Inter-Intra Information Preserving Attributed Network Embedding

KAI WANG1,2, LEI XU1,2, LING HUANG1,2, CHANG-DONG WANG1,2, YONG TANG3, AND CHENGZHOU FU3

1School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China
2Guangdong Province Key Laboratory of Computational Science, Guangzhou 510275, China
3School of Computer Science, South China Normal University, Guangzhou 510631, China

Corresponding author: Ling Huang (huanglinghl@hotmail.com)

This work was supported in part by Grant N00590/89985 from the National Research Foundation (NRF) of South Africa SARCHI ASN Chair, along with a Faculty Development Grant from the Tarleton State University.

ABSTRACT To alleviate the problem caused by the sparsity of network structure which is often the case in large-scale network, attributed network embedding has attracted an increasing amount of attention. Some existing attributed network embedding models integrate linkage structure and node attribute by adding a consistency criterion on the structure representation and attribute representation, whereby the similarity between them can be ensured. However, to enforce the structure and attribute representations to be similar may cause the information distortion due to the inherent difference between these two kinds of information. Additionally, the existing models mainly focus on learning the inter-relation between structure and attribute information, while the intra-characteristic of each information is ignored, leading to the information loss. To address the above two problems, we propose a novel model named Inter-Intra Information Preserving Attributed Network Embedding (IINE) to effectively learn the node representations of attributed network, which can not only capture the inter-relation between structure and attribute information with less information distortion but also effectively preserve the intra-characteristic of each information. The proposed model is composed of a primary model named coupled autoencoder and two auxiliary models named structure miner and attribute miner. The coupled autoencoder trains the node representation by smoothly combining both structure and attribute information, while the structure miner and attribute miner are utilized to further mine the intra-characteristic from the corresponding information so as to assist the primary model. The Extensive experiments are conducted on seven real-world datasets, and the results confirm the superior performance of IINE over several state-of-the-art models.

INDEX TERMS Network embedding, attributed network, inter-relation, intra-characteristic.

I. INTRODUCTION

Nowadays, many kinds of information are represented in the form of network. For instance, the following relationship between users is represented by social network, e.g. Weibo,1 and the citation relationship between papers is represented by paper citation network, e.g. DBLP.2 Due to the large-scale and complexity of many networks, it is intractable for manually analyzing the meaningful knowledge inside them, which makes the importance of network analysis [1]. However, the traditional network analysis tasks, e.g. node classification, network clustering [2], graph classification [3], [4], and recommendation [5] are becoming more and more challenging due to problems like data sparsity, curse of dimensionality caused by the explosive growth of network scale. In this case, the network embedding method, which aims to learn effective low-dimensional network representation [6]–[11], [11]–[13], plays a key role in solving the network analysis problems. For instance, in [14], an effective network embedding model is proposed to integrate different kinds of information in heterogeneous network to learn node representations, which can both reduce the dimensionality and enrich the knowledge inside representations. Besides, network embedding can be useful in real life. As one of the most popular social networks, there are more than 300 million users in weibo, while one user may only interact with hundreds of them, which makes the traditional network representation methods like adjacent matrix be memory-wasting and inefficiency, due to the inevitable
problems of high-dimensional and sparsity. In this case, network embedding algorithms aim to learn a low-dimensional and dense representation that can not only reduce the computational complexity but also better reflect the specific feature of different users.

Conventional network embedding models mainly focus on exploiting the network structure information during the learning of node representations. Some of them attempt to preserve the global proximity of network structure [6], [7], [15], some of them pay more attention to preserve the local proximity of each node [8], [16], and some of them tend to generally learn both the global and local proximity by jointly integrating the lower- and higher-order proximities to generate the node representations [17]–[20]. However, due to the increasing scale of networks, most of them suffer seriously from the sparsity of network structure. Take the Citeseer dataset3 as an example, there are 3,312 nodes but only 4,732 edges in this network, which means each node is only connected with few neighbors on average. In this case, there is a great challenge to learn efficient node representations due to such sparse structural information. Fortunately, in many sparse networks, the nodes are associated with specific attributes, such as user profile in social network and paper abstract in citation network [21], [22]. These attributes contain rich semantic information which can reflect the inherent property of the corresponding node [14]. Therefore, the data sparsity problem of node representation learning can be efficiently alleviated by properly utilizing both the node structure information and network attribute information.

One of the most intuitive ideas is to separately train structure representation and attribute representation and then concatenate them to generate the final representation. However, according to [9], this method cannot learn the inter-relation between structure and attribute information. Recently, some attempts have been made for training node representations by coupling network structure and node attribute information [9], [10], [23]–[27]. However, many existing models integrate those two kinds of information by keeping the consistency between node structure representation and node attribute representation, which is usually implemented by adding a constraint to ensure these two kinds of representation to be similar. Nevertheless, the structure information of each node reflects the interacting relationship between the node and its neighbors, while the attribute information represents the inherent feature of each node. From this perspective, consistency criterion may cause the information distortion, for the reason that it enforces the representations learned from different information to be similar. Besides, the existing attributed network embedding methods mainly focus on learning how to combine these two kinds of information so as to obtain the inter-relation between them, which ignore the significance of preserving the intra-characteristic inside each single information. Therefore, it remains an open challenge how to not only incorporate both structure and attribute information to capture the inter-relation without causing the information loss but also preserve the intra-characteristic of each single information simultaneously when learning node representations.

In this paper, we provide a solution by proposing a novel network embedding model named Inter-Intra Information Preserving Attributed Network Embedding (IINE). IINE is composed of a primary model named coupled autoencoder and two auxiliary models named structure miner and attribute miner. The coupled autoencoder integrates both structure and attribute information to generate node representations which can capture the inter-relation between them with less information distortion. In the meanwhile, the structure miner and attribute miner focus on learning node structure representations and node attribute representations from the corresponding information respectively so as to further capture the intra-characteristic. In particular, the representations learned by structure miner and attribute miner are considered as the auxiliary representations of the node representations learned by the coupled autoencoder, which can effectively help preserve the intra-characteristic when learning the inter-information.

The main contributions of this paper are summarized as follows:

- We define a novel and meaningful problem which is to not only capture the inter-relation between structure and attribute information with less information distortion but also preserve the intra-characteristic of each single information.
- To solve the above new problem, a novel model called IINE is proposed, which is composed of a primary model named coupled autoencoder and two auxiliary models named structure miner and attribute miner.
- Two case study experiments are conducted to investigate the influence of intra-characteristic and consistency criterion on the attributed network embedding problem, and the results empirically confirm the significance of the novel problem studied in this paper.
- Extensive experiments are conducted on seven real-world datasets through the classification and clustering tasks to demonstrate the superior performance of IINE.

The rest of this paper is organized as follows. We briefly introduce related work in section II. Then some important notations and formal definitions of the studied problem are presented in Section III. Section IV describes in detail the proposed IINE model. The experimental results are reported in Section V. Finally, the conclusion is drawn in Section VI.

II. RELATED WORK

Network embedding, which aims to learn low-dimensional representations for networks, has attracted an increasing amount of attention in recent years. The history of network embedding can be tracked back to the eigen-based methods such as LLE [28], IsoMap [29] and Laplacian Eigenmaps [30]. These works represent nodes in lower

3http://linqs.cs.umd.edu/projects/projects/lbc/index.html
embedding methods adopt consistency criterion to integrate both the attribute information and the structural information. Node representations in ANEM [39] are required to preserve the consistency between them. Furthermore, DANE [38] adopts two deep Autoencoder to train structure and attributed features separately. Besides, DANE [38] adopts two deep Autoencoder to train structure and attributed features separately. Then keep the consistency between them. For keeping the local structure of network, Line [8] focuses on preserving the neighbor structure of each node by proposing the first- and second-order proximity to constrain the training process. Based on the equivalence between matrix factorization and skip-gram based embedding as proved in [32], both GraRep [17] and M-NMF [18] utilize the matrix factorization to integrate both the global and local structure into node representation. GraRep [17] preserves them by concatenating the representations obtained by factorizing several lower and higher orders’ node similarity matrices. The M-NMF model [18] obtains low-dimensional node representations which can preserve both the community structure and node local proximity simultaneously. With the development of deep models, DNGR [33] and SDNE [34] utilize deep autoencoder to capture network’s non-linearity in learning node representations. In [35], an attention-based network embedding model is proposed to exploit the social structures for user alignment. Furthermore, ANE [36] adopts the adversarial learning principle of GAN [37] to train the node representations.

However, the aforementioned methods only exploit the topological structure information of networks, which may suffer seriously from the data sparsity problem. To alleviate this problem, node attribute information, which contains rich semantic information, is integrated into the network embedding model. The recently developed attributed network embedding models [10], [25], [38], [39] empirically prove that the performance of representation learning can be improved by taking into account the node attributes. In TADW [25], the textual features are integrated into the matrix factorization so as to guide the node representations to learn attribute information. LANE [10] jointly adopts the label, attribute, and structure information to learn node representations. Besides, DANE [38] adopts two deep Autoencoders to train structure and attributed features separately and then keep the consistency between them. Furthermore, node representations in ANEM [39] are required to preserve both the attribute information and the structural information including the local proximity structure and mesoscopic community structure. Despite success, many attributed network embedding methods adopt consistency criterion to integrate both structure and attribute information, which may cause the information distortion. Additionally, the intra-characteristic of each information is ignored, leading to the information loss.

III. PROBLEM FORMULATION

In this section, we will present the important notations and formally define the studied problem.

A. NOTATIONS

Let \( \mathcal{G} = \{\mathcal{V}, \mathcal{F}, \mathcal{E}\} \) denote an attributed network, where \( \mathcal{V} = \{v_1, v_2, \ldots, v_n\} \) and \( \mathcal{F} = \{f_1, f_2 \ldots, f_d\} \) represent the node set and the attribute set of attributed network respectively. Mathematically, two important information matrices are introduced to describe those information as follows. One of them is the structure proximity matrix \( \mathbf{A} \in \mathbb{R}^{n \times n} \), which is composed of a series of structure matrices of different orders. Each element \( a_{ij} \) represents the transition probability between node \( i \) and node \( j \), and the sum of each row vector \( \mathbf{a}_i \) is equal to 1. In practice, we sum up the first-, second- and third-orders’ structure matrices and then normalize the matrix to obtain the structure proximity matrix. The other is the attribute matrix \( \mathbf{P} \in \mathbb{R}^{n \times d} \), where each element \( p_{ij} \) represents whether node \( i \) contains the \( j \)-th attribute \( f_j \), and each row \( \mathbf{p}_i \) is the attribute vector of each node. Finally, the definition of edge set is \( \mathcal{E} = \{(i, j) | \text{if node } i \text{ is connected with node } j\} \).

B. PROBLEM DEFINITION

Different from the network consisting of only topological structure, nodes in attributed network are usually associated with different attributes, for instance, user profile in social network, paper abstract in citation network, and so on. Therefore, for better analyzing the real-world network, attributed network embedding is of vital significance.

Definition 1 (Attributed Network Embedding): Given the structure proximity matrix and attribute matrix \( \mathbf{A} \) and \( \mathbf{P} \) of the graph \( \mathcal{G} \), the attributed network embedding is to learn an appropriate model which can map both of structure and attribute information into low-dimensional network representation \( \mathbf{H} \in \mathbb{R}^{n \times k} \), where each row vector \( \mathbf{h}_i \in \mathbb{R}^{1 \times k} \) is the \( k \)-dimensional representation of node \( i \).

How to learn efficient node representations from these two kinds of information is the key to attributed network embedding. The existing attributed network embedding methods can be generally categorized as two classes. One of them is the consistency based network embedding model like [10], and another is combination based network embedding model like [9].

Definition 2 (Consistency Based Attributed Network Embedding): Consistency based Attributed Network Embedding adopts a similarity constraint on structure and attribute representations of the same node, so as to keep the consistency between structure information and attribute information.

Definition 3 (Combination Based Attributed Network Embedding): Combination based Attributed Network
Embedding aims at combining two kinds of information into the learned representation, which usually adopts a combination function to incorporate the structure representation with attribute representation.

However, both kinds of methods would to some extent mislead node embeddings to lose information, where the first model trains the final representation by forcing the latent representations of two different kinds of information to be similar, while the second model pays much attention to learn the inter-relation between structure information and attribute information while ignoring the intra-characteristic of each single information. Based on this, we define a novel problem called ‘Inter-Intra Information Preserving Attributed Network Embedding’ as follows:

**Definition 4 (Inter-Intra Information Preserving Attributed Network Embedding):**

Given the structure proximity matrix $A$ and the attribute matrix $P$, the Inter-Intra Information Preserving Attributed Network Embedding is to smoothly incorporate $A$ and $P$ to train the low-dimensional representation $H$ that can not only capture the inter-relation between two kinds of information with less information distortion but also preserve the intra-characteristic of each single information.

**IV. THE PROPOSED MODEL**

For solving the above problem, we propose the IINE model, whose general architecture is shown in Figure 1. As shown in the dashed line rectangle, the coupled autoencoder is proposed to integrate the structure and attribute information to generate the node representations which can capture the inter-relation between them with less information distortion.

For preserving the intra-characteristic, two auxiliary models named **structure miner** and **attribute miner** are established to further mine the intra-characteristic of the corresponding information so as to assist the coupled autoencoder.

**A. COUPLED AUTOENCODER**

In attributed network, structure information and attribute information describe the network from two different aspects. For smoothly integrating the structure and attribute information to train the node representations, we propose a novel unsupervised model named **Coupled Autoencoder**. Without loss of generality, the coupled autoencoder consists of encoding and decoding processes:

1) **ENCODING**

The object of the encoding process is to mine the latent information of network structure and node attribute. To this end, both of the structure and attribute information, namely the structure proximity matrix $A$ and the attribute matrix $P$, are firstly encoded as the latent structure representation $U^A \in \mathbb{R}^{n \times \frac{k}{2}}$ and latent attribute representation $U^P \in \mathbb{R}^{n \times \frac{k}{2}}$ by using corresponding encoding functions. Then, the node representation $H$ is generated by the weighted combination of $U^A$ and $U^P$, where the combination weight is set as $\gamma$.

Although there are many rational combination methods, for preserving information as much as possible, the concatenating combination is adopted. Therefore, the encoding process can be formulated as follows:

$$U^A = f^A(A; \theta^A)$$  \hspace{1cm} (1)

$$U^P = f^P(P; \theta^P)$$  \hspace{1cm} (2)
where \( \theta^A \) and \( \theta^P \) are the parameters of the structure encoder and the attribute encoder respectively.

Then the node representation is generated by

\[
H = [\gamma U^A; (1 - \gamma)U^P] \tag{3}
\]

2) DECODING

As shown in the upper part of dashed line rectangle of Figure 1, during the decoding process, the decoder should reconstruct both structure and attribute information from \( H \) by using different decoding functions, which can be formulated as follows:

\[
\tilde{A} = g^A(H; \phi^A) \tag{4}
\]

\[
\tilde{P} = g^P(H; \phi^P) \tag{5}
\]

where \( \phi^A \) and \( \phi^P \) represent the parameters of the structure decoder and the attribute decoder respectively.

In summary, the encoding process maps both the structure and attribute information into a latent space and combines them to obtain node representations \( H \). In contrast to the encoding process, the decoding process is to decode the node representation \( H \) back to the original structure and attribute spaces by using different decoding functions so as to reconstruct both kinds of information. The more complete information the node representations can reconstruct, the more representative they are. Therefore, the reconstruction losses from structure information and attribute information are adopted to guide the training of the coupled autoencoder:

\[
L_r = \| (\tilde{A} - A) \|_F + \| (\tilde{P} - P) \|_F \tag{6}
\]

However, due to the sparsity issue of the given network, the reconstruction error is mainly controlled by zero elements. For balancing the losses caused by zero and non-zero elements, the confident matrix [40] should be taken into consideration:

\[
L_r = \| (\tilde{A} - A) \odot C^A \|_F + \| (\tilde{P} - P) \odot C^P \|_F \tag{7}
\]

where \( \| \cdot \|_F \), \( \odot \), \( C^A \in \mathbb{R}^{n \times n} \) and \( C^P \in \mathbb{R}^{n \times d} \) denote the Frobenius norm, element-wise dot product, structure confident matrix and attribute confident matrix respectively. In \( C^A \), each element \( c^A_{ij} = \beta \) if \( a_{ij} > 0 \), and \( c^A_{ij} = 0 \) otherwise. Similarly, \( c^P_{ij} = \beta \) if \( p_{ij} = 1 \), and \( c^P_{ij} = 0 \) otherwise.

B. STRUCTURE MINER AND ATTRIBUTE MINER

Although the coupled autoencoder can guide node representations to learn both structure information and attribute information, the intra-characteristic of each single information like the local proximity may be missed during the training process. For addressing this problem, we propose the structure miner and the attribute miner to further mine the intra-characteristic inside the corresponding information so as to assist the training of node representations.

1) STRUCTURE MINER

As shown in the left part of Figure 1, the structure miner is a deep neural network. This model takes \( A \) as input and encodes it into the latent space to obtain the structure auxiliary representation \( \tilde{V}^A \in \mathbb{R}^{n \times k} \), which is able to keep the intra-characteristic of structure information at both global and local aspects. Globally, the \( \tilde{A} \) decoded from \( \tilde{V}^A \) is expected to be able to reconstruct the original topological structure. Locally, node representations in \( \tilde{V}^A \) are supposed to reflect the proximity between a node and its neighbors, and the dissimilarity between a node and a part of the non-neighbors. Therefore, the loss of structure miner is composed of both the reconstruction loss and the negative-sampling based pair-wise proximity loss:

\[
L_s = \| (\tilde{A} - A) \odot C^A \|_F - \sum_{(i,j) \in \mathcal{E}} a_{ij} \log \sigma((v_i^A)^T v_j^A) - \sum_{(i,j) \in \mathcal{N} \mathcal{E}} \log \sigma(1 - (v_i^A)^T v_j^A) \tag{8}
\]

where \( \mathcal{N} \mathcal{E} \) is a sampling set of \( (i, j) \notin \mathcal{E} \), \( \sigma(\cdot) \) is the sigmoid function, and \( v_i^A \) is the structure auxiliary representation of the \( i \)-th node respectively. As we can see, to minimize this loss is equivalent to minimizing the reconstruction loss of structure information, minimize the dissimilarity (the minus of similarity) between neighbor nodes, and maximize the dissimilarity between the negative sampled non-negative node pairs.

2) ATTRIBUTE MINER

The attribute miner aims to learn node attribute representations which can further learn the intra-characteristic of attribute information. Similar to the structure auxiliary representation, the attribute auxiliary representation \( \tilde{V}^P \in \mathbb{R}^{n \times k} \) is expected to learn both global attribute information reconstruction and local proximity-keeping abilities. However, unlike the structure proximity matrix, the attribute matrix cannot directly reflect the pairwise relationship between each node. Fortunately, it is intuitive that nodes usually share similar features with their neighbors, while non-neighbors may not. For instance, the connected papers in paper citation network share similar research topics, while a part of the disconnected papers are likely to be in different research areas. Inspired by this circumstance, it is better to keep the attribute representations of nodes and their neighbors to be close while keeping at a distance for some of non-neighbors.

By integrating with the negative sampling method, the loss function of the attribute miner can be written as follows:

\[
L_p = \| (\tilde{P} - P) \odot C^P \|_F - \sum_{(i,j) \in \mathcal{E}} a_{ij} \| v_i^P - v_j^P \|_2 + \sum_{(i,j) \in \mathcal{N} \mathcal{E}} \| v_i^P - v_j^P \|_2 \tag{9}
\]

where \( v_i^P \) is the attribute auxiliary representation of the \( i \)-th node. Similar to the loss of structure miner, optimizing the loss of attribute miner is to minimize the reconstruction loss of attribute information and the dissimilarity (the Euclidean
distance) between neighbor nodes, while maximizing the dissimilarity between the negative sampled non-negative node pairs.

C. INTER-INTRA INFORMATION PRESERVING ATTRIBUTED NETWORK EMBEDDING

By incorporating the coupled autoencoder with the structure miner and the attribute miner, the IINE model is established. In IINE, the reconstruction loss from the coupled autoencoder will guide the node representations to learn the coupled information. Meanwhile, two auxiliary representations $V^A$ and $V^P$ generated from the structure miner and the attribute miner are utilized to guide the structure component $U^A$ and the attribute component $U^P$ of node representations to preserve the intra-characteristic of each single information, and the auxiliary loss can be written as follows:

$$L_a = \sum_i \|u^A_i - v^A_i\|^2 + \sum_i \|u^P_i - v^P_i\|^2$$  \hspace{1cm} (10)

where $u^A_i$ and $u^P_i$ are the structure component and the attribute component of the representation of the $i$-th node respectively.

Finally, the objective function of our IINE model can be written as follows:

$$L = L_r + L_s + L_p + \alpha L_a$$  \hspace{1cm} (11)

where $\alpha$ is the weight to control the strength of auxiliary loss. In our implementation, the loss function in Eq. (11) is optimized by the Adam [41] function in Python.

V. EXPERIMENTS

In this section, extensive experiments are conducted on seven publicly available networks. First of all, we introduce in detail the experimental settings, including the datasets and evaluation measures. Secondly, parameter analyses on both classification and clustering tasks are conducted to investigate the sensitivity of the parameters on the IINE model. Then, comparison experiments are conducted to compare the proposed IINE model with several existing network embedding models, and the results confirm the superiority of the proposed model. As an auxiliary experiment, visualization of the embedding results on the Citeseer dataset generated by different algorithms is also presented. Finally, two case study experiments are conducted to investigate the influence of intra-characteristic and consistency criterion on the attributed network embedding tasks, and the results empirically confirm the significance of the novel problem studied in this paper.

A. DATASETS

In experiment, we adopt seven publicly available attributed networks which are Citeseer network [42], Wiki network [43], Terrorist network [44], and four subnetworks (Cornell, Texas, Washington, Wisconsin) from WebKB network. The datasets are described as follows:

1) Citeseer: This dataset is a citation network which contains 3,312 scientific publications and 4,732 citation links, where each publication is associated with a 3,703-dimensional binary word vector. In addition, those publications are from 6 distinct research areas.

2) Wiki: This dataset is a web network, containing 2,405 nodes and 17,981 links, where each node and link represent the web page and hyperlink respectively. Totally, those web pages are from 17 categories, and each web page is associated with a 4,973 dimensional tf-idf content feature vector.

3) Terrorist: This dataset is the attack network having 1,293 incidents and 3,172 links, where each node belongs to one of the 6 terrorist communities. Each attack is associated with 106-dimensional 0-1 vector.

4) WebKB: This dataset is composed of the hypertexts of four universities, which are Cornell, Texas, Washington and Wisconsin, with each of them being a subnetwork. Generally, there are 877 nodes and 1,608 edges, where each node and edge are the web page and hyperlink respectively. Besides, each web page is associated with a 1,703-dimensional word vector and belongs to one of the 5 communities which are Student, Course, Staff, Project and Faculty. Specifically, Cornell has 195 web pages and 304 hyperlinks, Texas has 187 web pages and 328 hyperlinks, Washington has 230 web pages and 446 hyperlinks, and Wisconsin has 265 web pages and 530 hyperlinks.

B. EVALUATION MEASURES

The performance of each network embedding model is evaluated in the tasks of node classification and node clustering where the LinearSVM classifier and the $k$-means clustering are used respectively. In addition to the classification and clustering results, the 2-d visualization graph is also adopted to assess the quality of the embeddings generated by each algorithm.

1) CLASSIFICATION

During the classification task, 20%, 50%, 80% of labels are used as the training data to guide the node representations to learn a proper classifier, where the remaining labels are used as the testing data. The classification results are evaluated in terms of both Micro-F1 (Mic-F1) and Macro-F1 (Mac-F1) scores [6] when comparing the predicted labels with the groundtruth labels. The calculations of Mic-F1 and Mac-F1 are presented as follows:

$$\text{Mic-F1} = 2 \times \frac{\text{recall}_{mi} \times \text{precision}_{mi}}{\text{recall}_{mi} + \text{precision}_{mi}}$$  \hspace{1cm} (12)

$$\text{Mac-F1} = 2 \times \frac{\text{recall}_{ma} \times \text{precision}_{ma}}{\text{recall}_{ma} + \text{precision}_{ma}}$$  \hspace{1cm} (13)

4http://www.cs.cmu.edu/~webkb/

5http://scikit-learn.org/stable/
where

\[
\text{recall}_{mi} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} FN_i}
\]

\[
\text{precision}_{mi} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} FP_i}
\]

\[
\text{recall}_{ma} = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_i}{TP_i + FN_i}
\]

\[
\text{precision}_{ma} = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_i}{TP_i + FP_i}
\]

with \( N \) being the number of nodes. Generally, the higher values of Mic-F1 and Mac-F1 scores indicate the better performance of the classification result.

2) CLUSTERING

In the clustering task, the number of clusters is set to be the label number of the corresponding dataset. The clustering results are evaluated by both Purity and Normalized Mutual Information (NMI).

\[
P\text{urity}(\Omega, C) = \frac{1}{N} \sum_{i=1}^{k} \max_{j} |\omega_i \cap c_j|
\]

where \( \Omega = \{w_1, \ldots, w_k\} \), \( C = \{c_1, \ldots, c_k\} \) are the set of clustering result, and groundtruth label respectively.

\[
\text{NMI} = \frac{I(\Omega; C)}{[H(\Omega) + H(C)]/2}
\]

where \( I(\cdot; \cdot) \), \( H(\cdot) \) are the mutual information and the entropy functions respectively. The calculation of \( I(\Omega; C), H(\Omega) \) and \( H(C) \) are listed as follows:

\[
I(\Omega; C) = \sum_{i} \sum_{j} \frac{|\omega_i \cap c_j|}{N} \log \frac{N|\omega_i \cap c_j|}{|\omega_i||c_j|}
\]

\[
H(\Omega) = -\sum_{i} \frac{|w_i|}{N} \log \frac{|w_i|}{N}
\]

\[
H(C) = -\sum_{i} \frac{|c_i|}{N} \log \frac{|c_i|}{N}
\]

The higher values of Purity and NMI [45] indicate better performance of the clustering result.

C. PARAMETER ANALYSIS

We first study the performance of IINE in classification (80% samples as training set) and clustering tasks in terms of different values of model parameters \( \alpha, \beta \) and \( \gamma \), which are related to the weight of auxiliary loss, the confidence value of reconstruction loss, and the combination weight of embeddings respectively. The range of three parameters are set as \( 0.1 \leq \alpha \leq 0.9, 1 \leq \beta \leq 9, \) and \( 0.1 \leq \gamma \leq 0.9 \) respectively.

FIGURE 2. Parameter analysis on \( \alpha \) in the classification and clustering tasks. (a) Classification in terms of Mic-F1. (b) Classification in terms of Mac-F1. (c) Clustering in terms of Purity. (d) Clustering in terms of NMI.
K. Wang et al.: Inter-Intra Information Preserving Attributed Network Embedding

FIGURE 3. Parameter analysis on $\beta$ in the classification and clustering tasks. (a) Classification in terms of Mic-F1. (b) Classification in terms of Mac-F1. (c) Clustering in terms of Purity. (d) Clustering in terms of NMI.

1) ANALYSIS OF $\alpha$

The influence of the weight parameter $\alpha$ is investigated at first. By fixing $\gamma = 0.6$ and $\beta = 6$, Figure 2(a) and Figure 2(b) show the variation of Mic-F1 and Mac-F1 scores while Figure 2(c) and Figure 2(d) show the variation of Purity and NMI values as functions of $\alpha$ in classification and clustering tasks respectively. As shown in Figure 2(a) and Figure 2(b), in classification task, the Mic-F1 and Mac-F1 scores of ‘Citeseer’, ‘Wiki’ and ‘Terrorist’ networks are insensitive to the value of $\alpha$, while those of the remaining four networks are affected more obviously. What’s more, the classification results tend to be better at the edge of the setting interval (about $\alpha \leq 0.3$ or $\alpha \geq 0.6$). On clustering performance, as shown in Figure 2(c) and Figure 2(d), the clustering results of each dataset are better at the middle of the setting interval (about $0.2 \leq \alpha \leq 0.8$). This demonstrates that the performance of classification and clustering can be significantly enhanced by considering a proper auxiliary loss (e.g. $0.2 \leq \alpha \leq 0.3$ or $0.6 \leq \alpha \leq 0.8$), but when it becomes too large, the major loss may be adversely affected, which weakens the embedding performance.

2) ANALYSIS OF $\beta$

We then analyze the influence of the confidence parameter $\beta$. By fixing $\alpha = 0.6$ and $\gamma = 0.6$, the variation of the Mic-F1 and Mac-F1 values in the classification task as a function of $\beta$ are shown in Figure 3(a) and Figure 3(b), while Figure 3(c) and Figure 3(d) present the influence of $\gamma$ in the clustering task in terms of Purity and NMI. From Figure 3(a) and Figure 3(b), it can be observed that when $\beta \leq 3$, the performance of classification in terms of both Mac-F1 and Mic-F1 grows dramatically, and then it tends to be stable as $\beta$ increases. As can be seen from Figure 3(c) and Figure 3(d), the performance of clustering increases or decreases dramatically at the edge of interval ($\beta \leq 3$ or $\beta \geq 7$), while it keeps a relatively better and stable state in the middle of interval ($4 \leq \beta \leq 6$). This shows that it is suitable to increase the confidence value for assigning a larger loss on the non-zero elements so as to enhance the data utilization of sparse information. However, when the confidence value is too large, the information of zero elements may be neglected, and in the extreme case, the node representations may be misled to learn the ability to reconstruct an all-1 matrix.

3) ANALYSIS OF $\gamma$

At last, the influence of concatenating parameter $\gamma$ is explored in this part. By fixing $\alpha = 0.6$ and $\beta = 6$, the variation of the Mic-F1 and Mac-F1 values as a function of $\gamma$ is shown in Figure 4(a) and Figure 4(b) while Figure 4(c) and Figure 4(d) present the influence of $\gamma$ in the clustering task in terms of Purity and NMI. From Figure 4(a) and Figure 4(b), it can be seen that the classification performance can keep in a higher level in terms of both Mac-F1 and Mic-F1 scores when $\gamma$ is in the middle of interval ($0.3 \leq \gamma \leq 0.7$), while somewhat worse results are generated at the edge of interval. As can be observed in Figure 4(c) and Figure 4(d), the performance of clustering may be insensitive to $\gamma$ when it is relatively small ($\gamma \leq 0.7$), but decreases dramatically
as $\gamma$ becomes too large ($\gamma \geq 0.8$). The above phenomenon shows that properly combining structure and attribute information can enhance the performance of node representation. However, the larger $\gamma$ means the less importance of structure information in the node representation, which weakens the utilization of structure information.

Generally, according to the above analysis, we choose a proper parameter combination which are $\alpha = 0.6$, $\beta = 6$ and $\gamma = 0.6$ for all the datasets when conducting the following experiments.

D. COMPARISON RESULTS

In this section, we conduct several experiments to compare the performance of IINE with six state-of-the-art network embedding algorithms.

1) DeepWalk [6]: Inspired by Word2Vec [31], it generates the truncated random walks based on the topological structure and then feeds them into the Skip-gram model to generate the node representations.

2) Node2Vec [7]: It extends DeepWalk by introducing a biased random walk to explore diverse neighborhoods more efficiently which can better preserve the global structure.

3) M-NMF [18]: It preserves both community structure and node local proximity simultaneously when learning the node representations.

4) SDNE [34]: In SDNE, the information of the network topology will be embedded into the deep autoencoder so that the locality, globality and non-linearity of network will be captured in learning node representations.

5) LANE [10]: It jointly adopts the label, attribute, and structure information to generate node representations. And in our experiment, the version without the label information is used.

6) SNE [46]: SNE learns node representations which can preserve the global network structure and the homophily effect by utilizing both of structure and attribute information.

Parameters of these six compared algorithms are set by either the default settings which are suggested by the authors, or tuned by trials to get more proper results. For each algorithm, the dimensionality of node representations is set as 128 for the subnetworks of WebKB, and 256 for other networks. The best result is denoted by bold font.
TABLE 1. Comparison results: Node classification in terms of Mic-F1(%) and Mac-F1(%) on the seven datasets. The last column shows the average rank of each algorithm.

| Dataset & Training ratio | DeepWalk | Node2Vec | M-NMF | SDNE | LANE | SNE | IINE |
|------------------------|----------|----------|-------|------|------|-----|------|
|                        | Mic-F1   | Mac-F1   | Mic-F1| Mac-F1| Mic-F1| Mac-F1| Mic-F1| Mac-F1|
| Citeseer 20%           | 54.49    | 50.55    | 56.00 | 51.54 | 47.58 | 42.42 | 35.06 | 31.45|
|                        | 58.27    | 54.49    | 60.14 | 56.13 | 53.26 | 52.79 | 39.25 | 34.27|
|                        | 61.69    | 56.81    | 61.53 | 58.65 | 46.92 | 47.22 | 41.18 | 35.68|
| 50%                   | 52.29    | 42.30    | 55.80 | 43.14 | 57.23 | 40.48 | 45.73 | 36.52|
|                        | 64.75    | 53.73    | 66.00 | 39.32 | 63.26 | 45.56 | 33.03 | 44.49|
|                        | 67.31    | 57.87    | 66.11 | 40.33 | 64.45 | 51.12 | 56.96 | 48.85|
| 80%                   | 45.89    | 20.35    | 42.71 | 18.73 | 67.63 | 28.90 | 43.77 | 17.04|
|                        | 45.75    | 18.51    | 48.84 | 20.82 | 70.63 | 31.55 | 44.97 | 16.25|
|                        | 48.65    | 21.16    | 49.03 | 21.64 | 76.03 | 34.13 | 53.28 | 27.51|
| Wiki 20%               | 30.76    | 22.21    | 29.49 | 18.57 | 39.10 | 20.73 | 51.92 | 23.51|
|                        | 26.53    | 20.90    | 38.78 | 22.01 | 35.71 | 21.86 | 51.02 | 25.88|
|                        | 25.64    | 12.86    | 38.46 | 20.87 | 25.64 | 13.00 | 51.28 | 26.97|
|                        | 38.67    | 19.11    | 49.33 | 22.07 | 62.67 | 30.05 | 58.00 | 20.45|
|                        | 41.49    | 21.81    | 51.06 | 23.78 | 59.57 | 27.30 | 58.51 | 24.68|
|                        | 42.11    | 19.43    | 57.89 | 24.78 | 65.78 | 37.37 | 57.89 | 15.71|
|                        | 40.21    | 20.19    | 44.02 | 21.40 | 59.78 | 29.78 | 54.34 | 30.17|
| Washington 20%         | 36.52    | 21.78    | 46.95 | 25.10 | 60.87 | 30.17 | 53.04 | 27.94|
|                        | 34.78    | 17.06    | 45.65 | 25.44 | 52.17 | 31.48 | 56.52 | 35.56|
|                        | 40.56    | 25.28    | 40.56 | 23.40 | 46.23 | 23.29 | 44.34 | 21.52|
|                        | 39.09    | 20.66    | 42.11 | 25.89 | 48.12 | 27.35 | 49.62 | 23.89|
|                        | 35.85    | 23.89    | 37.74 | 25.57 | 39.62 | 24.60 | 47.16 | 24.08|
|                        | 5.52     | 5.10     | 4.90  | 4.43  | 3.81  | 3.90  | 4.43  | 4.86 |
| Avg                   | 7.94     | 7.50     | 7.36  | 7.05  | 6.97  | 7.33  | 7.36  | 7.84 |
| Dataset & Training ratio | Mic-F1   | Mac-F1   | Mic-F1| Mac-F1| Mic-F1| Mac-F1| Mic-F1| Mac-F1|
| Citeseer 20%           | 64.00    | 58.85    | 64.00 | 58.85 | 64.00 | 58.85 | 64.00 | 58.85|
|                        | 68.48    | 61.27    | 68.48 | 61.27 | 68.48 | 61.27 | 68.48 | 61.27|
|                        | 67.57    | 64.29    | 67.57 | 64.29 | 67.57 | 64.29 | 67.57 | 64.29|
| Avg                   | 1        | 1        | 1     | 1     | 1     | 1     | 1     | 1     |

The experimental results in Table 1 and Table 2 can to some extent reflect the preference of each algorithm. Among all the compared algorithms, the random walk based models like DeepWalk and Node2Vec are more suitable for the ‘Wiki’ and ‘Citeseer’ datasets, which contains a larger network, while others are more suitable for the relatively smaller networks namely, ‘Terrorist’ and subnetworks in ‘WebKB’. The above phenomenon may be caused by the reason that random walk based methods consider the neighbors in a long path, which may neglect the micro structure of the smaller network, while others mainly adopt the adjacent neighbors as structure information which may be limited to the macro structure of the larger network. Besides, although deep model is adopted in SNE and SDNE, SNE can achieve better results than SDNE on almost all the datasets, which can to some extent reflect the effectiveness of utilizing attribute information. While both attribute information and deep model are used in SNE and IINE, IINE can significantly outperform SNE, which empirically demonstrates that it can effectively enhance the efficiency of node embeddings by keeping the intra-characteristic of each single information when combining them.

E. NETWORK VISUALIZATION

To show the embedding results of different approaches more clearly, t-SNE [47] is utilized to further map the obtained embeddings to 2-d space, where nodes with the same label are marked with the same color. Specifically, we plot the visualization results of the Citeseer dataset in Figure 5.

As shown in Figure 5, the nodes with the same label in the visualized results of IINE are relatively more compact than those by the compared algorithms, which is beneficial for understanding the network structure.
### TABLE 2. Comparison results: Node clustering in terms of Purity(%) and NMI(%) on the seven datasets. The last column shows the average rank of each algorithm.

| Dataset   | DeepWalk | Node2Vec | M-NMF | SDNE | LANE | SNE | IINE |
|-----------|----------|----------|-------|------|------|-----|------|
|           | Purity   | NMI      | Purity | NMI  | Purity | NMI | Purity | NMI  | Purity | NMI  | Purity | NMI  | Purity | NMI  |
| Citeseer  | 41.03    | 12.66    | 48.85  | 20.41| 26.51 | 4.96| 34.30  | 7.58 | 21.50  | 2.96 | 47.01  | 16.58| **61.50** | **29.35** |
| Wiki      | 48.23    | 37.09    | 50.56  | 35.00| 39.10 | 25.89| 39.96  | 24.08| 40.38  | 11.66| 43.70  | 28.26| **51.43** | **38.44** |
| Terrorist | 45.86    | 1.85     | 47.33  | 3.50 | 45.17 | 3.01| 51.35  | 6.23 | 51.04  | 6.58 | 50.50  | 13.32| **76.88** | **25.23** |
| Cornell   | 44.10    | 8.27     | 44.62  | 6.08 | 47.69 | 14.18| 57.95  | 21.08| 46.15  | 5.87 | 48.21  | 10.58| **60.00** | **23.50** |
| Texas     | 55.08    | 4.63     | 55.08  | 4.06 | 57.75 | 14.28| 65.78  | 16.89| 55.61  | 2.26 | 59.89  | 17.27| **67.38** | **27.82** |
| Washington| 53.04    | 7.38     | 51.74  | 5.92 | 63.04 | 20.49| 60.87  | 19.56| 53.04  | 12.54| 61.74  | 21.52| **64.78** | **23.56** |
| Wisconsin | 47.92    | 3.24     | 49.81  | 7.08 | 52.08 | 9.88 | 60.00  | 13.45| 46.42  | 1.70 | 67.92  | 28.87| **68.68** | **34.87** |
| Rank      | 5.28     | 4.86     | 4.86   | 4.86 | 4.86  | 4.29| 3.43   | 3.86 | 5.43   | 6.29 | 3.14   | 2.86| 1      | 1    |

**FIGURE 5.** Visualization of the embedding results of Citeseer generated by the seven algorithms. (a) LANE. (b) SNE. (c) IINE. (d) SDNE. (e) LANE. (f) SNE. (g) IINE.

Besides, several obvious problems can be found in the visualized graph of the compared algorithms. As can be seen from Figure 5(a) and Figure 5(b), both DeepWalk and Node2Vec have a confusing part in their visualized graphs, which may be caused by random walk. In Figure 5(c), the visualized graph of M-NMF shows that, many nodes with different labels are too close together while several node groups with the same label are largely separated, which neglects the discriminative of node representation. As can be seen from Figure 5(d), Figure 5(e), and Figure 5(f), nodes in the visualized graphs of SDNE, LANE, and SNE are somewhat messy, which are likely to violate the rational distribution of the nodes.

**F. CASE STUDY**

In this part, two meaningful problems are investigated as follows:

- Whether the characteristic of each single information matter?
TABLE 3. Analysis of the influence of intra-characteristic of single information: Classification performance of IINE and IINE\(_w\) in terms of Mic-F1 and Mac-F1(\%).

| Dataset | Wiki | Citeeseer | Terrorist | Cornell | Texas | Washington | Wisconsin |
|---------|------|-----------|-----------|---------|-------|------------|------------|
| IINE\(_w\) | Mic-F1 | 66.81 | 72.97 | 86.59 | 79.48 | 68.42 | **82.61** | 77.35 |
| | Mac-F1 | 62.21 | 62.85 | 44.27 | 74.00 | 39.50 | 58.04 | 53.92 |
| IINE | Mic-F1 | **67.57** | 74.64 | 88.03 | 82.05 | 81.58 | 82.61 | **81.13** |
| | Mac-F1 | 64.29 | 69.03 | 44.91 | 81.54 | 56.57 | 74.73 | 63.49 |

TABLE 4. Analysis of the influence of intra-characteristic of single information: Clustering performance of IINE and IINE\(_w\) in terms of Purity and NMI(\%).

| Dataset | Wiki | Citeeseer | Terrorist | Cornell | Texas | Washington | Wisconsin |
|---------|------|-----------|-----------|---------|-------|------------|------------|
| IINE\(_w\) | Purity | 60.14 | 47.48 | 72.11 | 46.67 | 57.22 | 63.48 | 61.13 |
| | NMI | 28.13 | 37.15 | 21.10 | 7.67 | 8.95 | 21.97 | 21.52 |
| IINE | Purity | **61.50** | 51.43 | 76.88 | 60.00 | **67.38** | **64.78** | **68.68** |
| | NMI | 29.35 | 38.44 | 25.23 | 23.50 | 27.82 | 23.56 | 34.87 |

TABLE 5. Analysis of the influence of consistency: Classification performance of IINE and IINE\(_c\) in terms of Mic-F1 and Mac-F1(\%).

| Dataset | Citeeseer | Wiki | Terrorist | Cornell | Texas | Washington | Wisconsin |
|---------|-----------|------|-----------|---------|-------|------------|------------|
| IINE\(_c\) | Mic-F1 | 22.17 | 45.11 | 40.93 | 44.06 | 57.89 | 50.00 | 47.50 |
| | Mac-F1 | 9.74 | 35.21 | 10.19 | 11.11 | 14.67 | 16.67 | 11.35 |
| IINE | Mic-F1 | **68.62** | **74.64** | 88.03 | **82.05** | **81.58** | **82.61** | **81.13** |
| | Mac-F1 | 64.29 | 69.03 | 44.91 | 81.54 | 56.57 | 74.73 | 63.49 |

TABLE 6. Analysis of the influence of consistency: Clustering performance of IINE and IINE\(_c\) in terms of Purity and NMI(\%).

| Dataset | Wiki | Citeeseer | Terrorist | Cornell | Texas | Washington | Wisconsin |
|---------|------|-----------|-----------|---------|-------|------------|------------|
| IINE\(_c\) | Purity | 23.07 | 19.71 | 50.58 | 45.13 | 57.75 | 47.83 | 51.32 |
| | NMI | 0.40 | 4.65 | 3.30 | 6.93 | 7.50 | 3.48 | 11.22 |
| IINE | Purity | **61.50** | 51.43 | 76.88 | 60.00 | 67.38 | **64.78** | **68.68** |
| | NMI | 29.35 | 38.44 | 25.23 | 23.50 | 27.82 | 23.56 | 34.87 |

- Whether the consistency criterion will cause the information distortion in attributed network embedding?

In order to obtain the answer of the above problems, two case study experiments are conducted in both classification (with training ratio as 80%) and clustering tasks.

1) THE INFLUENCE OF INTRA-CHARACTERISTIC OF SINGLE INFORMATION

In our model, the intra-characteristic of single information is captured respectively by the structure miner and the attribute miner. To investigate the influence of intra-characteristic, the performance of the original IINE and that of the IINE without auxiliary models (denoted as IINE\(_w\)) are compared in both of the classification and clustering tasks, and the results are reported in Table 3 and Table 4.

From Table 3 and Table 4, we can find that on all the datasets, IINE can significantly outperform IINE\(_w\) in terms of both classification and clustering tasks. In the classification task, the improvements in terms of Mic-F1 and Mac-F1 are 4.38% and 15.14% respectively; While in the clustering task, the improvements in terms of Purity and NMI are 10.39% and 38.42% respectively. As analyzed above, in the clustering task, the performance of learned representation will degenerate sharply when there is no measure to preserve the intra-characteristic of each single information. This is mainly due to the reason that the unsupervised node clustering task requires the intra-characteristic information for making nodes more informative and distinctive. This case study experiment empirically demonstrates that the performance of node representations can be significantly enhanced when the intra-characteristic of each information is properly preserved.

2) THE INFLUENCE OF CONSISTENCY CRITERION

The consistency criterion believes that the structure representation and the attribute representation of the same node are supposed to be similar. Therefore, adding the consistency criterion is equivalent to adding similarity constraint on \(U_A\) and \(U_P\), which can be formulated as follows:

\[
L_c = \|U_A - U_P\|_F
\]  
(23)
After adding the above loss, the consistency criterion based model can be derived, which is denoted as IINE\textsubscript{s}. The comparison results of IINE and IINE\textsubscript{s} are shown in Table 5 and Table 6. As can be seen from both Table 5 and Table 6, the results of both classification and clustering of IINE\textsubscript{s} degenerate dramatically. As can be observed in the negative influence on the experimental performance, it is obvious that the consistency criterion is likely to distort the learned information of node representations, leading to the degradation of performance. Therefore, this study to some extent demonstrates the irrationality of consistency criterion on combining the different information.

VI. CONCLUSIONS

In this paper, we study a novel problem of attributed network embedding, namely how to learn node representations that can reflect the inter-relation between structure and attribute information while avoiding the information distortion and preserving the corresponding intra-characteristic of each single information. To solve this problem, we propose the IINE model. The basic idea is to design a coupled autoencoder to learn the inter-relation inside the deeply coupled information from both of the network structure and the node attributes. In addition, two auxiliary models called structure miner and attribute miner are designed to learn single information representations so as to further mine the intra-characteristic inside them. By jointly correlating the coupled information representations and single information representations, an overall objective function is proposed. Extensive experiments conducted on the seven real-world attributed networks show that our model outperforms most of the state-of-the-art network embedding methods in both of the node classification and clustering tasks. Additionally, two case study experiments are conducted by comparing IINE with two variations models namely IINE\textsubscript{w} and IINE\textsubscript{s}, which empirically demonstrates the significance of the novel problem proposed in this paper.

REFERENCES

[1] D. Benyamin, M. C. McGinley, M. A. Hall, and N. J. Bina, “Social advertisement network,” U.S. Patent 12,467,981, Dec. 10, 2009.
[2] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu, “Learning deep representations for graph clustering,” in Proc. 28th AAAI Conf. Artif. Intell., 2014, pp. 1293–1299.
[3] C. Li, Z. Li, S. Wang, Y. Yang, X. Zhang, and J. Zhou, “Semi-supervised network embedding,” in Proc. International Conference on Database Systems for Advanced Applications (DSAA’16), Suzhou, China, 2017, pp. 131–147.
[4] M. Bhuiyan and M. Al Hasan, “Representing graphs as bag of vertices and partitions for graph classification,” Data Sci. Eng., vol. 3, no. 2, pp. 150–165, 2018.
[5] X. Zhang, Z. Zhao, C. Li, Y. Zhang, and J. Zhao, “An interpretable and scalable recommendation method based on network embedding,” IEEE Access, vol. 7, pp. 9384–9394, 2019.
[6] B. Perozzi, R. Al-Rfou, and S. Skiena, “DeepWalk: Online learning of social representations,” in Proc. KDD, 2014, pp. 701–710.
[7] A. Grover and J. Leskovec, “Node2vec: Scalable feature learning for networks,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, San Francisco, CA, USA, Aug. 2016, pp. 855–864.
[8] 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 (WWW), Florence, Italy, May 2015, pp. 1067–1077.
[9] D. Yang, S. Wang, C. Li, X. Zhang, and Z. Li, “From properties to links: Deep network embedding on incomplete graphs,” in Proc. ACM Conf. Inf. Knowl. Manage. (CIKM), Singapore, Nov. 2017, pp. 367–376.
[10] X. Huang, J. Li, and X. Hu, “Label informed attributed network embedding,” in Proc. 10th ACM Int. Conf. Web Search Data Mining (WSDM), Cambridge, U.K., Feb. 2017, pp. 731–739.
[11] T. Li, J. Zhang, P. S. Yu, Y. Zhang, and Y. Yan, “Deep dynamic network embedding for link prediction,” IEEE Access, vol. 6, pp. 29219–29230, 2018.
[12] 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.
[13] X. Shen and F.-L. Chang, “Deep network embedding for graph representation learning in signed networks,” IEEE Trans. Cybern., to be published.
[14] Y. Shi, H. Gui, Q. Zhu, L. Kaplan, and J. Han, “AspEm: Embedding learning by aspects in heterogeneous information networks,” in Proc. SIAM Int. Conf. Data Mining, San Diego, CA, USA, 2018, pp. 144–152.
[15] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, “Asymmetric transitivity preserving graph embedding,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, San Francisco, CA, USA, Aug. 2016, pp. 1105–1114.
[16] C. Tu, Z. Zhang, Z. Liu, and M. Sun, “TransNet: Translation-based network representation learning for social relation extraction,” in Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI), Melbourne, VIC, Australia, Aug. 2017, pp. 2864–2870.
[17] S. Cao, W. Lu, and Q. Xu, “GraRep: Learning graph representations with global structural information,” in Proc. 24th ACM Int. Conf. Inf. Knowl. Manage. (CIKM), Melbourne, VIC, Australia, Oct. 2015, pp. 891–900.
[18] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, “Community preserving network embedding,” in Proc. 31st AAAI Conf. Artif. Intell., San Francisco, CA, USA, Feb. 2017, pp. 203–209.
[19] Y. Zhang, T. Lyu, and Y. Zhang, “COSINE: Community-preserving social network embedding from information diffusion cascades,” in Proc. 32nd AAAI Conf. Artif. Intell., 2018.
[20] L. Du, Z. Lu, Y. Wang, G. Song, Y. Wang, and W. Chen, “Galaxy network embedding: A hierarchical community structure preserving approach,” in Proc. 27th Int. Joint Conf. Artif. Intell., 2018, pp. 2079–2085.
[21] J. D. Burger, J. Henderson, G. Kim, and G. Zaretta, “Discriminating gender on Twitter,” in Proc. Conf. Empirical Methods Natural Lang. Process. Association for Computational Linguistics: Stroudsburg, PA, USA, 2011, pp. 1301–1309.
[22] M. Pennacchiotti and A.-M. Popescu, “Democrats, republicans and starbucks aficionados: User classification in Twitter,” in Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, San Francisco, CA, USA, Aug. 2011, pp. 1301–1309.
[23] C. Li, S. Wang, D. Yang, Z. Li, Y. Yang, X. Zhang, and J. Zhou, “PPNE: Property preserving network embedding,” in Proc. 22nd Int. Conf. Database Syst. Adv. Appl. (DASFAA), 2017, pp. 163–179.
[24] C. Tu, H. Liu, Z. Liu, and M. Sun, “CANE: Context-aware network embedding for relation modeling,” in Proc. 55th Annu. MeetIng Assoc. Comput. Linguistics (ACL), Vancouver, BC, Canada, vol. 1, Jul./Aug. 2017, pp. 1722–1731.
[25] 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. (IJCAI), Buenos Aires, Argentina, Jul. 2015, pp. 2111–2117.
[26] R. Hong, Y. He, L. Wu, Y. Ge, and X. Wu, “Deep attributed network embedding by preserving structure and attribute information,” IEEE Trans. Syst., Man, Cybern. Syst., to be published.
[27] J. Lin, L. Zhang, M. He, H. Zhang, G. Liu, X. Chen, and Z. Chen, “Multi-path relationship preserved social network embedding,” IEEE Access, vol. 7, pp. 26507–26518, 2019.
[28] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” Science, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.
[29] J. B. Tenenbaum, V. de Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” Science, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.
[30] M. Belkin and P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” in Proc. Adv. Neural Inf. Process. Syst., Vancouver, BC, Canada, Dec. 2001, pp. 585–591.
[31] 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., Lake Tahoe, NV, USA, Dec. 2013, pp. 3111–3119.

[32] O. Levy and Y. Goldberg, “Neural word embedding as implicit matrix factorization,” in Proc. Adv. Neural Inf. Process. Syst., Montreal, QC, Canada, Dec. 2014, pp. 2177–2185.

[33] S. Cao, W. Lu, and Q. Xu, “Deep neural networks for learning graph representations,” in Proc. 13th AAAI Conf. Artif. Intell., Phoenix, AZ, USA, Feb. 2016, pp. 1145–1152.

[34] D. Wang, P. Cui, and W. Zha, “Structural deep network embedding,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining, San Francisco, CA, USA, Aug. 2016, pp. 1225–1234.

[35] L. Liu, Y. Zhang, S. Fu, F. Zhong, J. Hu, and P. Zhang, “ABNE: An attention-based network embedding for user alignment across social networks,” IEEE Access, vol. 7, pp. 23595–23605, 2019.

[36] Q. Dai, Q. Li, J. Tang, and D. Wang, “Adversarial network embedding,” in Proc. 32nd AAAI Conf. Artif. Intell. (AAAI), 30th Innloc. Appl. Artif. Intell. (IAAI), and 8th AAI Symp. Educ. Adv. Artif. Intell. (EAAI), New Orleans, LA, USA, Feb. 2018, pp. 2167–2174.

[37] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, “Generative adversarial networks,” CoRR, 2014.

[38] H. Gao and H. Huang, “Deep attributed network embedding,” in Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI), Stockholm, Sweden, Jul. 2018, pp. 3364–3370.

[39] J.-H. Li, C.-D. Wang, L. Huang, D. Huang, J.-H. Lai, and P. Chen, “Attributed network embedding with micro-meso structure,” in Proc. Int. Conf. Database Syst. Adv. Appl., Gold Coast, QLD, Australia, May 2018, pp. 20–36.

[40] M. Ögretir and A. T. Cemgil, “Comparison of collaborative deep learning and nonnegative matrix factorization for recommender systems,” in Proc. 25th Signal Process. Commun. Appl. Conf. (SIU), Antalya, Turkey, May 2017, pp. 1–4.

[41] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980. [Online]. Available: https://arxiv.org/abs/1412.6980

[42] L. Liu, L. Xu, Z. Wang, and E. Chen, “Community detection based on structure and content: A content propagation perspective,” in Proc. IEEE Int. Conf. Data Mining (ICDM), Atlantic City, NJ, USA, Nov. 2015, pp. 271–280.

[43] J. Liu, Z. He, L. Wei, and Y. Huang, “Content to node: Self-translation network embedding,” in Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), London, U.K., Aug. 2018, pp. 1794–1802.

[44] B. Zhao, P. Sen, and L. Getoor, “Event classification and relationship labeling in affiliation networks,” in Proc. ICML, 2006, pp. 271–280.

[45] A. Strehl and J. Ghosh, “Cluster ensembles—A knowledge reuse framework for combining multiple partitions,” J. Mach. Learn. Res., vol. 3, pp. 583–617, Dec. 2002.

[46] L. Zhao, 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.

[47] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol. 9, pp. 2579–2605, Nov. 2008.

[48] K. Wang received the bachelor’s degree from Sun Yat-sen University, in 2018. He is a Graduate Student with Sun Yat-sen University, from September 2018. His current research interest includes data mining.

[49] LING HUANG received the undergraduate and master’s degree from South China University of Technology, in 2009 and 2013, respectively. She is currently pursuing the Ph.D. degree with Sun Yat-sen University. She has published several papers in international journals and conferences such as pattern recognition, IEEE ACCESS, information sciences, knowledge-based systems, KDD, AAAI, IJCAI, IEEE ICDM, IEEE BBIM, and DASFAA. Her research interest includes data mining.

[50] YONG TANG received the B.S. degree from Wuhan University, in 1985, and the Ph.D. degree from the University of Science and Technology of China, in 2001, all in computer science. He is currently a Professor and the Dean of the School of Computer Science at South China Normal University, and serves also as the Director of Services Computing Engineering Research Center of Guangdong Province. His research focuses on data base and cooperative software, temporal information processing, social network, and big data analytics. He has completed more than 30 research and development projects, and has authored or co-authored more than 100 publications in these areas.

[51] CHENGZHOU FU received the master’s degree in software engineering, in 2012 and the Ph.D. degree in computer science, in 2017, from South China Normal University, China. His current research interests include social network and data mining.

[52] KAI WANG received the bachelor’s degree from Sun Yat-sen University, in 2018. He is a Graduate Student with Sun Yat-sen University, from September 2018. His current research interest includes data mining.

[53] LEI XU received the bachelor’s degree from Sun Yat-sen University, in 2018. He is a Graduate Student with Sun Yat-sen University, from September 2018. His current research interest includes data mining.