IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 27, 2019, accepted December 9, 2019, date of publication December 16, 2019,
date of current version December 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2959816

Constrained Consistency Modeling for Attributed Network Embedding

XUAN ZANG[1,2], BO YANG[1,2], SHUANG YANG[1,2], AND HECHANG CHEN[1,3]

[1]Key Laboratory of Symbolic Computation and Knowledge Engineering Attached to the Ministry of Education, Jilin University, Changchun 130012, China
[2]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[3]School of Artificial Intelligence, Jilin University, Changchun 130012, China

Corresponding authors: Bo Yang (ybo@jlu.edu.cn) and Hechang Chen (chenhc@jlu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572226, Grant 61876069, and Grant 61902145, in part by the Jilin Province Key Scientific and Technological Research and Development Project under Grant 20180201067GX and Grant 20180201044GX, and in part by the University Science and Technology Research Plan Project of Jilin Province under Grant JJKH20190156KJ.

ABSTRACT Network embedding has emerged as a fundamental approach to network analysis tasks. Its main purpose is to learn a suitable mapping function to convert nodes in networks into a low-dimensional representations. The majority of existing studies concentrate solely on network topology structure. However, nodes are commonly associated with sufficient attribute information in real-world networks. Therefore, network embedding combining network topology structure and attribute information could be promisingly beneficial. Given this, we propose a novel attributed network embedding method called Consistency Constrained Attributed Network Embedding (CCANE), which preserves more complete information for nodes when learning the embedding representations. On the basis of the consistency of topology structure and node attributes, the CCANE is capable of learning the structure embeddings and attribute embeddings of nodes simultaneously, and then concatenate them to obtain the integrated vector representations. Moreover, the CCANE is scalable of dealing with large-scale of networks by decomposing the complicated optimization process into multiple sub-tasks in parallel. Experimental results testify the feasibility and superiority of the CCANE compared to the state-of-the-arts.

INDEX TERMS Attributed network, network embedding, representation learning.

I. INTRODUCTION

Mining valuable information from networks is of considerable significance as networks are widely existed in the real world [1], [2]. Studies on network analysis have attracted increasing attention, and various tasks have been raised and regarded as open problems, such as node classification [3], visualization [4], and community detection [5]. However, sparsity is a persistent problem when solving the tasks of network analysis using conventional methods of network representation [6]. Therefore, an innovative idea is desirable to learn appropriate representations for nodes in networks. Network embedding [7], a novel and effective network representation learning method, learns low-dimensional and dense node representations which can be directly utilized as the input of network analysis tasks, so as to avoid the problem of sparsity.

The associate editor coordinating the review of this manuscript and approving it for publication was Pasquale De Meo.

Diverse network embedding algorithms [12], [15], [16] have been proposed recently. They generally focus only on the linkage information between nodes. However, real-world networks are usually attributed networks [25] where nodes are usually associated with plentiful attribute information, e.g., profiles of users in social networks and topics of papers in citation networks. Properties of networks can hardly be described thoroughly only by structural information since the linkage information between nodes is limited in a portion of networks. In this case, it is necessary to mine attribute information of nodes in networks. Although topology structure and node attributes reveal the properties of the network from different perspectives, homophily implies that both of them are correlative with each other [8]. Moreover, attributes of nodes have a considerable impact on the formation of the network structure. For instance, people with similar hobbies tend to be friends in social networks; papers are more likely to be cited by papers with the same keywords. Attributes of nodes reflect the internal characteristics of networks to

some extent and benefit network embedding as supplementary information to network structure. Incorporating attribute information in network representation learning is advantageous to solve the problem of information loss caused by sparsity or noise [20].

Although plentiful attributed network embedding methods [17], [18], [23], [24], [27] have emerged, they all have more or less limitations. For example, attribute information is presented in various forms, and existing methods are not universal to handle the information beyond text features. In addition, many existing methods capture the proximity of linked nodes while ignoring nodes sharing common neighbors, making them challenging to describe network characteristics adequately.

To tackle the aforementioned problems, we propose an effective attributed network embedding method named CCANE. Specifically, the CCANE constructs similarity models in the network structure space and attribute space, respectively. In order to make up for the scarcity of linkage information, we consider the similarity of the neighbor structure of nodes by introducing the second-order proximity when building the structure proximity matrix. In addition, we utilize the features to obtain symmetric attribute proximity matrix, which can be applied to matrix decomposition, so that the CCANE can process multiple types of attribute information, not just text features. Then, we can obtain two embeddings of each node, i.e., structure embedding and attribute embedding, by matrix decomposition of both proximity matrices. The former preserves structure similarity between each pair of nodes, and the latter holds attribute similarity between nodes. Moreover, there is a certain degree of consistency between them in line with homophily. The final embedding for each node is to concatenate attribute embedding and structure embedding together. Our contributions are summarized as follows:

- A novel attributed network embedding method named CCANE is proposed, in which the structure proximity and attribute proximity between each pair of nodes are jointly considered when learning network representation, and thus, the consistency and complementarity relations between them are built.
- The model is accelerated by decomposing complex model optimization into a range of sub-problems in parallel, which is readily applied to distributed computing for large-scale datasets.
- Superiority of our proposed CCANE is validated comparing to state-of-the-arts on attributed datasets. Extensive experimental results demonstrate the feasibility and efficiency of the CCANE.

Organization: In Section II, the model CCANE is proposed, and the model optimization process is accelerated. As for Section III, we corroborate the superiority and effectiveness of the CCANE through experimental results. Section IV introduces the related work of this paper. The conclusion of our work and the future work are generalized in Section V.

II. CONSISTENCY CONSTRAINED ATTRIBUTED NETWORK EMBEDDING

In this section, we first demonstrate the problem definition. Then, the framework and modeling process of the CCANE is introduced. Finally, the parallel optimization method for the model is detailed.

A. PROBLEM DEFINITION

An attributed network can be represneted by an undirected graph $G = (V, E, W, Z)$, in which V and E represent node set and edge set. $\mathbf{W} \in \mathbb{R}^{n\times n}(n = |V|)$ is a weight matrix, and $\mathbf{W}_{i,j} \in \mathbf{W}$ means the weight of edge $e_{ij} \in E$. A larger value of $\mathbf{W}_{i,j}$ indicates a stronger topological connection between nodes i and j, in particular, $\mathbf{W}_{i,j} = 0$ indicates there is no edge between them. $\mathbf{Z} \in \mathbb{R}^{n\times m}$ is a attribute matrix, $\mathbf{Z}(i, :)$ represents a feature vector of node i with m attributes.

Definition 1 (Structure Proximity): The structure proximity matrix $\mathbf{X}$ integrates the first-order proximity $\mathbf{X^{(1)}}$ and the second-order proximity $\mathbf{X^{(2)}}$ through $\mathbf{X} = \mathbf{X^{(1)}} + \mathbf{X^{(2)}}$.

The first-order proximity describes the topological connection between nodes, i.e., $\mathbf{X^{(1)}}_{i,j} = \mathbf{W}_{i,j}$. The second-order proximity measures the similarity of neighbor structures between nodes. Since cosine similarity is a classical and widely used method and more conducive to the processing of sparse matrices, we define the second-order proximity $\mathbf{X^{(2)}}_{i,j}$ as the cosine similarity between $\mathbf{X^{(1)}}(i, :)$ and $\mathbf{X^{(1)}}(j, :)$.

Definition 2 (Attribute Proximity): The attribute proximity $\mathbf{P}_{i,j}$ between nodes is defined as the cosine similarity of their attribute vectors, i.e., $\mathbf{P}_{i,j} = \frac{\sum_{k=1}^{m} \mathbf{Z}(i,k)\times\mathbf{Z}(j,k)}{\sqrt{\sum_{k=1}^{m} \mathbf{Z}(i,k)^2}\times\sqrt{\sum_{k=1}^{m} \mathbf{Z}(j,k)^2}}$.

Problem Definition: Given an attributed network $G = (V, E, W, Z)$, the CCANE is designed for learning a low-dimensional representation $\mathbf{Y}(i, :) \in \mathbb{R}^d$ for each node i, where $d \ll |V|$. The CCANE is expected to retain proximity and ensure the consistency constraint in topology structure space and attribute space.

B. MODEL DESCRIPTION

The Fig. 1 illustrates the overall framework of our proposed CCANE. First, we build the structure proximity matrix $\mathbf{X}$ and the attribute proximity matrix $\mathbf{P}$ according to the weight matrix $\mathbf{W}$ and the attribute matrix $\mathbf{Z}$ respectively. Then the symmetric matrix factorization is applied to both proximity matrices so that we obtain the structure embedding $\mathbf{Y^W}$ and the attribute embedding $\mathbf{Y^Z}$, which are in line with the consistency constrains. Finally, we concatenate them to the final embedding $\mathbf{Y}$. We demonstrate the details now.

1) NETWORK STRUCTURE MODELING

The representation of network structure embedding denoted by $\mathbf{Y^W} \in \mathbb{R}^{n\times d}$ is supposed to preserve proximity in topology structure space. The structure proximity matrix $\mathbf{X} \in \mathbb{R}^{n\times n}$ can be calculated after the first- and second-order proximity are obtained. Specifically, inspired by the symmetric matrix

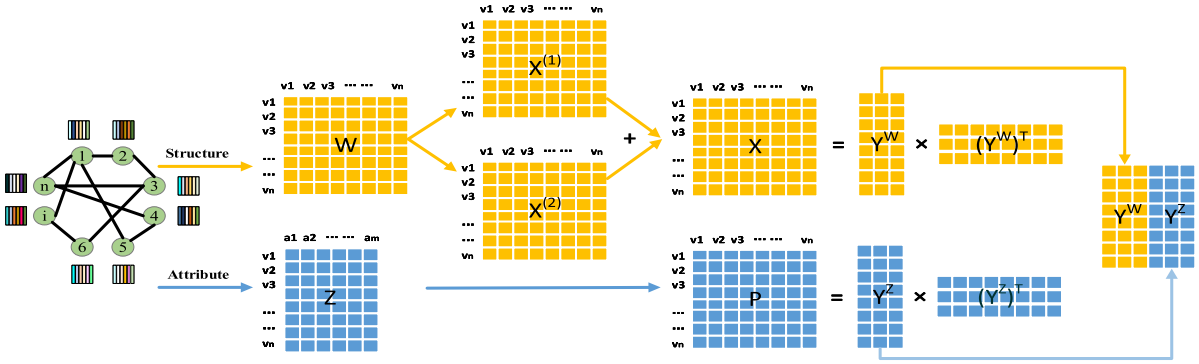


FIGURE 1. A framework of CCANE. The weight matrix W and the attribute matrix Z are utilized to build proximity matrix. Matrix factorization are performed to obtain the final embedding Y.

factorization [9], $\mathbf{X}$ can be approached by the product of $\mathbf{Y^W}$ and the transpose of $\mathbf{Y^W}$ (i.e., $(\mathbf{Y^W})^\top$) as it is a symmetric matrix in undirected networks. Then, on the basis of structure proximity, we define the loss function in the structure space as:

$$L_W = \left\| \mathbf{X} - \mathbf{Y^W}(\mathbf{Y^W})^\top \right\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\mathbf{X}_{i,j} - \mathbf{Y^W}(i,:)\mathbf{Y^W}(j,:)^\top\right)^2 \tag{1}$$

2) ATTRIBUTE PROXIMITY MODELING

The representation of attribute embedding denoted by $\mathbf{Y^Z} \in \mathbb{R}^{n\times d}$ is supposed to retain the attribute proximity. And the symmetric attribute proximity matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$ can be obtained by calculating the attribute proximity between each pair of nodes. Then, $\mathbf{P}$ can be approached by the product of $\mathbf{Y^Z}$ and $(\mathbf{Y^Z})^\top$ under the fundamental theory of symmetric matrix factorization. The loss function in the attribute space is defined as:

$$L_Z = \left\| \mathbf{P} - \mathbf{Y^Z}(\mathbf{Y^Z})^\top \right\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\mathbf{P}_{i,j} - \mathbf{Y^Z}(i,:)\mathbf{Y^Z}(j,:)^\top\right)^2 \tag{2}$$

3) CONSISTENCY CONSTRAINED REPRESENTATION

The network homophily expounds that things with the same characteristics are more likely to come together [28]. On the one hand, both topology structure and node attributes embody the nature of the identical network, and they should be consistent to a certain extent [10]. On the other hand, they provide complementary information about the same network. Therefore, the consistency and complementarity of structure embedding and attribute embedding should be insured accordingly.

By modeling structure information and attribute information, the structure embedding representation $\mathbf{Y^W}$ (reserving proximity in network structure space) and attribute embedding representation $\mathbf{Y^Z}$ (reserving proximity in attribute space) can be acquired, respectively. Then, a simple method is to concatenate $\mathbf{Y^W}$ and $\mathbf{Y^Z}$ directly as the final embedding since attributed network representation should reckon on both network topology structure and attribute information. For example, the final representation of node i can be expressed as $\mathbf{Y}(i,:) = (\mathbf{Y^W}_{i,1}, \ldots, \mathbf{Y^W}_{i,d}, \mathbf{Y^Z}_{i,1}, \ldots, \mathbf{Y^Z}_{i,d})$. In this way, one can characterize the network information based on two aspects of similarity and guarantee their complementarity, yet it is powerless to guarantee consistency. Therefore, considering the consistency of structure embedding and attribute embedding, we take the Euclidean distance of $\mathbf{Y^W}(i,:)$ and $\mathbf{Y^Z}(i,:)$ as metric to measure differences between these two representation vectors and define the loss function for consistency constraints as:

$$L_C = \sum_{i=1}^{n}\left\| \mathbf{Y^W}(i,:) - \mathbf{Y^Z}(i,:) \right\|_2^2 \tag{3}$$

4) JOINT EMBEDDING REPRESENTATION LEARNING

With the purpose to preserve the similarity in both structure space and attribute space, and guaranteeing the consistency and complementarity of learned representations, we propose an integrated representation learning framework, for which the objective function is formulated as follows:

$$\min_{\mathbf{Y^W},\mathbf{Y^Z}} L = \left\| \mathbf{X} - \mathbf{Y^W}(\mathbf{Y^W})^\top \right\|_F^2 + \left\| \mathbf{P} - \mathbf{Y^Z}(\mathbf{Y^Z})^\top \right\|_F^2 + \lambda \sum_{i=1}^{n}\left\| \mathbf{Y^W}(i,:) - \mathbf{Y^Z}(i,:) \right\|_2^2 \tag{4}$$

where λ is a constraint parameter for consistency between structure embedding and attribute embedding. $\lambda = 0$ means that structure embeddings and attribute embeddings are learned independently and irrelevantly. When an optimal $\lambda > 0$, the learned embeddings for nodes satisfy the requirements of consistency and complementarity.

C. MODEL OPTIMIZATION SCHEME

Stochastic gradient descent is a commonly used optimization algorithm. However, on account of the limitation of memory space and running time, stochastic gradient descent is infeasible for large-scale datasets. Given this, we propose to decompose the complex model optimization problem into multiple

sub-tasks. In this case, the parameter optimization procedure can be executed in parallel, and ultimately we can get the optimal solution of primal function faster. Specifically, we use Alternating Direction Method of Multipliers(ADMM) [11] to optimize the objective function in attributed network representation learning.

In line with consistency and complementarity, the objective function can be optimized in parallel. The embedding matrix $\mathbf{Y^W}$ and $\mathbf{Y^Z}$ can be updated simultaneously, and vectors in all rows of $\mathbf{Y^W}$ or $\mathbf{Y^Z}$ can be updated independently during the process of iteration. We accelerate the model optimization by ADMM as follows.

First, we set copies $\mathbf{B^W} = \mathbf{Y^W}$ and $\mathbf{B^Z} = \mathbf{Y^Z}$, then, Eq. (1) and Eq. (2) can be rewritten as:

$$\begin{aligned} L_W &= \sum_{i=1}^{n} \left\| \mathbf{X}(i,:) - \mathbf{Y^W}(i,:)(\mathbf{B^W})^\top \right\|_2^2 \\ &= \sum_{i=1}^{n} \left\| (\mathbf{X}(i,:))^\top - \mathbf{Y^W}(\mathbf{B^W}(i,:))^\top \right\|_2^2 \qquad (5) \\ L_Z &= \sum_{i=1}^{n} \left\| \mathbf{P}(i,:) - \mathbf{Y^Z}(i,:)(\mathbf{B^Z})^\top \right\|_2^2 \\ &= \sum_{i=1}^{n} \left\| (\mathbf{P}(i,:))^\top - \mathbf{Y^Z}(\mathbf{B^Z}(i,:))^\top \right\|_2^2 \qquad (6) \end{aligned}$$

Further, the objective function Eq. (4) can be rewritten in a linearly constrained form:

$$\begin{aligned} \min_{\mathbf{Y^W},\mathbf{Y^Z}} L = & \sum_{i=1}^{n} \left\| \mathbf{X}(i,:) - \mathbf{Y^W}(i,:)(\mathbf{B^W})^\top \right\|_2^2 + \sum_{i=1}^{n} \|\mathbf{P}(i,:) \\ & - \mathbf{Y^Z}(i,:)(\mathbf{B^Z})^\top \|_2^2 + \lambda \sum_{i=1}^{n} \left\| \mathbf{Y^W}(i,:) - \mathbf{B^Z}(i,:) \right\|_2^2 \\ s.t. \quad & \mathbf{Y^W}(i,:) = \mathbf{B^W}(i,:), \quad \mathbf{Y^Z}(i,:) = \mathbf{B^Z}(i,:), \\ & for \quad i = 1, \ldots, n \qquad (7) \end{aligned}$$

Solving the objective in Eq. (7) is a bi-convex optimization problem, where $2n$ vectors (n rows $\mathbf{Y^W}(i,:)$ and n rows $\mathbf{Y^Z}(i,:)$) can be updated independently by ADMM, and the augmented Lagrangian corresponding to Eq. (7) can be formulated as:

$$\begin{aligned} L' = & \sum_{i=1}^{n} \left\| \mathbf{X}(i,:) - \mathbf{Y^W}(i,:)(\mathbf{B^W})^\top \right\|_2^2 \\ & + \frac{\rho^W}{2} \sum_{i=1}^{n} \left(\left\| \mathbf{Y^W}(i,:) - \mathbf{B^W}(i,:) + \mathbf{U^W}(i,:) \right\|_2^2 \right) \\ & + \sum_{i=1}^{n} \left\| \mathbf{P}(i,:) - \mathbf{Y^Z}(i,:)(\mathbf{B^Z})^\top \right\|_2^2 \\ & + \frac{\rho^Z}{2} \sum_{i=1}^{n} \left(\left\| \mathbf{Y^Z}(i,:) - \mathbf{B^Z}(i,:) + \mathbf{U^Z}(i,:) \right\|_2^2 \right) \\ & + \lambda \sum_{i=1}^{n} \left\| \mathbf{Y^W}(i,:) - \mathbf{B^Z}(i,:) \right\|_2^2 \qquad (8) \end{aligned}$$

where ρ^W and $\rho^Z > 0$ are two penalty factors of network structure information and attribute information, respectively. $\mathbf{U^W}(i,:)$ and $\mathbf{U^Z}(i,:)$ are dual variables. Eq. (8) is not convex, yet it is bi-convex. Eq. (8) is separable for both $\mathbf{Y}(\mathbf{Y^W}, \mathbf{Y^Z})$ and $\mathbf{B}(\mathbf{B^W}, \mathbf{B^Z})$. Because l2-norm is convex, it is well-reasoned to prove that the convex optimization problem corresponding to $\mathbf{Y}(\mathbf{Y^W}, \mathbf{Y^Z})$ is convex when $\mathbf{B}(\mathbf{B^W}, \mathbf{B^Z})$ is invariable and vice versa. On the basis of ADMM, we keep $\mathbf{Y}$ fixed, then Eq. (8) is convex involving $\mathbf{B}$ and easy to be solved. Similarly, we keep $\mathbf{B}$ fixed when updating $\mathbf{Y}$. The problem of minimizing Eq. (8) is transformed into updating $\mathbf{Y}$ and $\mathbf{B}$ iteratively, and in the process of $(r+1)^{th}$ iteration, $\mathbf{Y^W}(i,:)$ is updated as:

$$\begin{aligned} & (\mathbf{Y^W}(i,:))^{r+1} \\ & = \frac{2\mathbf{X}(i,:)(\mathbf{B^W})^r + \rho^W \left((\mathbf{B^W}(i,:))^r - (\mathbf{U^W}(i,:))^r \right) + 2\lambda (\mathbf{B^Z}(i,:))^r}{2\left((\mathbf{B^W})^r\right)^\top (\mathbf{B^W})^r + (\rho^W + 2\lambda)\mathbf{I}} \end{aligned} \qquad (9)$$

Similarly, we get the update rule of $\mathbf{Y^Z}(i,:)$ as:

$$\begin{aligned} & (\mathbf{Y^Z}(i,:))^{r+1} \\ & = \underset{\mathbf{Y^Z}(i,:)}{argmin} \left\| \mathbf{P}(i,:) - \mathbf{Y^Z}(i,:)((\mathbf{B^Z})^r)^\top \right\|_2^2 \\ & \quad + \frac{\rho^Z}{2} \left\| \mathbf{Y^Z}(i,:) - (\mathbf{B^Z}(i,:))^r + (\mathbf{U^Z}(i,:))^r \right\|_2^2 \\ & = \frac{2\mathbf{P}(i,:)(\mathbf{B^Z})^r + \rho^Z((\mathbf{B^Z}(i,:))^r - (\mathbf{U^Z}(i,:))^r)}{2((\mathbf{B^Z})^r)^\top (\mathbf{B^Z})^r + \rho^Z \mathbf{I}} \end{aligned} \qquad (10)$$

In the $(r+1)^{th}$ iteration, the convex optimization process and update rules corresponding to variables $\mathbf{B^W}(i,:)$ and $\mathbf{B^Z}(i,:)$ are shown in (11) and (12), as shown at the bottom of the next page.

Then, the dual variables $\mathbf{U^W}$ and $\mathbf{U^Z}$ are updated as follows:

$$(\mathbf{U^W})^{r+1} = (\mathbf{U^W})^r + ((\mathbf{Y^W})^{r+1} - (\mathbf{B^W})^{r+1}) \qquad (13)$$

$$(\mathbf{U^Z})^{r+1} = (\mathbf{U^Z})^r + ((\mathbf{Y^Z})^{r+1} - (\mathbf{B^Z})^{r+1}) \qquad (14)$$

Note that $(\mathbf{B^W})^{r+1}$ is calculable only when $(\mathbf{Y^W})^{r+1}$ is known at each iteration. To update $\mathbf{Y^W}$, one needs to update all rows of $\mathbf{Y^W}$. We can decompose them to n updating steps in parallel since they are independent of each other. When we update $\mathbf{Y^W}$, the previous iteration of $\mathbf{B^W}$ and corresponding $\mathbf{B^Z}$ should be given. Similarly, the process of updating $\mathbf{B^W}$ still can be converted into n sub-problems in parallel. In this paper, the original optimization is decomposed into $4n$ sub-problems (i.e., n rows $\mathbf{Y^W}(i,:)$, n rows $\mathbf{Y^Z}(i,:)$, n rows $\mathbf{B^W}(i,:)$, and n rows $\mathbf{B^Z}(i,:)$), and n nodes update in parallel. Please refer to Algorithm 1 for more details about the optimization procedure.

D. COMPLEXITY ANALYSIS

According to the characteristics of ADMM, a few iterations can achieve a moderate precision. In each iteration, according to the update rule of $\mathbf{Y^W}(i,:)$ in Eq. (9), it can be known that

Algorithm 1 CCANE

Input: $\mathbf{W}$, $\mathbf{Z}$.
Output: $\mathbf{Y} = [\mathbf{Y^W Y^Z}]$.

Initialize $(\mathbf{Y^W})^0$, $(\mathbf{Y^Z})^0$;
Set $(\mathbf{B^W})^0 = (\mathbf{Y^W})^0$, $(\mathbf{B^Z})^0 = (\mathbf{Y^Z})^0$, $(\mathbf{U^W})^0 = (\mathbf{U^Z})^0 = 0$, $r = 0$;
repeat
  Calculate $((\mathbf{B^W})^r)^\top(\mathbf{B^W})^r$, $((\mathbf{B^Z})^r)^\top(\mathbf{B^Z})^r$;
  for $i = 1$ to n in parallel **do**
    Compute proximity vectors;
    Update $(\mathbf{Y^W}(i,:))^{r+1}$, $(\mathbf{Y^Z}(i,:))^{r+1}$ by Eq. (9), (10);
  end for
  Calculate $((\mathbf{Y^W})^{r+1})^\top(\mathbf{Y^W})^{r+1}$, $((\mathbf{Y^Z})^{r+1})^\top(\mathbf{Y^Z})^{r+1}$;
  for $i = 1$ to n in parallel **do**
    Compute proximity vectors;
    Update $(\mathbf{B^W}(i,:))^{r+1}$, $(\mathbf{B^Z}(i,:))^{r+1}$ by Eq. (11), (12);
  end for
  Update $(\mathbf{U^W})^{r+1}$, $(\mathbf{U^Z})^{r+1}$ by Eq. (13), (14);
  $r \leftarrow r + 1$;
until Convergence;
return $\mathbf{Y} = [\mathbf{Y^W Y^Z}]$.

the update time of d-dimensional vector $\mathbf{Y^W}(i,:)$ is $O(dn)$, since only one calculation $((\mathbf{B^W})^r)^\top(\mathbf{B^W})^r$ is needed. Due to $d \ll n$, the time complexity of each sub-problem can be summarized as $O(n)$. T_X represents the time required to calculate the proximity between nodes, w represents the number of processors, then the time complexity of CCANE is $O(T_X + \frac{n^2}{w})$, which is equivalent to $O(nN_W + nN_Z + \frac{n^2}{w})$, where N_W and N_Z represent the number of non-zero elements in the weight matrix $\mathbf{W}$ and the attribute matrix $\mathbf{Z}$ respectively.

III. EXPERIMENTS

We conduct experiments to assess the contributions of the CCANE, including node classification and visualization. In the experiments, the processor is Intel(R) Core(TM) i5-8400, the main frequency of CPU is 2.80ghz, the memory is 8G, the running system is WIN10, and the programming software is Matlab + Python.

A. DATASETS

Before performing the specific experiments, we first introduce three attributed datasets. Please refer to Table 1 for more details.

TABLE 1. Statistics of datasets.

Dataset	BlogCatalog	Flickr	Wiki
Nodes($\|V\|$)	5,196	7,564	2405
Edges($\|E\|$)	171,743	239,365	17981
Attribute(m)	8,189	12,047	4973
Label(l)	6	9	17

- **BlogCatalog** is a blogger website in which each node represents a user, the following relationship constitutes edges and the keywords of blogs posted by users are served as node attributes.
- **Flickr** is an online platform for users to share photos and provides online community services. A social network can be formed when users follow each other. We take the information users annotated when they upload photos as attribute information.
- **Wiki** is a document network containing documents collected from Wikipedia, in which a node represents a document, the edges represent relationships between documents, and the TF-IDFs of words are used as the attributes of documents.

$$
\begin{aligned}
(\mathbf{B^W}(i,:))^{r+1} &= \underset{\mathbf{B^W}(i,:)}{argmin} \left\| (\mathbf{X}(i,:))^\top - (\mathbf{Y^W})^{r+1}(\mathbf{B^W}(i,:))^\top \right\|_2^2 \\
&\quad + \frac{\rho^W}{2} \left\| \mathbf{B^W}(i,:) - (\mathbf{Y^W}(i,:))^{r+1} + (\mathbf{U^W}(i,:))^r \right\|_2^2 \\
&= \frac{2\mathbf{X}(i,:)(\mathbf{Y^W})^{r+1} + \rho^W((\mathbf{Y^W}(i,:))^{r+1} + (\mathbf{U^W}(i,:))^r)}{2((\mathbf{Y^W})^{r+1})^\top(\mathbf{Y^W})^{r+1} + \rho^W\mathbf{I}} \qquad (11) \\
(\mathbf{B^Z}(i,:))^{r+1} &= \underset{\mathbf{B^Z}(i,:)}{argmin} \left\| (\mathbf{P}(i,:))^\top - (\mathbf{Y^Z})^{r+1}(\mathbf{B^Z}(i,:))^\top \right\|_2^2 \\
&\quad + \frac{\rho^Z}{2} \left\| \mathbf{B^Z}(i,:) - (\mathbf{Y^Z}(i,:))^{r+1} + (\mathbf{U^Z}(i,:))^r \right\|_2^2 \\
&\quad + \lambda \left\| \mathbf{B^Z}(i,:) - (\mathbf{Y^W}(i,:))^{r+1} \right\|_2^2 \\
&= \frac{2\mathbf{P}(i,:)(\mathbf{Y^Z})^{r+1} + \rho^Z((\mathbf{Y^Z}(i,:))^{r+1} + (\mathbf{U^Z}(i,:))^r) + 2\lambda(\mathbf{Y^W}(i,:))^{r+1}}{2((\mathbf{Y^Z})^{r+1})^\top(\mathbf{Y^Z})^{r+1} + (\rho^Z + 2\lambda)\mathbf{I}} \qquad (12)
\end{aligned}
$$

TABLE 2. Node classification results on BlogCatalog.

Metrics	Micro-F1						Macro-F1					
Training Percentage	10%	20%	30%	40%	50%	60%	10%	20%	30%	40%	50%	60%
DeepWalk	0.5849	0.6247	0.6305	0.6527	0.6605	0.6609	0.5788	0.6189	0.6242	0.6474	0.6459	0.6559
Node2vec	0.5771	0.5973	0.6272	0.6302	0.6432	0.6522	0.5701	0.5911	0.6211	0.6243	0.6373	0.6483
LINE	0.6675	0.6943	0.6987	0.7142	0.7039	0.7042	0.6614	0.6878	0.6959	0.7085	0.6982	0.6996
TADW	0.8265	0.8446	0.8478	0.8537	0.8545	0.8533	0.8239	0.8422	0.8454	0.8515	0.8528	0.8520
AANE	0.5676	0.6870	0.7292	0.7354	0.7378	0.7407	0.5164	0.6699	0.7220	0.7213	0.7232	0.7257
CCANE*	**0.8756**	**0.9197**	**0.9349**	**0.9397**	**0.9430**	**0.9399**	**0.8739**	**0.9184**	**0.9335**	**0.9386**	**0.9420**	**0.9388**

TABLE 3. Node classification results on Flickr.

Metrics	Micro-F1						Macro-F1					
Training Percentage	10%	20%	30%	40%	50%	60%	10%	20%	30%	40%	50%	60%
DeepWalk	0.4147	0.4698	0.4818	0.5023	0.5018	0.5346	0.4127	0.4632	0.4755	0.4951	0.4928	0.5275
Node2vec	0.4391	0.4627	0.4691	0.5061	0.5111	0.5353	0.4360	0.4566	0.4625	0.4997	0.5036	0.5277
LINE	0.5105	0.5302	0.5338	0.5430	0.5448	0.5551	0.5021	0.5225	0.5260	0.5353	0.5391	0.5483
TADW	0.5601	0.5988	0.6179	0.6376	0.6441	0.6545	0.5417	0.5871	0.6072	0.6285	0.6371	0.6469
AANE	0.4467	0.5990	0.7959	0.8099	0.8313	0.8471	0.4718	0.6276	0.7937	0.8106	0.8276	0.8472
CCANE*	**0.7559**	**0.8068**	**0.8285**	**0.8370**	**0.8489**	**0.8620**	**0.7292**	**0.8041**	**0.8259**	**0.8354**	**0.8506**	**0.8603**

TABLE 4. Node classification results on Wiki.

Metrics	Micro-F1						Macro-F1					
Training Percentage	10%	20%	30%	40%	50%	60%	10%	20%	30%	40%	50%	60%
DeepWalk	0.5870	0.6522	0.6609	0.6749	0.6816	0.6985	0.4531	**0.5454**	0.5659	0.5787	0.5855	0.5979
Node2vec	0.5834	0.6148	0.6247	0.6383	0.6384	0.6642	0.4081	0.4935	0.5240	0.5472	0.5331	0.5432
LINE	0.4527	0.5093	0.5255	0.5502	0.5660	0.5748	0.3055	0.3520	0.3665	0.3856	0.3949	0.4049
TADW	0.6286	0.6684	0.7013	0.7228	0.7315	0.7307	0.4315	0.4742	0.5394	0.5700	0.5922	0.5997
AANE	0.6028	0.6709	0.7090	0.7277	0.7390	0.7339	0.3703	0.4512	0.4992	0.5381	0.5466	0.5394
CCANE*	**0.6711**	**0.7099**	**0.7548**	**0.7757**	**0.7855**	**0.7890**	**0.4578**	0.5177	**0.5887**	**0.6085**	**0.6297**	**0.6306**

B. BASELINES

To demonstrate the effectiveness of the CCANE, five representative network representation learning methods are compared and analyzed.

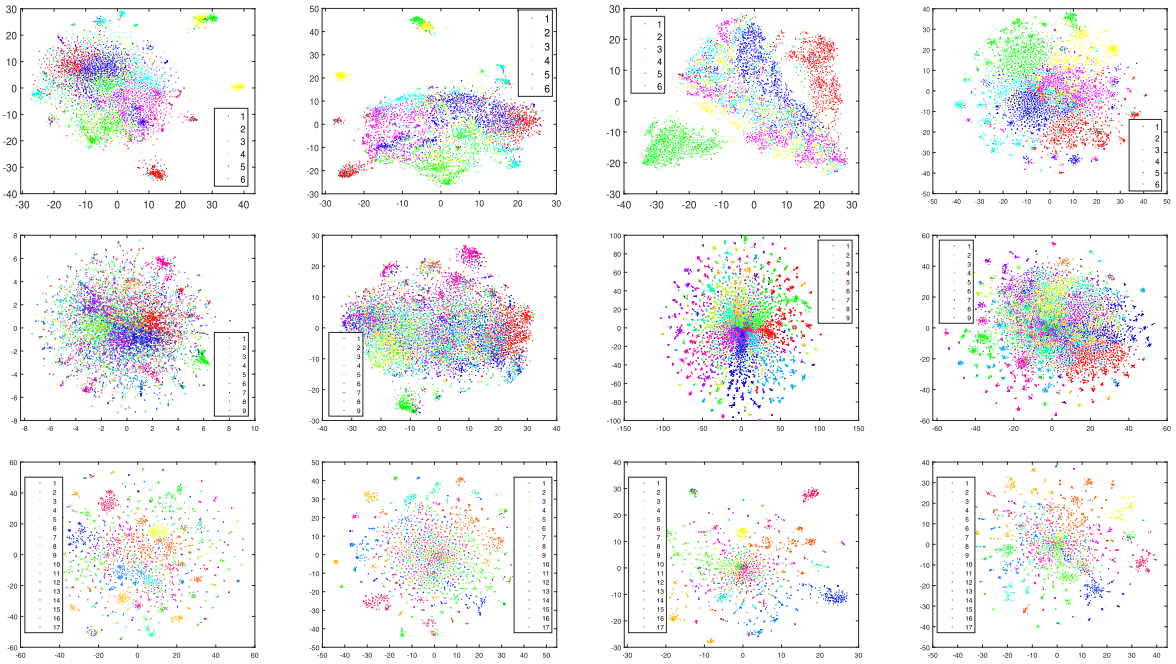
- **DeepWalk** [12] learns vector representations of nodes using random walk path to simulate nature language as the input of the skip-gram model.
- **LINE** [15] (Large-scale Information Network Embedding) considers both local and global topology structures when learning network embedding.
- **Node2vec** [16] preserves the information of neighbor structure by presenting a biased random walk strategy based on DeepWalk.
- **TADW** [17] (Text-Associated DeepWalk) absorbs text attributes of nodes when learning network representation by matrix factorization.
- **AANE** [18] (Accelerated Attributed Network Embedding) aims at dealing with large-scale attribute networks by incorporating lasso penalty into an attribute network representation framework.

To ensure the fairness, we set the dimensionality d of node vectors to 200 for all methods. Particularly, for DeepWalk and Node2vec based on random walk, each node walks 10 times with length 80, and the context window is set to 10. For Node2vec, the optimal parameters are determined by two parameters p and q through grid search($p = 4$, $q = 0.25$ on the BlogCatalog, $p = 0.25$, $q = 4$ on the Flickr and $p = 0.5$, $q = 2$ on the Wiki).

C. NODE CLASSIFICATION

To verify the effectiveness of the CCANE in terms of node classification, training dataset is randomly extracted under different percentage of labeled nodes from the whole dataset, and then, the remainding parts of the dataset are used for testing. For each dataset, all comparisons and our proposed method run 10 times, and Micro-F1 and Macro-F1 are employed as metrics to quantitatively estimate the performance.

Experimental results of node classification on three datasets are shown in Table 2, 3 and 4, respectively. Overall, the CCANE outperforms all these comparisons in different settings, followed by attributed network embedding methods of TADW and AANE. Specifically, the performance of DeepWalk, LINE, and Node2vec, which focus only on network structure, is significantly worse than that of the CCANE, implying the necessity of considering the attribute information. The TADW is slightly worse than the CCANE on BlogCatalog and Wiki, and declines fast in terms of accuracy on Flickr in any cases. That's because it uses Singular Value Decomposition(SVD) to get attribute matrices and integrate attribute features into the final embeddings, which cannot extract features accurately. The performance of AANE is barely satisfactory when the proportion of the training dataset is no bigger than 30%, since AANE only considers the proximity of connecting nodes, while ignoring high-order proximity. By contrast, the CCANE has the ability to assign labels to unlabelled nodes, and thus, it only needs



(a) DeepWalk (b) LINE (c) AANE (d) CCANE

FIGURE 2. Visualization of node embeddings learned by different methods in order of BlogCatalog, Flickr and Wiki.

a small portion of labels to achieve a better performance. LINE outperforms DeepWalk and Node2vec, indicating the effectiveness of uniting the first- and second-order proximity explicitly.

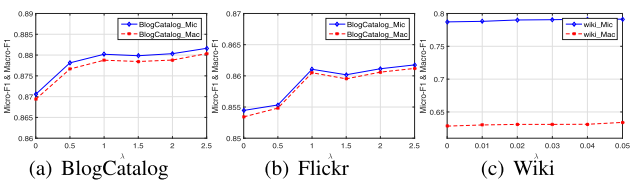
D. VISUALIZATION

With the purpose of exhibiting the experimental results more intuitively, we visualize the results of node embeddings learned by the CCANE and other comparisons. As shown in Fig. 2, node embeddings are colored according to their labels using t-SNE [19]. We can see that the distribution of node embeddings learned by the CCANE is relatively uniform, and the nodes of the same color are mainly concentrated in the same area. It indicates that the CCANE is more capable of providing high-quality representations of nodes (i.e., embeddings), which enables the CCANE to cluster and classify the nodes with different features more accurately. The AANE is slightly worse than the CCANE. On the BlogCatalog, it has the ability to separate the red and the green labels accurately while other labels are not well grouped. Besides, on the Flickr and Wiki, the nodes of the same label are grouped through AANE. However, the learned embeddings does not make good use of solution space, and the nodes of different labels in the center are mixed up. The visualization results of DeepWalk and LINE are not satisfactory in all three datasets. In summary, methods incorporating attribute information can enhance the quality of network embedding.

E. PARAMETER TUNING

1) THE PARAMETER SENSITIVITY OF λ

The parameter λ influences on the consistency of structure embedding and attribute embedding. To show the impact of λ intuitively, curves of Micro-F1 and Macro-F1 on node classification are plotted in Fig. 3. On the BlogCatalog and Flickr, when λ increases from 0 to 1, the effect of it grows rapidly, and when λ is greater than 1, the effect is relatively flat. Therefore, in order to ensure the consistency of structure embedding and attribute embedding, we set the value of λ between 1 and 2.5. In the same way, we observe that the optimal range of on the Wiki is $0 < \lambda < 0.05$ through multiple experiments. When the value of λ increases in this range, the performance of node classification improves steadily. Particularly, when $\lambda = 0$, it means that the consistency between structure embedding and attribute embedding is not considered in the model (Eq. (3)). Generally speaking, the CCANE outperforms all these comparisons when $\lambda > 0$, which indicates that it is necessary to take into account the consistency constraints between structure and attribute when learning network embedding.



(a) BlogCatalog (b) Flickr (c) Wiki

FIGURE 3. Impact of parameter λ.

TABLE 5. Node classification results on different weight ratio of proximity.

Datasets	BlogCatalog		Flickr		Wiki	
Metrics	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
$X = X^{(1)} + X^{(2)*}$	**0.9400**	**0.9388**	**0.8446**	**0.8439**	**0.7880**	**0.6241**
$X = X^{(1)} + 2X^{(2)}$	0.9392	0.9380	0.8273	0.8263	0.7814	0.6188
$X = X^{(1)} + 3X^{(2)}$	0.9372	0.9362	0.8147	0.8137	0.7780	0.6161
$X = X^{(1)} + 4X^{(2)}$	0.9365	0.9353	0.8054	0.8045	0.7747	0.6119
$X = X^{(1)} + 5X^{(2)}$	0.9296	0.9283	0.7965	0.7955	0.7689	0.6069
$X = 2X^{(1)} + X^{(2)}$	0.9396	0.9386	0.8424	0.8411	0.7830	0.6215
$X = 3X^{(1)} + X^{(2)}$	0.9346	0.9336	0.8258	0.8245	0.7814	0.6194
$X = 4X^{(1)} + X^{(2)}$	0.9346	0.9336	0.7920	0.7906	0.7797	0.6176
$X = 3X^{(1)} + X^{(2)}$	0.9353	0.9342	0.7714	0.7700	0.7772	0.6176

2) THE PARAMETER SENSITIVITY OF THE WEIGHT RATIO OF PROXIMITY

To verify the validity of the final integrated structure proximity formula $X = X^{(1)} + X^{(2)}$, we conduct the node classification experiments where we set the weight ratio of $X^{(1)}$ and $X^{(2)}$ in the formula to be different values. Judging from the results in Table 5, the node classification works best when the weight ratio is 1, i.e., the first-order proximity is as important as the second-order proximity when calculating the structure proximity. In summary, the node embeddings can better describe the characteristics of network structure when the impact of the first-order proximity equals to that of the second-order proximity. Therefore, the final structure proximity is set to $X = X^{(1)} + X^{(2)}$.

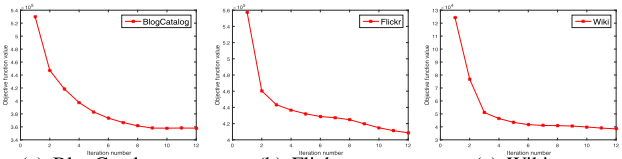
F. EFFICIENCY EVALUATION

The efficiency of the CCANE is validated on three datasets, i.e., BlogCatalog, Flickr, and Wiki through comparing the running time with TADW and AANE. The other baselines are not selected as comparisons because they do not consider attribute information when learning node embeddings. In Table 6, the running time of the CCANE and AANE is significantly less than TADW. The CCANE is slightly faster than AANE, and they cost almost the same amount of time to learn node embeddings as they all use ADMM for optimization.

TABLE 6. Running time (in seconds) of different methods.

Dataset	BlogCatalog	Flickr	Wiki
TADW	447	691	1385
AANE	21	27	10
CCANE*	**16**	**23**	**9**

Fig. 4 depicts the value of objective function concerning the number of iterations, which demonstrates the convergence of the CCANE. Observed from the performance on these three datasets, the value of objective function drops sharply in the first two iterations, and then, the trend steadily slowed down until convergence. In summary, the CCANE can converge stably and quickly when training the parameters of the model.



(a) BlogCatalog (b) Flickr (c) Wiki

FIGURE 4. The value of objective function with respect to iterations.

G. LARGE-SCALE SYNTHETIC DATASET

To assess the potential of our proposed CCANE to deal with synthetic attributed networks and large-scale data, we synthesize an attributed network and the statistics are listed in Table 7. The CCANE learns the embeddings for nodes on this network, and the learned embeddings are utilized to conduct node classification experiments, then we compare the running time of the CCANE with baselines.

TABLE 7. Statistics of synthetic dataset.

Item	Value
Nodes($\|V\|$)	22,000
Edges($\|E\|$)	459,397
Attribute(m)	22,000
Label(l)	10

The node classification results in Table 8 show that the Micro-f1 and Macro-f1 of our proposed CCANE can reach 1.0 in the node classification experiments on the synthetic dataset. Since synthetic datasets are usually more stable, the value of Micro-f1 and Macro-f1 can reach 1.0. In contrast, the real-world datasets have more uncertainty, therefore the performances are worse on them than the synthetic datasets. From the results, the performances of TADW, AANE and CCANE are better than other baselines. This proves that our algorithm CCANE can handle synthetic datasets.

Furthermore, we verify the efficiency of our proposed CCANE in processing large-scale data in terms of the running time. In Table 9, the results show that the running time of the CCANE is still less than other attributed network embedding methods in the large-scale network, which proves that the CCANE can deal with large-scale networks efficiently.

TABLE 8. Node classification results on the synthetic dataset.

Metrics	Micro-F1			Macro-F1		
Training Percentage	10%	30%	50%	10%	30%	50%
DeepWalk	0.9996	0.9995	0.9996	0.9996	0.9995	0.9996
Node2vec	0.9997	0.9999	0.9997	0.9997	0.9999	0.9997
LINE	0.9996	0.9995	0.9995	0.9996	0.9995	0.9996
TADW*	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
AANE*	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
CCANE*	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**

TABLE 9. Running time (in seconds) on the synthetic dataset.

Methods	Running time
TADW	190
AANE	99
CCANE*	**57**

H. FURTHER DISCUSSION

Experiments are conducted to testify the feasibility of the CCANE, the improvement of computing efficiency, and the ability of processing large-scale data. First, validation on node classification is conducted comparing to three network embedding methods(DeepWalk, Node2vec, and LINE) and two attributed network embedding methods(TADW and AANE). The advantages of attributed network embedding methods in the experiments indicate the importance of combining attribute information with network representation learning. In addition, the CCANE is superior to other attributed network embedding methods by comparing the results of network analysis tasks. The visualization results state that the learned embeddings through the CCANE can better retain network characteristics. Then, we prove the efficiency of the CCANE in terms of the running time and iteration times. By accelerating the optimization process, the running time of the CCANE reduced greatly, which enables the model to handle large-scale networks. And the objective function of the CCANE has a stable trend of convergence in a few steps. Finally, we synthesize a large-scale attributed network and perform node classification experiments on it, which illustrates that the CCANE can process large-scale data and synthetic datasets. In summary, our proposed attributed network representation learning method CCANE is both effective and efficient.

IV. RELATED WORK

A. NETWORK EMBEDDING

Network embedding has been a popular research topic and various methods emerge one after another in recent years. The objective of network embedding is that the node representations can retain and characterize the original features of the network structure. In view of this, some network representation learning methods are designed to describe the first-order similarity between nodes, i.e., the similarity reflected by the node pairs through the adjacency matrix. For example, spectral clustering [29] obtains the node representation of d-dimension by calculating the first d feature vectors of the normalized Laplace matrix.

With the rapid development of Natural Language Processing (NLP) [30], topological relationships between nodes can be represented by combining network representation learning with natural language models. For example, DeepWalk [12] proposed by Perozzi et al. in 2014 utilizes the random walk [13] sequence in the network to simulate nature language based on Word2vec [21], then applies the skip-gram model [14] to represent node embeddings. Grover and Leskovec [16] extend DeepWalk algorithm by introducing width- and depth-first search in the process of random walk. To describe the similarity of two nodes reflected by neighbor structure, Tang *et al.* [15] propose preserving second-order similarity and model similarity based on the joint probability distribution. In addition, some researchers have also discovered some special structures contained in network topology information. For example, Wang *et al.* [22] take the community structure contained in network topology into consideration, based on the joint non-negative matrix decomposition model, to maximize the modularity and similarity between nodes.

In the real world, nodes in networks usually contain abundant attribute information. However, existing studies are based on network topology structure solely, neglecting valuable attribute information of nodes which usually plays an essential role in characterizing network features. Subsequently, studies on attributed network embedding have received increasing attention.

B. ATTRIBUTED NETWORK EMBEDDING

Recently, studies on network embedding are developed by taking into account the attribute information of nodes, which can make full use of network information.

Yang *et al.* [17] first prove that the embeddings learned by DeepWalk are equivalent to the sum of the first k-order state transition matrices obtained by matrix decomposition. One can obtain the network representations i.e., a low-dimensional matrix, which depicts both topology structure and node text attributes by taking the text attribute characteristics as input of matrix decomposition. However, subsidiary information considered in this model is limited to the attributes information of text. Moreover, it is incapable of handling large-scale networks. Huang et al. propose an optimization method AANE [18] to learn node representations for large-scale networks, and the solutions are obtained by maximizing and minimizing the similarity and difference between directly connected nodes, respectively. However, it merely considers the similarity of nodes directly connected by edges, ignoring

the similarity of nodes sharing similar neighbor structure but without a direct link. He *et al.* [32], [33] adopt the scheme of multi-view learning [31] for processing attributed network. However, they are task-specific aiming at discovering clusters in graphs, therefore they are limited for handling other network analysis tasks.

As deep learning [26] gets more and more attention, studies on attributed network representation learning based on deep learning have emerged. To integrate structure and attribute information when learning node representation, Liao *et al.* [23] propose putting the structure and attribute information into the same deep learning model and train parameters together, by maximizing the link probability between nodes on the basis of the similarity of node representation. Gao and Huang [24] propose considering the structure and node attribute information when training deep auto-encoders with consistency and complementarity constraints, and then, concatenate the embeddings obtained to form the final representation. Kipf and Welling [27] adopt convolutional neural network to learn coded local network structure and latent representations of node attribute information.

V. CONCLUSION AND FUTURE WORK

This paper presents an attributed network embedding model CCANE, which can learn low-dimensional vectors for nodes incorporating attribute information. The main objective is to obtain node embeddings through matrix factorization considering both structure proximity and attribute proximity. Moreover, the final embeddings of nodes have the ability to guarantee the consistency and complementarity. To improve the scalability of the model, the complex optimization process is decomposed into several sub-tasks in parallel. In the experiments, the feasibility and superiority of the CCANE are validated through network analysis tasks comparing to the state-of-the-arts. In the future, incorporating label information into semi-supervised methods when learning attributed network representation will be a new direction, since partially labeled information of nodes can provide more useful information beyond attribute information.

REFERENCES

[1] A. Alarifi, M. Alsaleh, and A. Al-Salman, "Twitter turing test: Identifying social machines," *Inf. Sci.*, vol. 372, pp. 332–346, Dec. 2016.
[2] X. Huang, J. Jiang, B. Choi, J. Xu, and Z. Zhang, "PP-DBLP: Modeling and generating attributed public-private networks with DBLP," in *Proc. IEEE Int. Conf. Data Mining Workshops*, Nov. 2018, pp. 986–989.
[3] L. Jian, J. Li, and H. Liu, "Toward online node classification on streaming networks," *Data Mining Knowl. Discovery*, vol. 32, no. 1, pp. 231–257, Jan. 2018.
[4] J. Tang, J. Liu, M. Zhang, and Q. Mei, "Visualizing large-scale and high-dimensional data," in *Proc. 25th Int. Conf. World Wide Web*, 2016, pp. 287–297.
[5] J. Yang, J. Mcauley, and J. Leskovec, "Community detection in networks with node attributes," in *Proc. IEEE 13th Int. Conf. Data Mining*, Dec. 2013, pp. 1151–1156.
[6] A. K. Bhowmick, K. Meneni, and B. Mitra, "On the network embedding in sparse signed networks," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*, Cham, Switzerland, 2019, pp. 94–106.
[7] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1225–1234.
[8] J. Tuke and M. Roughan, "All networks look the same to me: Testing for homogeneity in networks," 2015, *arXiv:1512.00877*. [Online]. Available: https://arxiv.org/abs/1512.00877
[9] D. Kuang, C. Ding, and H. Park, "Symmetric nonnegative matrix factorization for graph clustering," in *Proc. SIAM Int. Conf. Data Mining*, 2012, pp. 106–117.
[10] X. Huang, J. Li, N. Zou, and X. Hu, "A general embedding framework for heterogeneous information learning in large-scale networks," *ACM Trans. Knowl. Discovery Data*, vol. 12, no. 6, p. 70, Oct. 2018.
[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.
[12] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 701–710.
[13] F. Spitzer, *Principles of Random Walk*. New York, NY, USA: Springer, 1964.
[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[15] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 1067–1077.
[16] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 855–864.
[17] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Chang, "Network representation learning with rich text information," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015.
[18] X. Huang, J. Li, and X. Hu, "Accelerated attributed network embedding," in *Proc. SIAM Int. Conf. Data Mining*, 2017, pp. 633–641.
[19] X. Shen, X. Zhu, X. Jiang, and L. Gao, "Visualization of non-metric relationships by adaptive learning multiple maps t-SNE regularization," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2017, pp. 3882–3887.
[20] A. Levordashka and S. Utz, "Ambient awareness: From random noise to digital closeness in online social networks," *Comput. Hum. Behavio*, vol. 60, pp. 147–154, Jul. 2016.
[21] Y. Goldberg and O. Levy, "word2vec Explained: Deriving Mikolov et al.'s negative-sampling word-embedding method," 2014, *arXiv:1402.3722*. [Online]. Available: https://arxiv.org/abs/1402.3722
[22] X. Wang, P. Cui, and J. Wang, "Community preserving network embedding," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017.
[23] L. Liao, X. He, H. Zhang, and T.-S. Chua, "Attributed social network embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2257–2270, Dec. 2018.
[24] H. Gao and H. Huang, "Deep attributed network embedding," in *Proc. IJCAI*, 2018, pp. 3364–3370.
[25] X. Huang, J. Li, and X. Hu, "Label informed attributed network embedding," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, Feb. 2017, pp. 731–739.
[26] S. Pouyanfar, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Comput. Surv.*, vol. 51, no. 5, p. 92, 2018.
[27] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: https://arxiv.org/abs/1609.02907
[28] 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.
[29] X. Wu and W. Sun, "A semi-supervised locally linear embedding spectral clustering algorithm," *Int. J. Adv. Media Commun.*, vol. 7, no. 2, pp. 101–112, 2017.
[30] Y. Adi, E. Kermany, Y. Belinkov, and O. Lavi, "Analysis of sentence embedding models using prediction tasks in natural language processing," *IBM J. Res. Develop.*, vol. 61, no. 4, pp. 1–9, 2017.
[31] S. Sun, "A survey of multi-view machine learning," *Neural Comput. Appl.*, vol. 23, nos. 7–8, pp. 2031–2038, 2013.
[32] T. He, Y. Liu, and T. H. Ko, "Contextual correlation preserving multiview featured graph clustering," *IEEE Trans. Cybern.*, to be published.
[33] T. He, K. C. Chan, and L. L. Yang, "Clustering in networks with multi-modality attributes," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Dec. 2018, pp. 401–406.

XUAN ZANG received the B.Sc. degree from the College of Computer Science and Technology, Jilin University, China, in 2018, where she is currently pursuing the master's degree. She is also a Graduate Student with the Key Laboratory of Symbolic Computation and Knowledge Engineering Attached to the Ministry of Education, Jilin University. Her current research interests include network representation learning and heterogeneous data mining.

BO YANG received the B.Sc., M.S., and Ph.D. degrees in computer science from Jilin University. He is currently a Professor with the College of Computer Science and Technology, Jilin University. He is also the Director of the Key Laboratory of Symbolic Computation and Knowledge Engineer, Ministry of Education, China. He has published more than 90 articles on international journals, including the IEEE TPAMI, the IEEE TKDE, the IEEE TCYB, ACM TWEB, DKE, JAAMAS, and KBS and international conferences, including IJCAI, AAAI, ICDM, WI, PAKDD, and ASONAM. His current research interests are in the areas of data mining, machine learning, knowledge engineering, and complex/social network modeling and analysis. He has served as an Associate Editor and a Peer Reviewer for several international journals and the Program Chair and the SPC for several international conferences.

SHUANG YANG received the B.Sc. and M.S. degrees from the College of Computer Science and Technology, Jilin University, China, in 2016 and 2019, respectively. She is currently a Graduate Student with the Key Laboratory of Symbolic Computation and Knowledge Engineering Attached to the Ministry of Education, Jilin University. Her current research interests include complex networks and data mining.

HECHANG CHEN received the M.S. and Ph.D. degrees from the College of Computer Science and Technology, Jilin University (JLU), China, in 2014 and 2018, respectively. He is currently an Assistant Professor with the School of Artificial Intelligence, JLU. He was enrolled in the University of Illinois at Chicago (UIC), as a joint training Ph.D. Student, from 2015 to 2016, and Hong Kong Baptist University (HKBU), as a Visiting Student, from July 2017 to January 2018. He has published more than 20 articles in international journals and conferences, including the IEEE TPAMI, KBS, ICDE, ICDM, PAKDD, and KSEM. His current research interests include the areas of data mining, machine learning, and complex/social network analysis. He has served as a Peer Reviewer for several international journals and conferences, including the IEEE TKDE, the IEEE TCYB, the IoT, the IEEE TITS, the IEEE TKDD, IEEE Access, *Neurocomputing*, IODP, KDD, and KSEM.