C: Updating $\mathbf{C}$ when fixing the other variables leads to the standard NMF optimization, which is implemented as follows,

$$\mathbf{C} \leftarrow \mathbf{C} \odot \frac{\mathbf{H}^T\mathbf{U}}{\mathbf{CU}^T\mathbf{U}} \tag{10}$$

Updating N: Updating $\mathbf{N}$ when fixing the other variables leads to the standard NMF optimization, which is implemented as follows,

$$\mathbf{N} \leftarrow \mathbf{N} \odot \frac{\mathbf{D}^T\mathbf{U}}{\mathbf{NU}^T\mathbf{U}} \tag{11}$$

Updating H: Updating $\mathbf{H}$ when fixing the other variables leads to the following optimization problem:

$$\begin{aligned} \min_{\mathbf{H}} \ & \alpha||\mathbf{H} - \mathbf{UC}^T||_F^2 - \beta\Big(\sum_{(u,v)\in\mathcal{E}} \log(1 - \exp(-\mathbf{H}_u\mathbf{H}_v^T)) - \sum_{(u,v)\notin\mathcal{E}} \mathbf{H}_u\mathbf{H}_v^T \Big) \\ & \text{s.t.} \quad \mathbf{H} \geq 0. \end{aligned} \tag{12}$$

By rewriting the above objective function into the following objective function w.r.t. each row of $\mathbf{H}$

$$\begin{aligned} L(\mathbf{H}_u) = \alpha||\mathbf{H}_u - (\mathbf{UC}^T)_u||^2 - \beta\Big(\sum_{(u,v)\in\mathcal{E}} \log(1 - \exp(-\mathbf{H}_u\mathbf{H}_v^T)) - \sum_{(u,v)\notin\mathcal{E}} \mathbf{H}_u\mathbf{H}_v^T \Big) \\ \text{s.t.} \quad \mathbf{H}_u \geq 0 \end{aligned} \tag{13}$$

we can obtain the gradient of the variable H_u as follows,

$$\begin{aligned}\nabla L(\mathbf{H}_u) =&2\alpha(\mathbf{H}_u - (\mathbf{UC}^T)_u) - \beta\Big(\sum_{v\in\mathcal{N}(u)} \mathbf{H}_v \frac{\exp(-\mathbf{H}_u\mathbf{H}_v^T)}{1-\exp(-\mathbf{H}_u\mathbf{H}_v^T)} - \sum_{v\notin\mathcal{N}(u)} \mathbf{H}_v\Big)\\ =&2\alpha(\mathbf{H}_u - (\mathbf{UC}^T)_u)\\ &-\beta\Big(\sum_{v\in\mathcal{N}(u)} \mathbf{H}_v \frac{\exp(-\mathbf{H}_u\mathbf{H}_v^T)}{1-\exp(-\mathbf{H}_u\mathbf{H}_v^T)} - (\sum_{v} \mathbf{H}_v - \mathbf{H}_u - \sum_{v\in\mathcal{N}(u)} \mathbf{H}_v)\Big)\\ =&2\alpha(\mathbf{H}_u - (\mathbf{UC}^T)_u)\\ &-\beta\Big(\sum_{v\in\mathcal{N}(u)} \mathbf{H}_v \frac{1}{1-\exp(-\mathbf{H}_u\mathbf{H}_v^T)} - (\sum_{v} \mathbf{H}_v - \mathbf{H}_u)\Big) \qquad (14)\end{aligned}$$

By applying the gradient descent algorithm, $\mathbf{H}_u$ can be updated as follows,

$$\mathbf{H}_u \leftarrow \mathbf{H}_u - \lambda\nabla L(\mathbf{H}_u) \qquad (15)$$

where λ is the learning rate, which is set $\lambda = 40$ uniformly in this paper.

The above five variables are randomly initialized and then iteratively updated until the number of iterations reaches the predefined maximum number of iterations (i.e., 100 here) or the relative difference of the objective function in two adjacent iteration steps, i.e.

$$\left|\frac{L \text{ in the current iteration} - L \text{ in the last iteration}}{L \text{ in the last iteration}}\right| \qquad (16)$$

is smaller than 10^{-3}.

4.1 Complexity Analysis

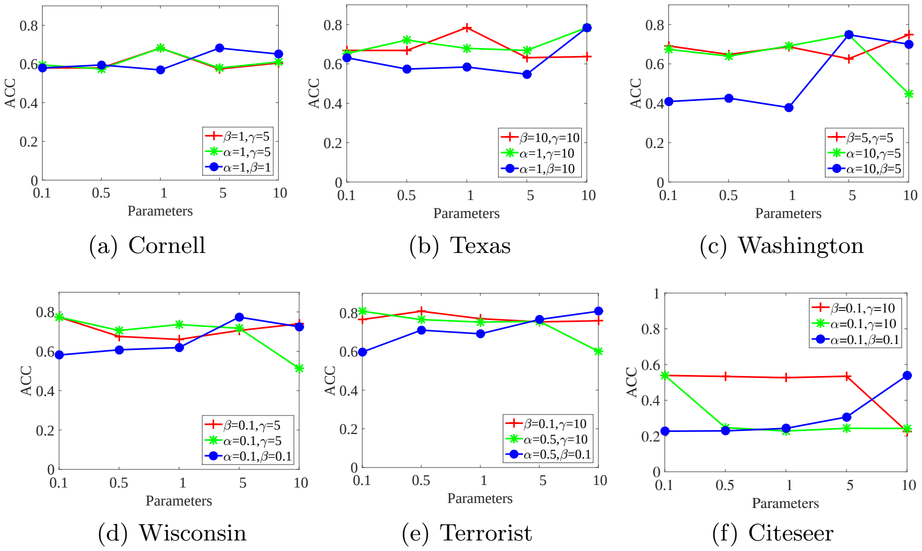
The time complexity of ANEM mainly depends on the matrix computation in the update procedure, namely the updating rules in Eqs. (8), (9), (10), (11) and (15). For these five equations, by introducing the average number of the neighbors of each node denoted by p, the computational time complexities are $O(n^2d+d^2n)$, $O(n^2d+ncd+nmd+d^2n+d^2c+d^2m)$, $O(cnd+d^2n)$, $O(nmd+d^2n)$ and $O(n^2dc + npc + n^2c)$, respectively. Since in most cases, $d, c < n$, the major computation of ANEM is in Eq. (15). Therefore, the overall computational time complexity of ANEM is $O(n^2dc)$.

5 Experiments

5.1 Experimental Settings

Six publicly available networks with node attributes are used in our experiments, which are four subnetworks from the WebKB network[1], the Terrorist Attack network [31], and the Citeseer network [32].

[1] http://www.cs.cmu.edu/~webkb/



(a) Cornell

(b) Texas

(c) Washington

(d) Wisconsin

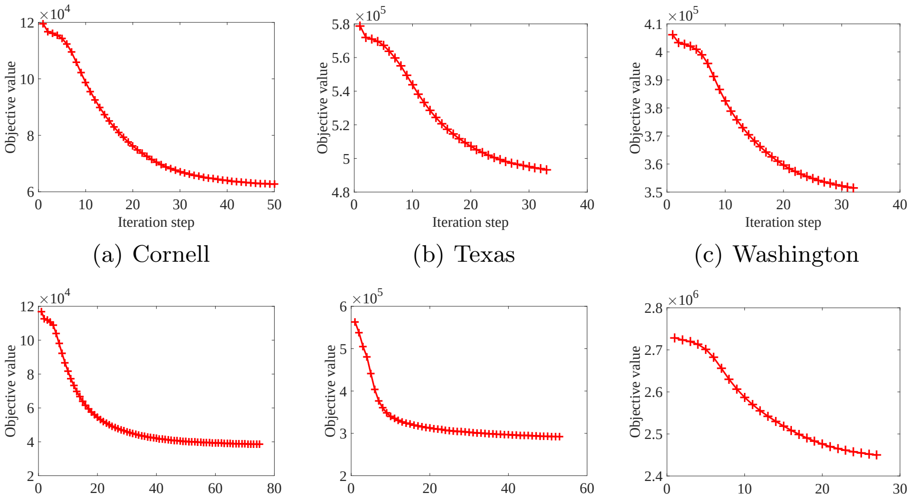
(e) Terrorist

(f) Citeseer

Fig. 2: Parameter analysis of α, β, γ: the classification task in terms of ACC.

1. **WebKB:** This dataset contains the hypertexts collected from various universities, i.e. Cornell, Texas, Washington and Wisconsin, each taken as a subnetwork. It consists of 877 webpages and 1608 edges, where Cornell has 195 webpages and 304 edges, Texas has 187 webpages and 328 edges, Washington has 230 webpages and 446 edges, and Wisconsin has 265 webpages and 530 edges. Each webpage is associated with a 1703-dimensional attribute vector. Moreover, each subnetwork is divided into 5 communities according to the following labels: Course, Student, Faculty, Project and Staff.
2. **Terrorist Attack:** This dataset is the affiliation network classified into 6 communities from the Profile in Terror project. It consists of 1293 terror attacks and 3172 links. Each attack is associated with 106-dimensional 0-1 vector indicating the attributes that are present and the attributes that are absent. We briefly use Terrorist to represent this dataset in the following paper.
3. **Citeseer:** This dataset is a citation network consisting of 3312 scientific publications and 4732 links, where each publication is associated with a 3703-dimensional 0-1 word feature vector. The publications are from distinct research areas which are divided into 6 classes.

The performance of network embedding is evaluated on the tasks of both node classification and clustering where the k-nearest neighbor (KNN) classifier and the spectral clustering based on the normalized cut are used respectively. In the network embedding, the default value of the dimensionality d is set to be 100. In the KNN classification task, k is set to be 3, 4, 5, 6, 7, and 80% of the learned node representations are randomly selected as the training data with class labels, with the remaining nodes as the testing data without class labels. In the clustering task, the similarity matrix is built from the k-nearest neighbor



(a) Cornell (b) Texas (c) Washington

(d) Wisconsin (e) Terrorist (f) Citeseer

Fig. 3: Convergence analysis on the six datasets.

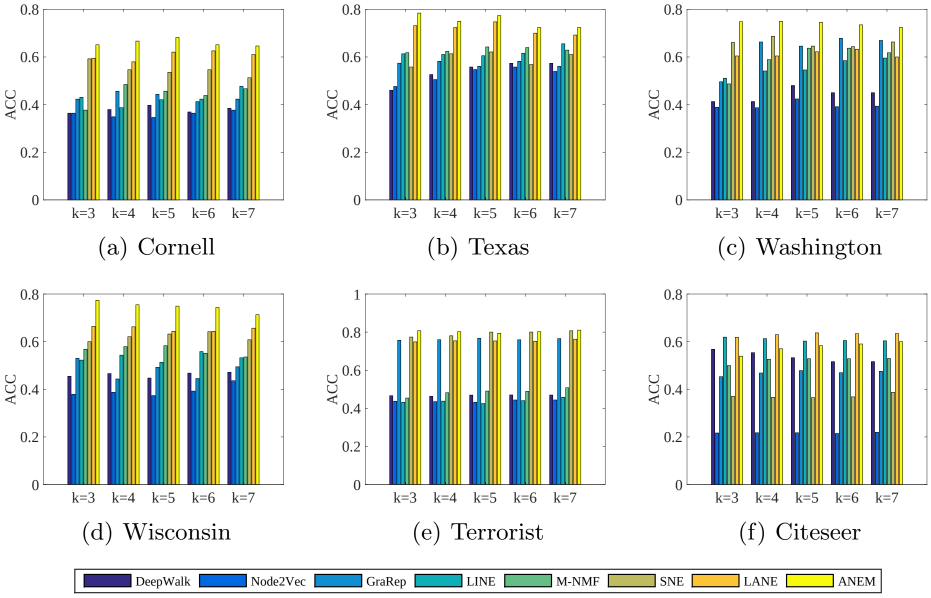
graph [33] with k set to be 3, 4, 5, 6, 7. The classification results and the clustering results are evaluated by comparing the obtained class labels with the ground-truth class labels, in terms of both accuracy (ACC) [34] and Purity. Higher ACC and Purity values indicate better classification and clustering performance.

5.2 Parameter Analysis

We first analyze the impact of the parameters α, β, γ on the performance of our ANEM model in terms of ACC with each parameter varying from {0.1, 0.5, 1, 5, 10}. Fig. 2 shows the ACC results by varying one parameter when fixing the other two parameters. Taking the results shown in Fig. 2(b) as an example, when fixing the values of parameters $\beta = 10$ and $\gamma = 10$, ANEM has achieved the best result when $\alpha = 1$ on the Texas dataset. Similarly, when $\beta = 10$ and $\gamma = 10$, the corresponding best results are achieved. Therefore, the best values of the three parameters are $\{\alpha = 1, \beta = 10, \gamma = 10\}$ for the Texas dataset. For the remaining datasets, the values of α, β, γ are determined using the same evaluation strategy as Texas. The reason for using the different values of α, β, γ for different datasets is due to the diverse impact of the microscopic proximity structure, the mesoscopic community structure, and the node attributes in different networks. Hence, in the following experiments, the values of α, β, γ are set when the best result is obtained on each dataset separately.

5.3 Convergence Analysis

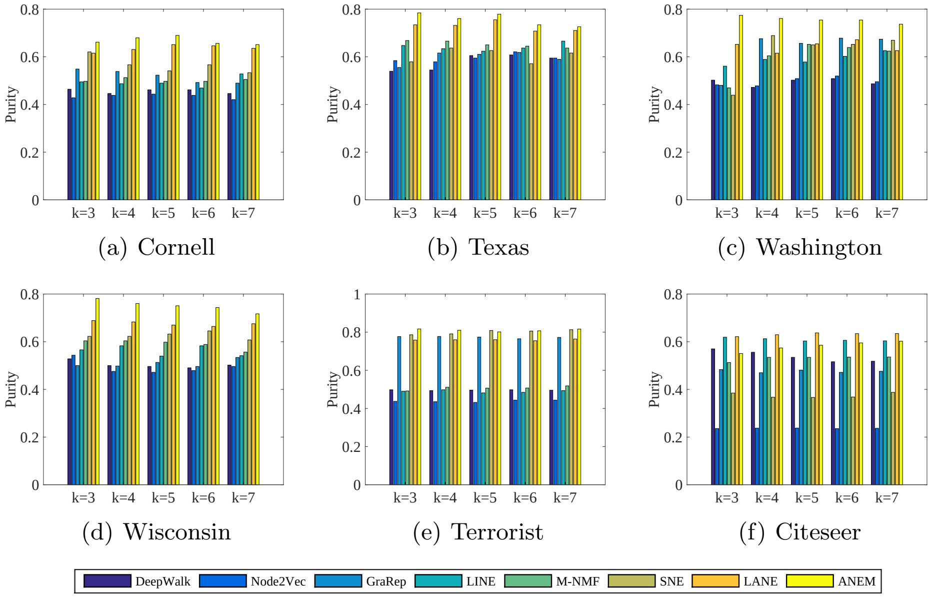
To evaluate the convergence property of our model, the objective value as a function of the iteration step on each of the six datasets is reported in Fig. 3. In this figure, we can see that on most of the datasets, the algorithm tends to



(a) Cornell (b) Texas (c) Washington

(d) Wisconsin (e) Terrorist (f) Citeseer

Fig. 4: Comparison results: the classification task in terms of ACC.

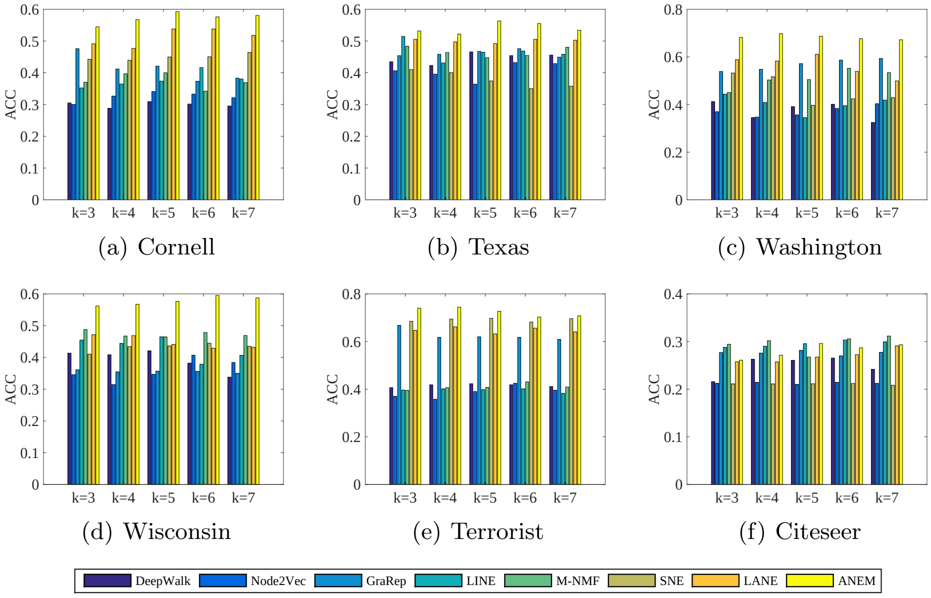


(a) Cornell (b) Texas (c) Washington

(d) Wisconsin (e) Terrorist (f) Citeseer

Fig. 5: Comparison results: the classification task in terms of Purity.

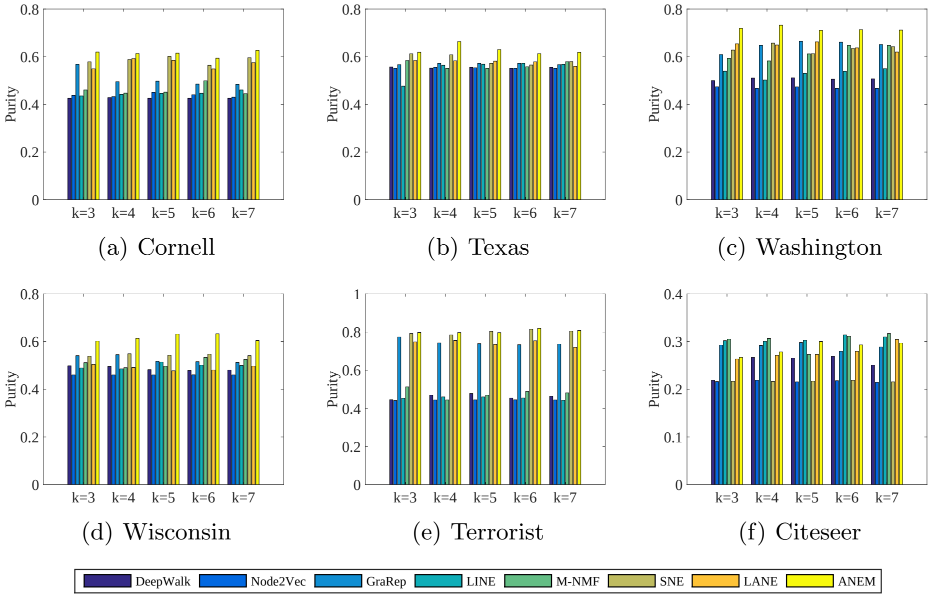
converge when the iteration number is larger than 20, and converges to a stable value when the iteration number reaches 30. In particular, on the Citeseer dataset



(a) Cornell (b) Texas (c) Washington

(d) Wisconsin (e) Terrorist (f) Citeseer

Fig. 6: Comparison results: the clustering task in terms of ACC.



(a) Cornell (b) Texas (c) Washington

(d) Wisconsin (e) Terrorist (f) Citeseer

Fig. 7: Comparison results: the clustering task in terms of Purity.

as shown in Fig. 3(f), the algorithm converges relatively faster and reaches the stable state when the iteration number is larger than 20.

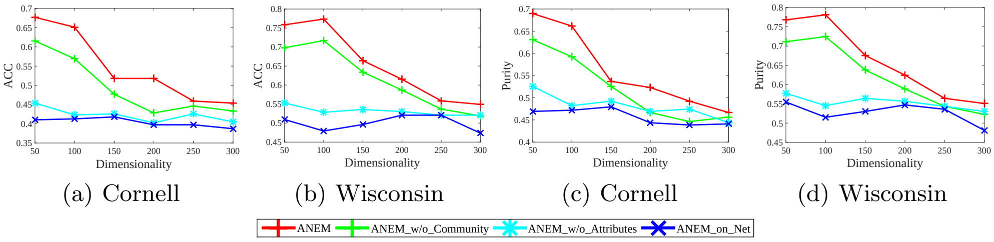
5.4 Comparison Results with the Existing Algorithms

In this section, comparison experiments are conducted to compare the performance of our model with seven state-of-the-art network embedding algorithms.

1. DeepWalk [1]: It combines the truncated random walks based on the topological structure and the Skip-gram to produce the node embeddings.
2. Node2Vec [22]: It preserves the neighborhoods of nodes based on the topological structure to generate node embeddings.
3. GraRep [15]: It utilizes the topological structure to define different loss functions and learns the node embeddings from the loss functions.
4. LINE [13]: The first-order and second-order proximities of the topological structure are both preserved to learn the node embeddings.
5. M-NMF [17]: Both the microscopic topological structure and the mesoscopic community structure are preserved to learn the node embeddings.
6. LANE [11]: The information of the network topology, node attributes and the labels are preserved. In our experiment, the version without the label information is used.
7. SNE [25]: The information of the network topology, node attributes are preserved through utilizing the multi-layer neural network.

All the codes of the above methods are obtained from the authors' websites. The parameters for these seven compared algorithms are set in such a way that either the default settings suggested by the authors are utilized or they are tuned by trials to find the best settings. And the dimensionality d is set in such a way that either 100 or the default settings suggested by the authors. After applying these network embedding algorithms, low-dimensional node representations can be obtained respectively. In the classification task, the node representations are fed into the KNN classifier with 80% randomly selected nodes as the training data while the remaining 20% as the testing data. The cross-validation process is repeated 10 times and the mean values of ACC and Purity are reported in Fig. 4 and Fig. 5 respectively. In the clustering task, the node representations are used to build the k-nearest neighbor graph as the similarity matrix where the spectral clustering is performed. Similarly, we repeat the spectral clustering 10 times, and the average values of ACC and Purity are reported in Fig. 6 and Fig. 7 respectively.

Overall, compared with the existing methods, the proposed ANEM method exhibits the best performance on most of the datasets in terms of both ACC and Purity. In particular, on the Washington and Wisconsin datasets, ANEM has obtained significantly higher ACC values than the second best algorithm in both the classification and clustering tasks. Among the seven existing algorithms, the M-NMF method incorporates the community structure, while LANE and SNE both account for the node attributes. Compared with the M-NMF method, ANEM achieves 53.6% and 64.8% improvement in terms of ACC and Purity respectively on the Washington dataset in the 3nn classification task, and 38.7% and 25.8% in the spectral clustering task when $k = 4$. For the two algorithms which consider the node attributes, i.e., the LANE and SNE methods, LANE



(a) Cornell (b) Wisconsin (c) Cornell (d) Wisconsin

Fig. 8: Comparison results of ANEM and its variations: the classification task in terms of ACC and Purity on the Cornell and Wisconsin datasets.

shows relatively better performance on most of the datasets. Compared with the LANE method, ANEM achieves 16.5% and 13.4% improvement in terms of ACC and Purity respectively on the Wisconsin dataset in the 3nn classification task, and 21.1% and 25.0% in the spectral clustering task when $k = 4$. Overall, the comparison results have demonstrated the superior performance of our model, i.e., confirming the effectiveness of incorporating all the three components.

5.5 Comparison Results with its Variations

To investigate the effectiveness of incorporating the community structure and node attributes, we compare network embedding performance of ANEM and its variations, i.e., ANEM_w/o_Community, ANEM_w/o_Attributes and ANEM_on_Net on the Cornell and the Wisconsin dataset, by means of the KNN classification. The first two variations omit the information of the community structure and node attributes respectively, while the third one only leverages the network structure. The k in the KNN classification is fixed to 3, and the dimensionality d of the node representations varies from $\{50, 100, 150, 200, 250, 300\}$. The average ACC and Purity values of 10 times cross-validation process are presented in Fig. 8.

Experimental results in Fig. 8 show that without the community structure and node attributes, ANEM_on_Net achieves the worst result, which confirms the effectiveness of preserving the community structure and node attributes information in network embedding. As d becomes larger, the performance of ANEM and its variations degenerates. When $d > 150$, ANEM_w/o_Attributes has similar performance as ANEM_on_Net. On both of the Cornell and Wisconsin datasets, ANEM_w/o_Community achieves higher ACC and Purity values than the ANEM_w/o_Attributes, which demonstrates the more significant impact of preserving the node attributes. Overall, the proposed ANEM method generates the best results compared with all its variations, further confirming the necessity of incorporating all the three components.

6 Conclusions

Network embedding has attracted an increasing amount of attention in recent years. However, it remains an open challenge to incorporate the microscopic proximity structure, the mesoscopic community structure and the node attributes for the network embedding. To this end, we developed a novel Attributed Network Embedding with Micro-meso structure (ANEM) method, triply preserving the first- and second-order proximities, community membership strength matrix generated by BigCLAM from linkage structure and the information about node attributes. By jointly correlating these three components, an overall objective function is designed, leading to an alternating optimization under the NMF framework. Extensive experiments conducted on the six publicly available attributed networks show that ANEM achieves superior performance on both of the node classification and clustering tasks over the state-of-the-art network embedding methods.

7 Acknowledgments

This work was supported by NSFC (61502543 & 61602189), Guangdong Natural Science Funds for Distinguished Young Scholar (2016A030306014), the PhD Start-up Fund of Natural Science Foundation of Guangdong Province, China (2016A030310457), and Tip-top Scientific and Technical Innovative Youth Talents of Guangdong special support program (2016TQ03X542).

References

1. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: KDD. (2014) 701–710
2. Yang, D., Wang, S., Li, C., Zhang, X., Li, Z.: From properties to links: Deep network embedding on incomplete graphs. In: CIKM. (2017) 367–376
3. Li, C., Li, Z., Wang, S., Yang, Y., Zhang, X., Zhou, J.: Semi-supervised network embedding. In: DASFAA, Springer (2017) 131–147
4. Li, H., Wang, H., Yang, Z., Odagaki, M.: Variation autoencoder based network representation learning for classification. In: Proceedings of ACL 2017, Student Research Workshop. (2017) 56–61
5. Cavallari, S., Zheng, V.W., Cai, H., Chang, K.C.C., Cambria, E.: Learning community embedding with community detection and node embedding on graphs. In: CIKM. (2017)
6. Lai, Y.A., Hsu, C.C., Chen, W.H., Yeh, M.Y., Lin, S.D.: Preserving proximity and global ranking for node embedding. In: NIPS. (2017) 5261–5270
7. Wang, H., Zhang, F., Hou, M., Xie, X., Guo, M., Liu, Q.: SHINE: Signed heterogeneous information network embedding for sentiment link prediction. In: WSDM. (2018)
8. Burger, J.D., Henderson, J., Kim, G., Zarrella, G.: Discriminating gender on twitter. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. (2011) 1301–1309
9. Pennacchiotti, M., Popescu, A.M.: Democrats, republicans and starbucks afficionados: user classification in twitter. In: KDD. (2011) 430–438
10. Pan, S., Wu, J., Zhu, X., Zhang, C., Wang, Y.: Tri-party deep network representation. In: IJCAI. (2016) 1895–1901
11. Huang, X., Li, J., Hu, X.: Label informed attributed network embedding. In: WSDM. (2017) 731–739

12. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500) (2000) 2319–2323
13. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: WWW. (2015) 1067–1077
14. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: KDD. (2016) 1225–1234
15. Cao, S., Lu, W., Xu, Q.: GraRep: Learning graph representations with global structural information. In: CIKM. (2015) 891–900
16. Ribeiro, L.F., Saverese, P.H., Figueiredo, D.R.: struc2vec: Learning node representations from structural identity. In: KDD. (2017) 385–394
17. Wang, X., Cui, P., Wang, J., Pei, J., Zhu, W., Yang, S.: Community preserving network embedding. In: AAAI. (2017) 203–209
18. Yang, J., Leskovec, J.: Overlapping community detection at scale: a nonnegative matrix factorization approach. In: WSDM. (2013) 587–596
19. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: NIPS. (2001) 556–562
20. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755) (1999) 788
21. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
22. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: KDD. (2016) 855–864
23. Ou, M., Cui, P., Pei, J., Zhang, Z., Zhu, W.: Asymmetric transitivity preserving graph embedding. In: KDD. (2016) 1105–1114
24. Yang, C., Liu, Z., Zhao, D., Sun, M., Chang, E.Y.: Network representation learning with rich text information. In: IJCAI. (2015) 2111–2117
25. Liao, L., He, X., Zhang, H., Chua, T.S.: Attributed social network embedding. arXiv preprint arXiv:1705.04969 (2017)
26. Li, C., Wang, S., Yang, D., Li, Z., Yang, Y., Zhang, X., Zhou, J.: PPNE: Property preserving network embedding. In: DASFAA. (2017) 163–179
27. Huang, X., Li, J., Hu, X.: Accelerated attributed network embedding. In: SDM. (2017) 633–641
28. Li, J., Dani, H., Hu, X., Tang, J., Chang, Y., Liu, H.: Attributed network embedding for learning in a dynamic environment. arXiv preprint arXiv:1706.01860 (2017)
29. Huang, X., Song, Q., Li, J., Hu, X.B.: Exploring expert cognition for attributed network embedding. In: WSDM. (2018)
30. Akata, Z., Thurau, C., Bauckhage, C.: Non-negative matrix factorization in multimodality data for segmentation and label prediction. In: 16th Computer vision winter workshop. (2011)
31. Zhao, B., Sen, P., Getoor, L.: Event classification and relationship labeling in affiliation networks. In: Proceedings of the Workshop on Statistical Network Analysis (SNA) at the 23rd International Conference on Machine Learning (ICML). (2006)
32. Liu, L., Xu, L., Wangy, Z., Chen, E.: Community detection based on structure and content: A content propagation perspective. In: ICDM. (2015) 271–280
33. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and computing **17**(4) (2007) 395–416
34. Cai, D., He, X., Han, J., Huang, T.S.: Graph regularized nonnegative matrix factorization for data representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(8) (2011) 1548–1560