This article has been accepted for publication in a future issue of this conference proceedings, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/DSAA49011.2020.00047, 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)

Attributed Network Embedding with Community Preservation

Tong Huang, Lihua Zhou*, Lizhen Wang, Guowang Du
School of Information
Yunnan University
Kunming, China
huangtong@mail.ynu.edu.cn, lhzhou@ynu.edu.cn,
lzhwang@ynu.edu.cn, bingwei2642@qq.com

Kevin Lü
Brunel University
Uxbridge, UB8 3PH, UK
Kevin.lu@brunel.ac.uk

***Abstract*—Network embedding (NE) is a method that maps nodes in a network into a low-dimensional and continuous vector space while maintains inherent features of the network. Most existing algorithms for NE focus on one or two of the aspects of topological structure, node attributes or community structure information, but without integrating the three in a unified framework. In this study, we develop a deep neural network-based framework for Attributed Network Embedding with Community Preservation (ANECP), which simultaneously incorporates the topological structure, node attributes as well as community structure together to obtain the low-dimensional distributed representations of nodes in the network. The use of deep neural networks captures the underlying high non-linearity in both topology and attribute information, while the incorporation of the community structure resolves the issues of data sparsity from microscopic perspective. Consequently, the obtained node representations can preserve proximity and discriminative. We conducted experimental studies using six real-world datasets. The experimental results show that proposed ANECP has superior performance over the existing methods.**

Keywords—Network embedding, topological structure, node attribute, community structure, conditional variational autoencoder

I. INTRODUCTION

Network embedding (NE), i.e., mapping nodes in a network into a low-dimensional and continuous vector space while maintaining inherent features of the network, is a fundamental procedure of networks analysis, because a variety of downstream tasks, such as node clustering [1], node classification [2], [3], link prediction [4], and network visualization [5], can be directly carried out through the ready-made machine learning algorithms in the latent feature space. Network analysis heavily relies on the low-dimensional vector representations of nodes. Therefore, How to represent network node characteristics is a key issue in network analysis tasks, an effective strategy to obtain it in traditional methods is to use adjacency matrices (i.e., first-order proximity), which can preserve the correlation between nodes in the network and visually display the network structure. However, the adjacency matrices are usually very sparse and insufficiently described relationship between vertices. In recent years, with the rapid growth of various networks, such as communication networks, protein-protein interaction networks, and academic citation networks, NE has attracted a large amount of research interests in the community of network analysis.

However, NE is not a trivial task. Essentially, the highly non-linear structure, the proximity preservation, and the consistency and complementarity of heterogeneous information are three challenges faced by NE to obtain satisfactory embedding results [6]. To answer these three challenges, various approaches have been proposed. These existing approaches can be divided into three categories: Structure-only approaches, Structure+Attribute approaches, and Structure+Community approaches. Structure-only approaches (network embedding), such as DeepWalk [7], Node2Vec [8], and SDNE [9], utilize only the topological structure (connections amongst nodes), but ignore the rich attributed information associated with nodes, such as profiles or preferences of users in social networks and text information of the article's topic in academic citation networks for example. In fact, these informative attributes can benefit network analysis, because they can reflect and affect community structures of networks [10], [11]. Structure+Attribute approaches (attributed network embedding), such as SANE [12], PRRE [13] and NetVAE [14], incorporate node attributes with topological structure, simultaneously capturing the potential high non-linearity in both types of information. Structure-only and Structure+Attribute approaches in general, mainly concentrate on the microscopic structures of the networks, i.e., the local pairwise relationships or similarities of nodes, but neglect the important mesoscopic description of the topological structure, i.e., community structure. Community structure is a collection of node groups, in which nodes within a group are densely connected but sparsely between groups [15], indicating that nodes are more similar to each other within the same community than those belonging to different communities. As one of the most important characteristics of networks, community structure discloses the organizational structure and functional composition of a network [16]. Structure+Community approaches (community preserving network embedding), such as M-NMF [16], CNRL [17] and NECS [18], incorporate community structures into NE and which the representations of nodes are more similar to each other within the same community than those from different communities, such that the similarities amongst nodes within the same community can be strengthened, even though there are only weak relationships amongst nodes due to the data sparsity from microscopic perspective. This can be explained by an example shown in Fig. 1, where different colors of nodes represent different preferences of nodes, while dotted circles of different colors represent different communities, node v_1 is connected to its neighbor nodes v_2, v_3, v_4, v_5, so according to the network structure, the distances between v_1 and v_2, v_3, v_4, v_5 are equal in low-dimensional space, but v_1 should be closer to v_4 and v_5 than to v_2 and v_3, because v_1 has the same preference with v_4 and v_5; in addition, there is no edge connecting the node v_1 and v_6, and v_1 has different preference with v_6, thus v_6 is pushed away from v_1 in low-dimensional space according to the network structure and the node attribute, but

This article has been accepted for publication in a future issue of this conference proceedings, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/DSAA49011.2020.00047, 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)

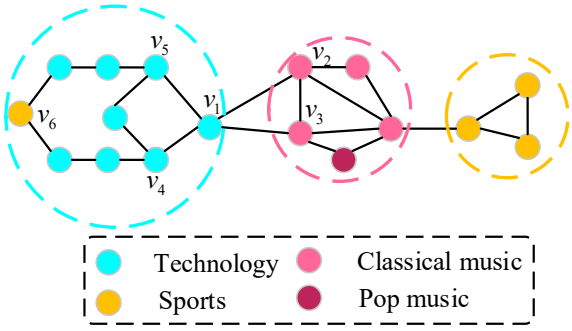

Fig. 1. An example

v_6 may close to v_1 under the consideration of community recognition. This example demonstrates that topological structure, node attributes and community structure are important information for NE.

Although both existing Structure+Attribute and Structure+Community approaches integrate two aspects in embedding, Structure+Attribute approaches do not consider community structure, while Structure+Community approaches do not take node attributes into consideration. Thus, it is more logical to examine topological structure, node attributes and community structure simultaneously to study different but complementary information.

In addition, most of existing methods, such as TADW [19] and AANE [20], only make use of the shallow structures, which are insufficient to disclose the highly non-linear property for attributed network embedding, because of the topological structure and attributes of nodes are highly non-linear [6]. Furthermore, topological structure, node attributes, and community structure describe nodes and their relationships from different views, but these three kinds of information are heterogeneous. How to preserve the consistent and complementary information in these three aspects is very important in NE.

This study aims to investigate how to simultaneously combine and incorporate the three different characteristics of networks (topological structure, node attributes and community structure) together in NE for network analysis. For this purpose, we propose ANECP framework, which combines micro-information (first- and high-order proximity) of topological structure, node attributes and meso-information (community information) to jointly learn the latent representations of nodes. In particularly, we employ a autoencoder (AE) with deep structure to maintain the first-order and high-order proximity of node attributes, use a full connected network (FCN) to extract meso-information contained in community structures, and utilize a conditional variational autoencoders (CVAE) with deep structure to protect the first-order and high-order proximity of network topology and to integrate the influence of the meso-information of community structure. The use of deep neural networks can capture the potentially high non-linearity in the topology and attribute information such that the proximity would be preserved, while the incorporation of the community structure would resolve the issues of data sparsity from microscopic perspective. Thus, the obtained node representations can encode the consistent and complementary information in the topology, attributes and community structure and preserve proximity and discriminative.

We conduct experiments using six real-world datasets and compare the results of our ANECP with that of other existing NE approaches to validate the effectiveness of ANECP via tasks of node classification, node clustering and visualization of latent representation of nodes.

The contributions included in this study are summarized as follows:

- A novel deep neural networks based approach for NE is proposed, which simultaneously incorporates the topological structure, node attributes and community structure in NE to explore three different but complementary information.
- The ANECP has been implemented, where we compute similarities amongst attributes as the input of the AE, rather than use attributes themselves as input directly, to capture the attribute global proximity. The similarities reveal correlation amongst nodes and indicate the possibility of connection of them.
- The ANECP has been evaluated via tasks of node classification, node clustering and visualization by using six real-world datasets. Experimental results indicate that the ANECP outperforms existing representative embedding methods.

The rest of the paper is arranged as follows. Section II introduces the related work briefly. The details of our approach are presented in Section III. Section IV provides experiments, results and discussion, and conclusion is presented in Section V.

II. Related Work

In this section, we review the related literature in three categories: Structure-only approaches, Structure+Attribute approaches, and Structure+Community approaches.

A. Structure-only approaches

Some earlier works such as Laplacian Eigenmaps [21] and Local Linear Embedding (LLE) [22] utilized manifold learning to capture local geometry structure. These methods are part of the dimension reduction technology and are regarded as pioneers in graph embedding technology. However, these methods cannot be extended to large networks embedding due to the high computational complexity of eigendecomposition operation. Recently, DeepWalk [7] employed truncated random walks to obtain the node sequences as local information, which treated walks as the equivalent of sentences in language models and feed the local information into the skip-gram model to get the feature representation of nodes. Node2Vec [8] further developed the weighted random walk based embedding algorithm by controlling two hyperparameter p and q to explore diverse neighborhoods. [23] proposed LINE for large scale networks, which focused on exploring first-order and second-order proximity during node representation learning process. Thereafter, GraRep [24] further extended k-order relationships to enhance network representation. However, these methods only employed the shallow model, failing to disclose the highly non-linear characteristics. SDNE [9] utilized a semi-supervised deep encoder model architecture that simultaneously preserved the first-order and second-order similarity of a network to obtain node representations. In addition, [36] proposed NetMF that to unify the negative sampling models of DeepWalk, LINE, PTE and node2vec

This article has been accepted for publication in a future issue of this conference proceedings, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/DSAA49011.2020.00047, 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)

into a closed-from matrix factorization framework, and also provide the theoretical connections between NE algorithms based on the skip-gram and the theory of graph Laplacian. VERSE [37] is proposed that to derive graph embeddings explicitly learns the distribution of a selected vertex-to-vertex similarity measure.

All these methods only utilized the topological structure information.

B. Structure+Attribute approaches

Several researchers have attempted to integrate the topology structure of a network with attributes of nodes to enhance the latent vector representations of nodes. [39] proposed a semi-supervised learning algorithm (GCN) that extending convolution neural networks into graph-structured data. [25] proposed a united framework SANE to joint optimize the topology structure and sparse attribute information. SANE utilized the attention mechanism and Continuous Bag-of-Words model (CBOW) [26] to weight the strength of interactions between nodes while learning the similarities of the topology and attribute information. [13] considered the partial correlation between topology and attributes characteristics of the same network and proposed PRRE, which learnt two thresholds through utilizing the Expectation-Maximization (EM) algorithm so that to define the difference of node relation. In order to disclose the highly non-linear property, [27] proposed a neighbor-enhanced autoencoder and attribute-aware skip-gram model (ANRL) as a unified framework for learning topological structure and node attribute information. [6] proposed DANE model that used two deep models to catch and preserve the high non-linear property and various similarities of the topology and attributes, and to guarantee the consistency and complementarity of the two heterogeneous information. [38] proposed a novel approach (GAT) that leveraging masked self-attentional layers to assign different weights to different nodes based on the features of its neighborhoods. [14] proposes a network-specific VAE method (NetVAE) for learning the embedding of topological and attribute information of a network, which considered different effects for information of topological structure and semantic information of nodes.

C. Structure+Community approaches

Structure-only and Structure+Attribute approaches consider micro-information, but the mesoscopic structure (community), one of the most distinguish characteristics of networks, has been neglected. Some researchers have studied the possibility of jointly embedding that combined the microscopic structure and mesoscopic structure into a common feature space to enhance the performance of NE. For instance, [16] proposed M-NMF based on nonnegative matrix factorization and module-based community detection to preserve the microscopic proximity information (first- and second-order proximity) and to incorporate the effect of community structures into NE. [18] proposed NECS which incorporated the community structure in node representation learning to further maintain the high-order proximity. [28] proposed a community-based variational autoencoder model, ComVAE, to combine both community information and deep learning techniques to obtain feature representations of nodes. In addition, [17] proposed a Community-enhanced Network Representation Learning (CNRL), which can simultaneously detected the community distribution of each node as well as the embeddings of both nodes and communities.

D. Community Detection approaches

In real-world networks, the nodes within same community often share common characteristics or play similar roles. Therefore, effectively identifying communities in the network and combining them with the node representation learning and jointly optimizing them can guide learning in order to obtain more discriminative node representations, and so as to be able to more effective facilitate network analysis tasks. In recent years, many community detection algorithms have been proposed to identify the communities contained in the network. For example, Infomap [30] employed information theory for encoding random walk sequence with shortest length, and through greedy search algorithm to detection community structure. [40] ranks popularity of nodes within community, and optimize a based stochastic generative model objective function through using a Bayesian approach. LEMON [41] is proposed that though performing a local spectral diffusion to find the community structure.

III. PROPOSED APPROACH

A. Notations and Problem Formulation

Let V be a set of m nodes, E be a set of edges, $\mathbf{A} \in \Re^{m \times n}$ be a matrix encoding all node attribute information, where n represents the number of attribute dimensions, and an attributed information network be denoted as $G = (V, E, \mathbf{A})$, where (V, E) denotes the topological structure. Let $\mathbf{S} \in \Re^{m \times m}$ denote the adjacency matrix of (V, E). In detail, if two nodes $v_i \in V$ and $v_j \in V$ is linked by an edge, $s_{ij} = 1$, otherwise, $s_{ij} = 0$, and then the problem of attributed network embedding with community preservation is defined as follows:

Definition 3.1 Given an attributed information network $G = (V, E, \mathbf{A})$, attributed network embedding with community preservation aims to find a mapping function $f : v_i \to \mathbf{y}_i \in \Re^d$, which can preserve the proximity of the topological structure, community structure and node attributes, and each node $v_i \in V$ can be expressed as an underlying feature vector $\mathbf{y}_i$ based on the mapping function, where $d (d \ll n)$ is the dimension of $\mathbf{y}_i$.

B. The Framework of ANECP

In this study, we develop a novel deep neural networks based approach for NE, ANECP, which combines micro-information (first-order and high-order proximity) of topology, attribute information and meso-information (community information) to jointly construct the feature representation of each node. Fig. 2 illustrates the framework of ANECP, which comprises of three components: Attribute Component, Community Component and Structure Component. The Attribute Component is composed of an AE with deep structure, aiming to extract the highly non-linearity information in node attributes, the Community Component is composed of a community detection module and a multi-layer FCN, aiming to detect community structure and encode the mesoscopic information contained in community structure, and the Structure Component is composed of a CVAE with deep structure, aiming to extract the highly non-

This article has been accepted for publication in a future issue of this conference proceedings, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/DSAA49011.2020.00047, 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)

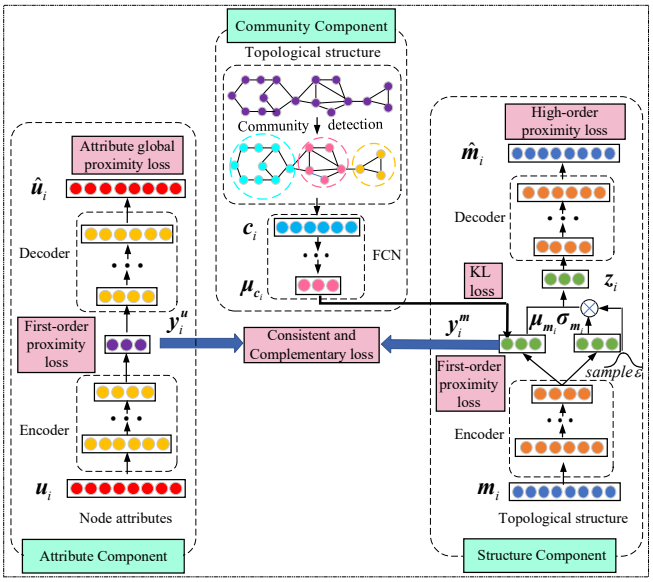

Fig. 2. The framework of ANECP

linearity information of topological structure under the constraint of the community structure.

1) The Attribute Component: In order to capture highly non-linearity information existing in node attributes, the "Attribute Component" in Fig. 2 uses an AE with deep structure, which is a powerful unsupervised deep neural network with a highly non-linearity map for feature learning, aiming to map the input to the latent low-dimensional space to disclose the high non-linearity in attributes. In addition, to capture attribute global information, we compute similarities amongst attributes as the input of the AE, rather than use attributes themselves directly.

Let $\mathbf{A} \in \Re^{m \times n}$ be the attribute matrix, $\mathbf{U} \in \Re^{m \times m}$ be the similarity matrix, $\mathbf{a}_i, \mathbf{u}_i \in \Re^m$ be the i-th row of $\mathbf{A}$ and $\mathbf{U}$ respectively, $\mathbf{a}_i$ indicate the attribute information of node $v_i \in V$, and $\mathbf{u}_i$ denote the similarity between node v_i with other nodes. $u_{ij}, j = 1,...,m$ can be computed through cosine similarity defined in Equation (1).

$$u_{ij} = \frac{\mathbf{a}_i \bullet \mathbf{a}_j}{\| \mathbf{a}_i \| \times \| \mathbf{a}_j \|} \tag{1}$$

Where $\bullet$ signifies the dot product of the two vectors, $\|\bullet\|$ indicates denotes $\ell 2$ norm, and $\times$ indicates the product of two scalars. u_{ij} measures the similarity of node v_i and node v_j, it indicates the possibility between node v_i and node v_j being connected if the attribute similarity of two nodes is regarded as an uncertain link [29].

Let $\mathbf{y}_i^{(k)} \in \Re^d$ be the hidden representation of k-th layer from the encoder, $\mathbf{y}_i^{\mathbf{u}}$ is the desired underlying compact representation of the node v_i in terms of attributes, $\hat{\mathbf{u}}_i \in \Re^m$ be the reconstructed data point from the decoder, and K be the number of layers in the encoder (correspondingly there will be K layers in the decoder), then $\mathbf{y}_i^{(k)} (k = 1,...,K)$ and $\hat{\mathbf{u}}_i$ are defined as:

$$\mathbf{y}_i^{(1)} = f(\mathbf{W}_{\mathbf{u}}^{(1)} \mathbf{u}_i + \mathbf{b}_{\mathbf{u}}^{(1)}) \tag{2}$$

$$\mathbf{y}_i^{(k)} = f(\mathbf{W}_{\mathbf{u}}^{(k)} \mathbf{y}_i^{(k-1)} + \mathbf{b}_{\mathbf{u}}^{(k)}), k = 2, \cdots, K \tag{3}$$

$$\hat{\mathbf{u}}_i = f(\mathbf{W}_{\mathbf{u}}^{(K')} \mathbf{y}_i^{(K')} + \mathbf{b}_{\mathbf{u}}^{(K')}) \tag{4}$$

Where $f(\cdot)$ represents the non-linear activation function, $\mathbf{W}_{\mathbf{u}}^{(k)}$ and $\mathbf{b}_{\mathbf{u}}^{(k)}$ denote the weight matrix and bias vector of the k-th layer. K' denote the k-th layer of the corresponding decoder. We call $\theta = \{\mathbf{W}_{\mathbf{u}}^{(k)}, \mathbf{b}_{\mathbf{u}}^{(k)}\} (k = 1,...,K)$ as the model parameters of the "Attribute Component".

The Attribute Component has two goals: (1) to minimize reconstruction error with respect to the attribute global proximity, and (2) to protect the first-order proximity of node attributes. The first-order proximity means that the more similar two nodes are, the closer their embedding should be in underlying low-dimensional space. To this end, we define the attribute global proximity loss $\mathcal{L}_r^{\mathbf{u}} = \sum_{i=1}^{m} \| \hat{\mathbf{u}}_i - \mathbf{u}_i \|_2^2$ and the first-order proximity loss $\mathcal{L}_f^{\mathbf{u}} = - \sum_{i,j,s_{ij}=1} \log p^{\mathbf{u}}(v_i, v_j)$, where $p^{\mathbf{u}}(v_i, v_j)$ is the joint probability of node v_i and node v_j, which is defined as: $p^{\mathbf{u}}(v_i, v_j) = \frac{1}{1 + \exp(-\mathbf{y}_i^{\mathbf{u}} (\mathbf{y}_j^{\mathbf{u}})^T)}$ [23], where $(\bullet)^T$ denotes transpose of a vector. The larger $p^{\mathbf{u}}(v_i, v_j)$ indicates that two nodes have more similarity with respect to attributes. Therefore, the loss function of the Attribute Component $\mathcal{L}^{\mathbf{u}}$ is defined as:

$$\mathcal{L}^{\mathbf{u}} = \mathcal{L}_r^{\mathbf{u}} + \mathcal{L}_f^{\mathbf{u}} = \sum_{i=1}^{m} \| \hat{\mathbf{u}}_i - \mathbf{u}_i \|_2^2 - \sum_{s_{ij}=1} \log p^{\mathbf{u}}(v_i, v_j) \tag{5}$$

2) The Community Component: The Community Component in Fig. 2 is composed of a community detection module and a multi-layer FCN. The community detection module may call community detection algorithms, such as Infomap [30], Fast Unfolding algorithm [31] and Label Propagation Algorithm (LPA) [32], to acquire a community structure (a collection of communities), while the multi-layer FCN is used to learn the mean of community distribution.

Suppose a network topological structure (V, E) has been divided into r communities, $\mathbf{C} \in \Re^{r \times m}$ is the membership matrix of nodes, $\mathbf{c}_i \in \Re^m$ is the i-th row of $\mathbf{C}$, and c_{ij} signifies the probability that the node v_j belongs to the i-th community. Community structure describes the structure of the network from the mesoscopic perspective. For two nodes in a community, even though there is only weak relationship between them in the micro-structure due to data sparsity, the influence between them will be strengthened by the constraints of the community structure [16].

The detected community structure is fed into the multi-layer FCN, which maps the input $\mathbf{c}_i$ to the latent low-dimensional space to capture the high non-linearity in

This article has been accepted for publication in a future issue of this conference proceedings, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/DSAA49011.2020.00047, 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)

communities. Let $\boldsymbol{\mu}_{\mathbf{c}_i}$ be the mean of community distribution, which is fed into the Structure Component to constrain the learning of topological structure.

Suppose there are K layers in the FCN, $\mathbf{y}_i^{(k)}(k=1,...,K)$ be the output of *k*-layer, then $\mathbf{y}_i^{(k)}(k=1,...,K)$ and $\boldsymbol{\mu}_{\mathbf{c}_i}$ are defined as:

$$\mathbf{y}_i^{(1)} = f(\mathbf{W}_{\mathbf{c}}^{(1)}\mathbf{c}_i + \mathbf{b}_{\mathbf{c}}^{(1)}) \tag{6}$$

$$\mathbf{y}_i^{(k)} = f(\mathbf{W}_{\mathbf{c}}^{(k)}\mathbf{y}_i^{(k-1)} + \mathbf{b}_{\mathbf{c}}^{(k)}), k = 2,\cdots,K\text{-}1 \tag{7}$$

$$\boldsymbol{\mu}_{\mathbf{c}_i} = f(\mathbf{W}_{\mathbf{c}}^{(K)}\mathbf{y}_i^{(K-1)} + \mathbf{b}_{\mathbf{c}}^{(K)}) \tag{8}$$

Where $f(\cdot)$ represents the non-linear activation function, $\mathbf{W}_{\mathbf{c}}^{(k)}$ and $\mathbf{b}_{\mathbf{c}}^{(k)}$ represent the weight matrix and bias vector of the *k*-th layer, respectively. We call $\varpi = \{\mathbf{W}_{\mathbf{c}}^{(k)}, \mathbf{b}_{\mathbf{c}}^{(k)}\}(k=1,...,K)$ as the model parameters of the Community Component.

3) The Structure Component: The Structure Component in Fig. 2 uses a CVAE to extract the highly non-linearity information of topological structure under the constraint of community structure. Variational Auto-Encoder (VAE) [33] is an unsupervised learning method based on traditional AE. Two learning objectives of VAE includes: (a) minimizing the reconstruction loss of samples; (b) minimizing the KL divergence between the encoded latent variable $\mathbf{z}$ and the standard Gaussian distribution. The difference between VAE and ordinary AE is that the encoder does not directly map the input to the latent space, but the mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$ is output for sampling the latent variables $\mathbf{z}$, and then generates $\mathbf{z}$ from the Gaussian distribution with the mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$. Finally, the sample vector $\mathbf{z}$ is fed to the decoder for generating output. Compared to other AE, VAE can preserve the distribution of data and add Gaussian noise to obtain more robust representations. CVAE [34] is an extension of VAE. It can use label information on the basis of VAE to generate data under specific label conditions.

Let $\mathbf{M} \in \Re^{m\times m}$ be the high-order proximity matrix, $\mathbf{m}_i \in \Re^m$ be the *i*-th row of $\mathbf{M}$, $\mathbf{m}_i$ signifies the high-order proximity between node v_i with other nodes respectively. $\mathbf{M}$ is defined as $\mathbf{M} = \hat{\mathbf{S}}^1 + \hat{\mathbf{S}}^2 + \cdots + \hat{\mathbf{S}}^t$ [24], where $\hat{\mathbf{S}}^t$ is the *t*-step probability transition matrix which is acquired from the row-wise normalization of the adjacency matrix $\mathbf{S}$, $\hat{\mathbf{S}}^1 = \mathbf{D}^{-1}\mathbf{S}$, $\hat{\mathbf{S}}^t = \underbrace{\hat{\mathbf{S}}^1 \cdots \hat{\mathbf{S}}^1}_{t}$, where $\mathbf{D}$ denotes the degree matrix for a graph, $\mathbf{D}_{ij} = \begin{cases} \sum_p \mathbf{S}_{ip}, & if\ i = j \\ 0, & otherwise \end{cases}$.

Let $\mathbf{y}_i^{(k)} \in \Re^d$ be the hidden representation of *k*-th layer from encoder, $\mathbf{y}_i^{\mathbf{m}}$ is the desired underlying feature representation of the node v_i in terms of the topological structure, $\hat{\mathbf{m}}_i \in \Re^m$ be the reconstructed data point from the decoder, $\boldsymbol{\mu}_{\mathbf{m}_i}$ and $\boldsymbol{\sigma}_{\mathbf{m}_i}$ be the mean and standard deviation of the input $\mathbf{m}_i$, $\mathbf{z}_i$ be the latent variable that can be sampled from a Gaussian distribution determined by $\boldsymbol{\mu}_{\mathbf{m}_i}$ and $\boldsymbol{\sigma}_{\mathbf{m}_i}$. Suppose there are K layers in the encoder and decoder respectively, $\mathbf{y}_i^{(k)}(k=1,...,K)$ be the output of k-layer, then $\mathbf{y}_i^{(k)}(k=1,...,K)$ and $\hat{\mathbf{m}}_i$ are defined as:

$$\mathbf{y}_i^{(1)} = f(\mathbf{W}_{\mathbf{m}}^{(1)}\mathbf{m}_i + \mathbf{b}_{\mathbf{m}}^{(1)}) \tag{9}$$

$$\mathbf{y}_i^{(k)} = f(\mathbf{W}_{\mathbf{m}}^{(k)}\mathbf{y}_i^{(k-1)} + \mathbf{b}_{\mathbf{m}}^{(k)}), k = 2,\cdots,K \tag{10}$$

$$\hat{\mathbf{u}}_i = f(\mathbf{W}_{\mathbf{m}}^{(K')}\mathbf{y}_i^{(K')} + \mathbf{b}_{\mathbf{m}}^{(K')}) \tag{11}$$

Where $f(\cdot)$ represents the non-linear activation function, $\mathbf{W}_{\mathbf{m}}^{(k)}$ and $\mathbf{b}_{\mathbf{m}}^{(k)}$ denote the weight matrix and bias vector of the *k*-th layer. K' represents the *k*-th layer of the corresponding decoder. We call $\vartheta = \{\mathbf{W}_{\mathbf{m}}^{(k)}, \mathbf{b}_{\mathbf{m}}^{(k)}\}(k=1,...,K)$ as the model parameters of the Structure Component.

The Structure Component has three goals: (1) to minimize reconstruction error, i.e., if two nodes share similar neighbors, they should be similar in the latent feature space, (2) to protect the first-order proximity of topological structure, i.e., if two nodes is linked by an edge, they are close in the latent feature space, and (3) to protect the global proximity of community, i.e., the nodes belonging to the same community are more closely to each other in the latent vector space. To this end, we define the high-order proximity loss $\mathcal{L}_r^{\mathbf{m}} = \sum_{i=1}^{m} \left\| \hat{\mathbf{m}}_i - \mathbf{m}_i \right\|_2^2$, the first-order proximity loss $\mathcal{L}_f^{\mathbf{m}} = -\sum_{s_{ij}=1} \log p^{\mathbf{m}}(v_i, v_j)$, where $p^{\mathbf{m}}(v_i, v_j)$ is the joint probability of two nodes v_i and v_j, which is defined as: $p^{\mathbf{m}}(v_i, v_j) = \frac{1}{1+\exp(-\mathbf{y}_i^{\mathbf{m}}(\mathbf{y}_j^{\mathbf{m}})^T)}$ [23].

In order to use community information to guide representational learning for topological structure, we control the mean of community distribution $\boldsymbol{\mu}_{\mathbf{c}_i}$ such that nodes within the same community can be embedded in close low-dimensional spaces, making latent variables $\mathbf{z}_i$ as close as possible to the community distribution. Thus, we define *KL* loss $\mathcal{L}_{KL}^{\mathbf{m}}$ as:

$$\begin{aligned} \mathcal{L}_{KL}^{\mathbf{m}} &= KL(N(\boldsymbol{\mu}_{\mathbf{m}}, \boldsymbol{\sigma}_{\mathbf{m}}^2) \,||\, N(\boldsymbol{\mu}_{\mathbf{c}}, \mathbf{I})) \\ &= \frac{1}{2}\sum_{i=1}^{\mathrm{d}} [(\boldsymbol{\mu}_{\mathbf{m}_i} - \boldsymbol{\mu}_{\mathbf{c}_i})^2 + \boldsymbol{\sigma}_{\mathbf{m}_i}^2 - \log \boldsymbol{\sigma}_{\mathbf{m}_i}^2 - 1] \end{aligned} \tag{12}$$

Therefore, the loss function of the Structure Component $\mathcal{L}^{\mathbf{m}}$ is defined as:

$$\begin{aligned} \mathcal{L}^{\mathbf{m}} &= \mathcal{L}_r^{\mathbf{m}} + \mathcal{L}_f^{\mathbf{m}} + \mathcal{L}_{KL}^{\mathbf{m}} \\ &= \sum_{i=1}^{m} \left\| \hat{\mathbf{m}}_i - \mathbf{m}_i \right\|_2^2 - \sum_{s_{ij}=1} \log p^{\mathbf{m}}(v_i, v_j) + \\ &\frac{1}{2}\sum_{i=1}^{\mathrm{d}} [(\boldsymbol{\mu}_{\mathbf{m}_i} - \boldsymbol{\mu}_{\mathbf{c}_i})^2 + \boldsymbol{\sigma}_{\mathbf{m}_i}^2 - \log \boldsymbol{\sigma}_{\mathbf{m}_i}^2 - 1] \end{aligned} \tag{13}$$

4) Consistent and Complementary: [6] pointed out that the learned low-dimensional representation from the topo-

This article has been accepted for publication in a future issue of this conference proceedings, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/DSAA49011.2020.00047, 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)

logical structure and node attributes should be consistent and complementary, because they are the two modal information of the same network, and these two kinds of information give the description of different aspects of the same node and the relationships between nodes, so they can provide complementary information. In addition, [6] also proposed the following consistent and complementary loss based on the most negative sampling strategy:

$$\mathcal{L}^{CC} = \sum_{i} \left(-\log p(v_i, v_i) - \log(1 - p(v_i, v_j)) \right) \quad (14)$$

Where $p(v_i, v_j) = \frac{1}{1 + \exp(-\mathbf{y}_i^{\mathbf{m}} (\mathbf{y}_j^{\mathbf{u}})^T)}$, v_j is the most negative sample with respect to node v_i, i.e., $j = \arg\min_{j, s_{ij}=0} \mathbf{y}_i^{\mathbf{m}} (\mathbf{y}_j^{\mathbf{u}})^T$.

Based on $\mathcal{L}^{CC}$, [6] obtained robust embedding results, thus we also adopt $\mathcal{L}^{CC}$ as our consistent and complementary loss.

In summary, to maintain microscopic first-order and high-order proximity of network topology, attribute information and mesoscopic community information, and to learn the consistency and complementarity in learning the vector representation of each node, ANECP optimize the following objective function jointly:

$$\begin{aligned} \mathcal{L} &= \alpha \mathcal{L}^{\mathbf{u}} + \beta \mathcal{L}^{\mathbf{m}} + \gamma \mathcal{L}^{CC} \\ &= \alpha \left\{ \sum_{i=1}^{m} \left\| \hat{\mathbf{u}}_i - \mathbf{u}_i \right\|_2^2 - \sum_{s_{ij}=1} \log p^{\mathbf{u}}(v_i, v_j) \right\} \\ &+ \beta \left\{ \frac{1}{2} \sum_{i=1}^{\mathrm{d}} [(\boldsymbol{\mu}_{\mathbf{m}_i} - \boldsymbol{\mu}_{\mathbf{c}_i})^2 + \boldsymbol{\sigma}_{\mathbf{m}_i}^2 - \log \boldsymbol{\sigma}_{\mathbf{m}_i}^2 - 1] \right. \\ &\left. + \sum_{i=1}^{m} \left\| \hat{\mathbf{m}}_i - \mathbf{m}_i \right\|_2^2 - \sum_{s_{ij}=1} \log p^{\mathbf{m}}(v_i, v_j) \right\} \\ &+ \gamma \left\{ \sum_{i} \left(-\log p(v_i, v_i) - \log(1 - p(v_i, v_j)) \right) \right\} \end{aligned} \quad (15)$$

Where α, β and γ are hyper-parameters, which are used as the weight of the Attribute Component, Structure Component, and the consistency and complementarity of three heterogeneous information with respect to topological structure and node attributes as well as community structure information. We concatenate $\mathbf{y}_i^{\mathbf{u}}$ and $\mathbf{y}_i^{\mathbf{m}}$ as the final representation $\mathbf{y}_i$ of node v_i.

IV. Experiments and Results

In this section, we present the performance of ANECP for representing nodes, via tasks of node classification (supervised task), node clustering (unsupervised task) and visualization of latent representation of nodes on six real-world datasets. We also evaluate the performance of ANECP through comparing with those of several baseline methods. In addition, we examine the hyper-parameter sensitivity of ANECP as well.

A. Datasets

The six real-world datasets we adopted in our experiments are extracted from WebKB networks and Facebook networks [35], where Facebook networks consist of 100 American institutions. WebKB[1] consists of Cornell (Corn), Texas (Texa), Washington (Wash), Wisconsin (Wisc) 4 subnetworks, and we used Hamilton (Hami) and Rochester[2] (Roch) two datasets from Facebook networks. In WebKB, nodes represent webpages and edges represent citation relations, while nodes act for users and edges act for friendship relations in Facebook. Each node of Cornell, Texas, Washington, Wisconsin dataset is described by attributes that are represented via a feature vector with 1703 dimensions, and all nodes are classified into courses, teachers, students, projects and staff 5 classes. Each node of Hamilton and Rochester is described by student/faculty status flag, gender, major, second major/minor, dorm/house, year, and high school 7 attributes, which are represented via a 144-dimensional and 235-dimensional feature vector respectively. We select student/faculty status flag as class label for Hamilton and Rochester two datasets, and all nodes are classified into 6 and 5 separately classes.

More detailed information of the six datasets are summarized in Table I.

TABLE I. DETAILS OF THE DATASETS

Datasets	*Number of nodes*	*Number of edges*	*Number of attributes*	*Number of classes*
Cornell	195	304	1703	5
Texas	187	328	1703	5
Washington	230	446	1703	5
Wisconsin	265	530	1703	5
Hamilton	2314	192788	144	6
Rochester	4563	322808	235	5

B. Baseline Methods

In order to evaluate the performance of our proposed ANECP, we compared it with 12 representative learning approaches, including 2 Attribute-only methods, 3 Structure-only methods, 3 Attribute+Structure methods, 3 Structure+Community methods and 1 Structure+Attribute+Community.

- **Attribute-only:** This group of baselines makes use of node attribute information only, which are used to verify the contribution of node attributes. ANECP-AF and ANECP-AS, two variants of ANECP, are used as two baseline algorithms that neglect topological structure and community information. ANECP-AF uses feature vectors of attributes as input directly, but ANECP-AS uses similarities amongst feature vectors of attributes as input. We want to know whether the attribute global information revealed via the similarities amongst feature vectors of attributes has more contribution to the attributed network embedding.
- **Structure-only:** This group of baselines utilizes topological structure information only, which are used to verify the contribution of topological structure. DeepWalk [7], Node2Vec [8] and SDNE [9] are selected as our baseline algorithms.
- **Structure+Attribute:** This group of algorithms tries to incorporate both attribute information and topological structure information simultaneously, which are used to evaluate the collective effect of

[1] https://linqs-data.soe.ucsc.edu/public/lbc/
[2] https://escience.rpi.edu/data/DA/fb100/

This article has been accepted for publication in a future issue of this conference proceedings, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/DSAA49011.2020.00047, 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)

node attributes and topological structure. We choose ANECP-TA, DANE [6] and ANRL [27] as our compared algorithms, where ANECP-TA is the simplified version of our model without incorporating community structure.

- **Structure+Community:** This group of approaches incorporates both in topological structure and community information at the same time, but ignores node attribute information. ANECP-TC, M-NMF [16] and ComVAE [28] are selected as our baseline algorithms. ANECP-TC is the reduced version of our model without introducing attribute information.
- **Structure+Attribute+Community:** ANECP and ANECP-VAE, proposed in this paper, integrates node attribute, topological structure and community information together. ANECP-VAE, one variant of ANECP, used a variational autoencoder in the "Attribute Component" and a conditional variational autoencoder in the "Structure Component", which is to verify the effectiveness of different frameworks.

For all baselines, we employ the implementation published by the original authors. The parameters for baselines are tuned to be optimal. The embedding size d is set to 128 in Cornell, Hamilton and Rochester datasets and 64 for the remaining datasets. For ComVAE, ANECP-VAE and ANECP, we use Infomap [30] to detect communities. For ANECP-TC, ANECP-VAE and ANECP, we utilize a FCN with two layers, and the frameworks of the encoder for topology and attributes of different datasets are presented in Table II. For each dataset, the first row corresponds to the topology, and the second row corresponds to node attributes. The framework of the decoder reverses that of the encoder.

TABLE II. THE FRAMEWORK OF ANECP FOR DIFFERENT DATASE

Datasets	***Number of neurons in each layer***	**Datasets**	***Number of neurons in each layer***
Cornell	195-200-100-64 195-500-100-64	Wisconsin (Wiscon)	256-128-64-32 256-128-64-32
Texas	187-128-64-32 187-128-64-32	Hamilton (Hamil)	2314-200-100-64 2314-500-100-64
Washington (Washing)	230-128-64-32 230-128-64-32	Rochester (Roches)	4563-200-100-64 4563-500-100-64

The framework of ANECP-TC is the consistent with first line of ANECP, while ANECP-AS and ANECP-AF are the same as the second row, but for ANECP-AF the number of neurons in the first layer is the dimensions of the feature vectors of attributes rather than the number of nodes. The framework of ANECP-AF is shown in Table III.

TABLE III. THE FRAMEWORK OF ANECP-AF FOR DIFFERENT DATASETS

Datasets	***Number of neurons in each layer***	**Datasets**	***Number of neurons in each layer***
Cornell	1703-500-100-64	Wiscon	1703-128-64-32
Texas	1703-128-64-32	Hamil	144-500-100-64
Washing	1703-128-64-32	Roches	235-500-100-64

TABLE IV. THE HYPER-PARAMETER SETTINGS OF ANECP FOR NODE CLASSIFICATION

Datasets	α	β	γ	**Datasets**	α	β	γ
Cornell	10	10	0.1	Wiscon	100	100	0.001
Texas	0.001	50	200	Hamil	1	1000	0.01
Washing	50	10	0.001	Roches	50	1	0.1

The hyper-parameter settings of ANECP for node classification and node clustering are shown in Table IV and Table V respectively.

TABLE V. THE HYPER-PARAMETER SETTINGS OF ANECP FOR NODE CLUSTERING

Datasets	α	β	γ	**Datasets**	α	β	γ
Cornell	500	1000	1	Wiscon	200	10	10
Texas	10	50	10	Hamil	0.1	500	10
Washing	500	10	0.001	Roches	0.01	100	0.1

C. Evaluation metrics

In this subsection, we evaluate the ability of node representation of our proposed ANECP in reconstructing the topological structure and node attributes, via tasks of node classification, node clustering and visualization. The performance of node classification is measured by *Micro-F1* (Mi-F1) and *Macro-F1* (Ma-F1) [9] metrics, and the performance of node clustering is measured by clustering Accuracy metric [6].

D. Results and Analysis

1) Node Classification: Node classification is carried out on the learned node representations and $\ell 2$-regularized Logistic Regression is used as the classifier. {10%, 30%, 50%} labeled nodes are randomly selected as the training set for training the classifier and the remained nodes as the testing set for evaluating the classifier respectively. In the experiments, five-fold cross-validation is used. This process is repeated 20 times, and the average performances with respect to both Macro-F1 and Micro-F1 are reported for each dataset. The detailed results are shown in Table VI~XI.

From Table VI~XI, we have followed four observations. First, ANECP obtains the best average performance in most situations and ANECP-VAE is second only to ANECP. Second, all Structure+Attribute methods (ANECP-TA, DANE and ANRL) perform better than Structure-only methods (DeepWalk, node2vec and SDNE) and Attribute-only methods (ANECP-AF and ANECP-AS) on almost all datasets. It shows that it is insufficient to exploit only the topological structure or the node attribute information individually. Third, Structure+Community methods (ANECP-TC, ComVAE and M-NMF) is inferior to Structure-only methods (DeepWalk, Node2Vec and SDNE)

TABLE VI. NODE CLASSIFICATION RESULTS ON SIX DATASETS

Datasets	**Method**	***10%***		***30%***		***50%***	
		Mi-F1	***Ma-F1***	***Mi-F1***	***Ma-F1***	***Mi-F1***	***Ma-F1***
Cornell	DeepWalk	0.338	0.206	0.347	0.210	0.358	0.208
	Node2Vec	0.231	0.194	0.391	0.228	0.397	0.230
	SDNE	0.369	0.225	0.423	0.171	0.428	0.229
	DANE	0.437	0.121	0.496	0.318	0.642	0.469
	ANRL	0.511	0.367	0.494	0.361	0.530	0.369
	ANECP-TA	**0.586**	**0.461**	0.669	0.526	**0.731**	0.571
	M-NMF	0.361	0.244	0.399	0.228	0.391	0.244
	ComVAE	0.242	0.204	0.391	0.183	0.368	0.189
	ANECP-TC	0.301	0.243	0.416	0.265	0.367	0.253
	ANECP-AF	0.527	0.382	0.560	0.418	0.648	0.499
	ANECP-AS	0.522	0.398	0.642	0.557	0.663	0.555
	ANECP-VAE	0.569	0.443	0.640	0.509	0.688	0.514
	ANECP	0.564	0.456	**0.691**	**0.581**	0.704	**0.579**

This article has been accepted for publication in a future issue of this conference proceedings, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/DSAA49011.2020.00047, 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)

Texas	DeepWalk	0.520	0.211	0.519	0.216	0.539	0.216
	Node2Vec	0.526	0.230	0.532	0.240	0.542	0.240
	SDNE	0.550	0.175	0.564	0.159	0.543	0.140
	DANE	0.551	0.142	0.663	0.320	0.781	0.534
	ANRL	0.532	0.258	0.587	0.291	0.542	0.140
	ANECP-TA	0.553	0.289	**0.782**	**0.524**	**0.827**	**0.586**
	M-NMF	0.553	0.142	0.557	0.143	0.542	0.140
	ComVAE	0.556	0.142	0.548	0.143	0.542	0.140
	ANECP-TC	0.538	0.196	0.557	0.143	0.542	0.140
	ANECP-AF	0.556	0.143	0.557	0.143	0.767	0.556
	ANECP-AS	0.585	0.298	0.717	0.437	0.787	0.542
	ANECP-VAE	0.586	0.312	0.699	0.447	0.789	0.543
	ANECP	**0.597**	**0.321**	0.755	0.496	0.815	0.581
Washing	DeepWalk	0.456	0.193	0.475	0.217	0.466	0.212
	Node2Vec	0.457	0.207	0.503	0.228	0.504	0.213
	SDNE	0.449	0.124	0.602	0.277	0.643	0.289
	DANE	0.449	0.124	0.677	0.338	0.771	0.537
	ANRL	0.449	0.124	0.658	0.375	0.643	0.384
	ANECP-TA	0.657	0.384	0.711	0.386	0.752	0.505
	M-NMF	0.455	0.251	0.474	0.296	0.507	0.305
	ComVAE	0.449	0.124	0.447	0.133	0.443	0.192
	ANECP-TC	0.443	0.124	0.503	0.222	0.608	0.268
	ANECP-AF	0.458	0.137	0.719	0.416	0.773	0.539
	ANECP-AS	0.468	0.149	0.757	0.507	**0.800**	0.581
	ANECP-VAE	0.669	0.379	0.746	0.505	0.770	0.541
	ANECP	**0.692**	**0.400**	**0.783**	**0.601**	0.799	**0.591**
Wiscon	DeepWalk	0.401	0.221	0.444	0.258	0.451	0.267
	Node2Vec	0.429	0.264	0.451	0.259	0.458	0.264
	SDNE	0.431	0.150	0.440	0.158	0.451	0.178
	DANE	0.573	0.329	0.758	0.501	0.797	0.590
	ANRL	0.514	0.277	0.430	0.120	0.541	0.333
	ANECP-TA	0.702	0.427	**0.794**	**0.566**	**0.833**	**0.649**
	M-NMF	0.457	0.240	0.430	0.235	0.421	0.279
	ComVAE	0.397	0.155	0.430	0.130	0.443	0.184
	ANECP-TC	0.439	0.211	0.430	0.120	0.496	0.210
	ANECP-AF	0.707	0.432	0.752	0.499	0.788	0.553
	ANECP-AS	0.627	0.370	0.779	0.495	0.774	0.536
	ANECP-VAE	0.726	0.443	0.786	0.556	0.820	0.639
	ANECP	**0.736**	**0.452**	0.777	0.537	0.831	0.621
Hamil	DeepWalk	0.906	0.285	0.915	0.290	0.918	0.292
	Node2Vec	0.917	0.293	0.922	0.295	0.924	0.296
	SDNE	0.899	0.280	0.899	0.278	0.887	0.270
	DANE	0.929	0.299	0.935	0.302	0.942	0.306
	ANRL	0.920	0.294	0.925	0.297	0.934	0.301
	ANECP-TA	0.927	0.299	0.930	0.299	0.939	0.304
	M-NMF	0.794	0.147	0.798	0.148	0.799	0.148
	ComVAE	0.797	0.147	0.800	0.148	0.804	0.148
	ANECP-TC	0.828	0.227	0.851	0.227	0.855	0.247
	ANECP-AF	0.920	0.292	0.926	0.297	0.933	0.301
	ANECP-AS	0.928	0.298	0.925	0.297	0.933	0.306
	ANECP-VAE	0.927	0.298	0.934	0.302	0.938	0.304
	ANECP	**0.941**	**0.306**	**0.938**	**0.305**	**0.948**	**0.311**
Roches	DeepWalk	0.860	0.288	0.870	0.319	0.872	0.329
	Node2Vec	0.852	0.265	0.867	0.307	0.874	0.338
	SDNE	0.850	0.239	0.854	0.244	0.862	0.249
	DANE	0.889	**0.328**	0.893	0.305	0.902	0.331
	ANRL	0.884	0.284	0.882	0.273	0.902	0.331
	ANECP-TA	0.892	0.312	0.894	0.303	0.906	0.329
	M-NMF	0.815	0.149	0.813	0.149	0.817	0.149
	ComVAE	0.811	0.149	0.811	0.149	0.817	0.149
	ANECP-TC	0.811	0.149	0.830	0.238	0.839	0.262
	ANECP-AF	0.882	0.290	0.876	0.268	0.888	0.316
	ANECP-AS	0.889	0.310	0.888	0.279	0.896	0.296
	ANECP-VAE	0.881	0.300	0.893	**0.325**	0.907	0.346
	ANECP	**0.894**	0.314	**0.897**	0.304	**0.912**	**0.352**

in most cases. It implies that the high-order structure is very important for NE, and community information cannot be considered more useful than first- and high-order proximity information, although community information is one of the most important characteristics of networks. Fourth, ANECP-AS is better than the ANECP-AF. It proves that the cosine similarity matrix of the feature vector of the original attributes revels more global information of nodes than the original attribute vector itself. In summary, ANECP achieves the best performance by the combination of the micro-structure information with meso-structure information of the network, as well as the cosine similarity of the node attributes.

2) Node Clustering: Node clustering is completed by running K-means++ method on the learned underlying feature representations of nodes. K-means++ clustering algorithm is run 20 times for each dataset, and the average accuracy is reported. The clustering result is presented in Table XII, where bold numbers represent the best results. From Table XII, we can find that:

- Our ANECP and its variants obtain the best clustering performance on 5 of the 6 networks. On average, ANECP improved upon the best baseline method M-NMF by 4.9%. It verifies the effectiveness of our proposed approach.
- In most cases, Structure+Attribute methods (ANECP-TA, DANE and ANRL) obtain superior performance than Structure-only methods (DeepWalk, node2vec and SDNE). Furthermore, the performance of Attribute-only methods (ANECP-AS and ANECP-AF) is second only to Structure+Attribute+Community methods (ANECP-VAE and ANECP). It verifies the importance of attribute information in NE.
- Structure+Community method M-NMF and ANECP-TC achieves superior performance than Structure-only methods (DeepWalk, node2vec and SDNE) and Structure+Attribute method (DANE and ANRL). How-ever, ComVAE is inferior to M-NMF and ANECP-TC. It further demonstrates the significance of the high-order structure for NE.

3) Visualization of latent representation of nodes： To further validate whether the embedding result of ANECP is discriminative, we visualize the latent representation of each node by using t-SNE [36], i.e., each node vector learned by a method is visualized as a point in a two-dimensional space, and different categories are labeled with different colors. The latent representations that can be well separated are more discriminative and easier to classify. Fig. 3 presents the visualization results for the Wisconsin dataset as the rep-

This article has been accepted for publication in a future issue of this conference proceedings, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/DSAA49011.2020.00047, 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)

TABLE VII. ACCURACY OF NODE CLUSTERING

Method	*Cornell*	*Texas*	*Washing*	*Wiscon*	*Hamil*	*Roches*
DeepWalk	0.328	0.380	0.357	0.343	0.265	0.287
Node2Vec	0.343	0.503	0.417	0.355	0.299	0.281
SDNE	0.364	0.578	0.396	0.408	0.274	0.279
DANE	0.393	0.377	0.424	0.398	0.370	0.334
ANRL	0.384	0.488	0.494	0.467	0.319	0.340
ANECP-TA	0.463	0.495	0.514	0.586	0.369	0.331
M-NMF	0.431	**0.631**	0.596	0.456	0.321	0.363
ComVAE	0.380	0.426	0.375	0.364	0.290	0.281
ANECP-TC	0.451	0.444	0.422	0.396	0.323	0.301
ANECP-AF	0.465	0.573	0.316	**0.622**	0.370	0.323
ANECP-AS	0.500	0.530	0.601	0.587	0.319	0.310
ANECP-VAE	**0.545**	0.524	0.628	0.605	0.374	0.360
ANECP	0.527	0.515	**0.667**	0.581	**0.385**	**0.394**

resentative case. The visualization for other datasets is omitted due to the space constraints.

From Fig. 3, we can observe that ANECP can obtain more clear visualization results compared with the other baseline methods, which are manifested as the nodes are closer within cluster and separated among clusters. This can further explain why it has achieved good performance in the aspect of both node classification and clustering tasks.

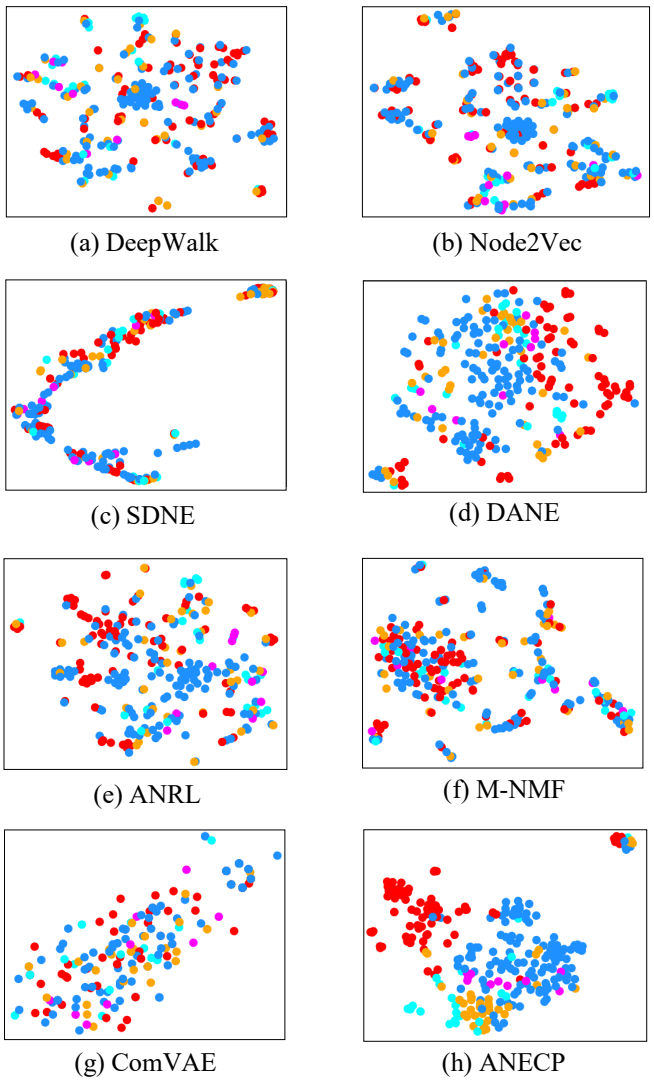

Fig. 3. Visualization of different embedding methods for the Wisconsin dataset

E. *Hyper-Parameter Sensitivity*

In this subsection, we study the hyper-parameter sensitivity of ANECP. We evaluate the effects of different hyper-parameters α, β and γ on the performance of tasks of node classification and node clustering. In the node classification, $\ell 2$-regularized Logistic Regression is utilized as the classifier, and 80% labeled nodes are randomly selected as the training set and the remained nodes as the testing set. Fig. 4 and Fig. 5 present *Micro-F1* of node classification and accuracy of node clustering with respect to different α, β and γ respectively. The trends of *Macro-F1* with respect to different α, β and γ is similar to the one of Micro-F1, so we do not present it.

From Fig. 4 and 5, we can see that values of *Micro-F1* are basically stable under different hyper-parameters expect Cornell and Texas, but for node clustering ANECP is more sensitive, for example, in Fig. 5 (c), the values of AC on Hamilton and Rochester present large fluctuations. Thus, how to set hyper-parameters of ANECP for node clustering is a sensitive issue.

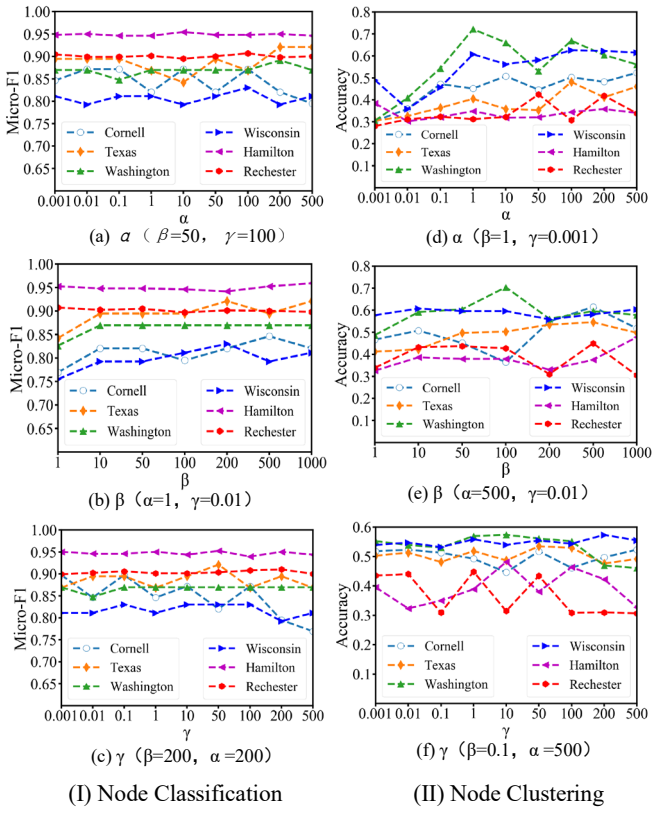

Fig. 4. Sensitivity of ANECP w.r.t. different α, β and γ of node classification and node clustering

V. CONCLUSION

In this study, we develop ANECP framework to investigate the representation learning on social networks. ANECP is the first to incorporate first- and high-order proximity of topological structure, node attributes and community information into the embedding. ANECP not only inherits the ability of deep neural networks for capturing the underlying high non-linearity, but also establishes the consistent and complementary relationship between the node representations and topological structure, node attributes and

the community structure. Consequently, the obtained node representations can preserve proximity and discriminative.

The approach proposed in this study is devoted to homogeneous networks (all nodes have the same type, and all edges have the same type). However, real-world networks are often heterogeneous, i.e., either nodes or edges have multiple types, which contain richer semantic information. So, one of our future work is to extend our ANECP framework to heterogeneous networks to obtain more effective representations.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (61762090, 61966036, and 61662086), The Natural Science Foundation of Yunnan Province (2016FA026), the Project of Innovative Research Team of Yunnan Province (2018HC019), and Program for Innovation Research Team (in Science and Technology) in University of Yunnan Province (IRTSTYN), the Education Department Foundation of Yunnan Province (2019Y0006), the National Social Science Foundation of China (18XZZ005).

REFERENCES

[1] H. Narayanan, M. Belkin, P. Niyogi, "On the relation between low density separation, spectral clustering and graph cuts," in *NeurIPS*, 2007, pp. 1025-1032.

[2] P. Sen, G. Namata, M.Bilgic, L. Getoor, B. Galligher and T. Eliassi-Rad, "Collective classification in network data," *AI MAG*, vol. 29, no. 3, pp. 93-93, 2008.

[3] S. Bhagat, G. Cormode and S. Muthukrishnan, Node classification in social networks, In *Social network data analytics*, Springer, 2011, pp. 115-148.

[4] L. Lü, and T. Zhou, "Link prediction in complex networks: A survey," *PHYSICA A*, vol. 390, no. 6, pp. 1150-1170, 2011.

[5] J. Tang, J. Liu, M. Zhang and Q. Mei,. "Visualizing large-scale and high-dimensional data". In *WWW*, 2016, pp. 287-297.

[6] H. Gao and H. Huang, "Deep Attributed Network Embedding," In *IJCAI*, 2018, pp. 3364-3370.

[7] B. Perozzi, R. Al-Rfou and S. Skiena, "Deepwalk: Online learning of social representations," In *SIGKDD*, 2014, pp. 701-710.

[8] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," In *SIGKDD*, 2016, pp. 855-864.

[9] D. Wang, P. Cui and W. Zhu, "Structural deep network embedding," In *SIGKDD*, 2016, pp. 1225-1234.

[10] P. V. Marsden and N. E. Friedkin. "Network studies of social influence," *Sociological Methods & Research*, vol. 22, no. 1, pp. 127-151, 1993.

[11] M. McPherson, L. Smith-Lovin and J. M. Cook. "Birds of a feather: Homophily in social networks," *Annu Rev Sociol*, vol. 27, no. 1, pp. 415-444, 2001,

[12] H. Wang, E. Chen, Q. Liu, T. Xu, D. Du, W. Su and X. Zhang, "A United Approach to Learning Sparse Attributed Network Embedding," In *ICDM,* 2018, pp. 557-566.

[13] S. Zhou, H. Yang, X. Wang, J. Bu, M. Ester, P.Yu and C. Wang, "Prre: Personalized relation ranking embedding for attributed networks," In *CIKM*, 2018, pp. 823-832.

[14] D. Jin, B. Li, P. Jiao, D. He and W. Zhang, "Network-Specific Variational Auto-Encoder for Embedding in Attribute Networks", In *IJCAI*, 2019, pp. 2663-2669.

[15] M Girvan, M. E. J. Newman, "Community structure in social and biological networks," *PNAS*, vol. 99, no. 12, pp. 7821-7826, 2002.

[16] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu and S. Yang, "Community preserving network embedding," In *AAAI*, 2017, pp. 203--209.

[17] C. Tu, X. Zeng, H. Wang, Z. Zhang, Z. Liu, M. Sun and L. Lin, "A unified framework for community detection and network representation learning," *TKDE*, vol. 31, no. 6, pp. 1051-1065, 2018,.

[18] Y. Li, Y. Wang, T. T. Zhang,J. W. Zhang and Y. Chang, "Learning Network Embedding with Community Structural Information," In *IJCAI*, 2019, pp. 2937-2943.

[19] C. Yang, Z. Liu, D. Zhao, M. Sun and E. Chang, "Network representation learning with rich text information," In *IJCAI*, 2015, pp. 2111-2117.

[20] X. Huang, J Li and X Hu, "Accelerated attributed network embedding," In *SIAM*, 2017, pp. 633-641.

[21] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," In *NIPS*, 2002, pp. 585-591.

[22] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323-2326, 2000.

[23] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan and Q. Mei, "Line: Large-scale information network embedding," In *WWW*, 2015, pp. 1067-1077.

[24] S. Cao, W. Lu and Q. Xu, "Grarep: Learning graph representations with global structural information," In *CIKM*, 2015, pp. 891-900.

[25] H. Wang, E. Chen, Q. Liu, T. Xu, D. Du, W. Su and X. Zhang, "A United Approach to Learning Sparse Attributed Network Embedding," In *ICDM*, 2018, pp. 557-566.

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

[27] Z. Zhang, H. Yang, J. Bu, S. Zhou, P. Yu, J. Zhang and C. Wang, "ANRL: Attributed Network Representation Learning via Deep Neural Networks," In *IJCAI*, 2018, pp. 3155-3161

[28] W. Shi, L. Huang, C. D. Wang, J. H. Li, Y. Tang and C. Fu, "Network embedding via community based variational autoencoder," *IEEE Access*, vol 7, pp. 25323-25333, 2019.

[29] J. Mo, N. Gao, Y. Zhou, Y. Pei and J. Wang, "NANE: Attributed Network Embedding with Local and Global Information," In *WISE*, 2018, pp. 247-261.

[30] M. Rosvall, C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *PNAS*, vol. 105, no. 4, pp. 1118-1123, 2008.

[31] V. D. Blondel, J. L. Guillaume, R. Lambiotte and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech. Theory Exp.*, vol. 2008, no. 10, p. P10008, 2008.

[32] U. N. Raghavan, R. Albert and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E.*, vol. 76, no. 3, pp. 036106, 2007.

[33] D. P. Kingma, M. Welling, "Auto-encoding variational bayes," In *arXiv preprint arXiv:*1312.6114, 2013.

[34] K. Sohn, H. Lee, X. Yan, "Learning structured output representation using deep conditional generative models," In *NIPS*, 2015, pp. 3483-3491.

[35] A. L. Traud, P. J. Mucha, M. A. Porter, "Social structure of facebook networks," *Physica A*, vol. 391, no. 16, pp. 4165-4180, 2012.

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

[37] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang and J. Tang, "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec," In *WSDM*, 2018, pp. 459-467.

[38] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio and Y. Bengio, "Graph attention networks," *CoRR*, abs/1710.10903, 2017.

[39] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, abs/1609.02907, 2016.

[40] D. Jin, H. Wang, J. Dang, D. He and W. Zhang, "Detect overlapping communities via ranking node popularities," In *AAAI*, 2016, pp. 172-178.

[41] Y. Li, K. He, K. Kloster, D. Bindel and J. Hopcroft, "Local spectral clustering for overlapping community detection," *Trans. Knowl. Discov. Data*, vol. 12, no.2, pp. 17:1-17:27.