Neurocomputing 456 (2021) 147–155

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Unsupervised learning for community detection in attributed networks based on graph convolutional network

Xiaofeng Wang [a,b,*], Jianhua Li [b,*], Li Yang [a], Hongmei Mi [b]

[a] College of Information Engineering, China Jiliang University, Hangzhou 310018, China
[b] School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 30 December 2020
Revised 6 April 2021
Accepted 17 May 2021
Available online 20 May 2021
Communicated by Zidong Wang

Keywords:
Community detection
Network representation learning
Attributed networks
Graph convolutional network
Label sampling

ABSTRACT

Community detection has emerged during the last decade as one of the most challenging problems in network science, which has been revisited with network representation learning recently and has attracted considerable attention. Many approaches have been proposed in recent years, including the latest methods based on graph convolutional network (GCN). Here, we propose a new network representation learning method based on GCN for community detection in attributed networks without prior label information. Inspired by the message pass mechanism of GCN and the local self-organizing property of community structure, we integrate a label sampling model and GCN into an unsupervised learning framework to uncover underlying community structures by fusing topology and attribute information. The label sampling model constructs a balanced training set by structural center location and neighbor node expansion to train the GCN. The experiments on various real-world networks give a comparison view to evaluate the proposed method. The experimental results demonstrate the proposed method performs more efficiently with a comparative performance over current state-of-the-art community detection algorithms.

1. Introduction

Networks have become a general form for describing and modeling complex systems in recent years, as most natural and engineered systems in which a set of entities interact with each other can be abstracted as complex networks [1]. Network theory is an important tool for gaining deep insights into complex systems throughout the physical, biological, social, and information sciences [2]. Community structure or cluster is an important structural characteristic of complex networks, as it often associates with a functional and organizational module of underlying systems [3,4]. A network community generally refers to a sub-graph where nodes are connected more tightly than that in different communities, such as social groups in online social networks. Uncovering community structures by dividing unlabeled nodes into different communities based on network topologies in networks is one of the most important tasks of network analysis. Community detection is helpful to reveal significant hidden properties of complex networks from a mesoscopic perspective, e.g., the organizational principles of an abnormal group, and the functional characteristics of a structural unit. How to uncover communities has drawn increasing attention in recent years.

A large number of community detection methods have been proposed from various perspectives over the past decade [5,6]. Most of the existing methods attempt to partition a network into disjoint or overlapping communities based on network topology. These conventional methods mainly include graph partitioning methods, clustering-based methods, modularity optimization methods, model-based methods, and so on. However, with the widespread use of information technologies, networks are becoming increasingly popular to capture more complex relationships, which brings challenges to these traditional methods. In reality, the large scale of complex networks often makes these methods computationally expensive or intractable. For example, many existing methods involving the spectral decomposition of a matrix [7] require at least quadratic time complexity concerning the number of vertices, which makes it hard to scale to large-scale networks. More importantly, except topological properties, there are large amounts of available attribute information on nodes or edges in networks, such as profiles, contents, and semantics, which actually help to uncover more meaningful communities but have been underutilized in community detection.

In recent years, network presentation learning (NRL) has attracted great attention and is expected to tackle these challenges

* Corresponding authors.
E-mail addresses: xfwang@cjlu.edu.cn (X. Wang), lijh888@sjtu.edu.cn (J. Li), lyang@cjlu.edu.cn (L. Yang), hongmei.mi@cjlu.edu.cn (H. Mi).

https://doi.org/10.1016/j.neucom.2021.05.058
0925-2312/

[8,9]. NRL aims to learn latent, low dimensional vectorized representations of network vertices while preserving structure proximity and attribute affinity [8]. The resulting vectorized representations can be then taken as input features of various vector-based machine learning tasks, such as node classification, link prediction, and visualization. Current significant efforts have been made to develop scalable and effective NRL techniques that are directly designed for complex networks [10]. Many NRL methods have been proposed to learn node representations, such as random walks based methods (e.g., DeepWalk [11], LINE [12] and struc2vec [13]), Matrix Factorization based methods (e.g., M-NMF [14] and TADW [15]), and deep learning-based methods (e.g., DNGR [16] and SDNE [17]). Although NRL methods usually use the node clustering performance to evaluate the validity of the learned representations on various networks, there is little work on NRL for community detection as an unsupervised learning task.

We would like to focus on the unsupervised learning-based NRL for community detection due to its nonlinear representation ability in fusing topology and attribute information. The existing work including methods based on graph autoencoders (GAE) [18,19] and graph neural networks (GNN) [20,21], followed the common approach that first uses an NRL algorithm to learn node representations, and then performs a clustering algorithm (e.g., k-means) to obtain clusters on the learned representations. However, these deep learning-based methods can be challenging to implement because there are many hyper-parameters to be adjusted, especially the number of clusters. For example, Wang et al. [19] tried to learn deep representations with a GAE and run the spectral clustering algorithm on resulting representations for graph clustering. Similarly, Yang et al. [22] proposed to reconstruct the modularity matrix based on stacked GAEs. Almost both of them are challenging to implement in large networks due to high computational cost and many parameters to be fined. Actually, the cluster number is generally unknown in advance in most real-world networks, especially when a network is large and heterogeneous.

More recently, a graph convolutional network (GCN) is introduced to address the problem of community detection on graphs [23,24]. GCN is effective to extract complex features from network topologies and node attributes by a stack of convolution operations like CNN [24]. Jin et al. [25] proposed to solve semi-supervised community detection by integrating GCN and the statistical modeling of Markov Random Fields. Sun et al. [26] developed a network embedding framework with graph convolutional autoencoder to learn the representations of nodes for clustering task. Besides, Jin et al. [27] also proposed an unsupervised model for community detection via joint GCN. In addition, Luo and Du used GCN based autoencoder to detect communities as well as structural hole spanners [28]. GCN is effective to integrate network topology and node attributes in community detection, however, there are still some challenges to this unsupervised task. Firstly, GCN learns representations of hidden layers by encoding local topological features and attributes of nodes but does not consider community properties. Moreover, GCN generally relies on prior knowledge such as a large number of labels which often are not available to achieve appropriate network structure and parameter settings.

To take advantage of graph convolutional network in localized feature extraction, we propose a new unsupervised learning framework based on GCN for community detection in attributed networks, which can tackle these drawbacks of the original model. The main contributions of our work are as follows:

(1) Without any prior knowledge about community structures, we propose a label sampling method by locating structural centers to sample a small number of nodes as a balanced label set, which makes subsequent training more effective to identify communities. We illustrate the effectiveness of the label sampling model by a specific example.
(2) In order to uncover underlying communities in attributed networks, we integrate the label sampling model into the design of an unsupervised learning framework based on the graph convolutional network. Both network topology and node attitudes are considered during model learning to obtain consistent community structures.
(3) Through experiments on various real-world networks, we present that the proposed method outperforms several baseline methods in the community detection task. We give a comparison view to evaluate the performance of the proposed method with four types of baseline methods.

The rest of this paper is organized as follows. In Section 2, we introduce related work and the background of the proposed method. Some preliminary notations and formulations related to this work are given in Section 3. Section 4 gives a detailed description of the proposed method for community detection in attributed networks. We present experimental results on various real-world networks to validate the performance of the proposed method in Section 5, we also compare it with other state-of-the-art algorithms. Finally, the conclusions are summarized in Section 6.

2. Related work

As mentioned above, our work was inspired by the recent work on network representation learning for community detection. In this section, the related work about community detection and network representation learning is introduced in detail.

2.1. Community detection

Community detection in complex networks is one of the most important topics of modern network science, which aims at uncovering underlying groups or clusters of vertices in a network. Communities generally show dense connectivity within communities and sparse connections between communities in topology, therefore the task of identifying community is also treated as unsupervised clustering in networks. Most real-world networks display community structures where nodes play similar roles or share common properties.

Early work mainly focused on the characteristics of network topological, and a large number of community detection methods had been proposed based on various metrics of similarity between nodes. Graph partitioning methods, such as min–max cut and normalized cut [5], sought to recursively divide the nodes into a predefined number of clusters that minimize inter-cluster edges. Hierarchical clustering methods divided a graph or aggregate nodes based on similarity measures between nodes [6]. Modularity-based optimization methods converted the task of community identification into a maximization problem of a modularity function of a community structure [29]. Fuzzy clustering-based methods aimed to relax the community membership of a node and uncover multi-resolution community structures [30,31]. Meanwhile, many local methods were proposed to detect disjoint or overlapping communities more flexibly [5]. For example, label propagation-based methods made use of the local spread process of node labels according to dynamic interaction rules to identify communities towards linear time [32]. Some local methods attempted to uncover communities from selected seeds by adding or removing nodes to optimize a local measure, which usually involve different label sampling strategies, such as random

node selection, PageRank sampling, random walk, and centrality sampling [29,33,34]. For example, centrality sampling algorithms identify important nodes based on various centrality measures like degree centrality, closeness centrality and so on [35]. However, these methods only use the network topology to identify community structures via greedy optimization.

Meanwhile, increasing attribute information except for network topology becomes available and is considered in community detection to determine more meaningful communities. The additional information of a network such as node attributes, link content, and semantic information always carries valuable characteristics of communities and can also effectively alleviate the structural sparsity problem in network analysis. Many methods have been proposed to address this issue [36,37]. For example, Wang et al. [38] proposed a nonnegative matrix factorization model by integrating network topology and node attributes to uncover semantic communities. He et al. [39] introduced a generative model by combing a nested EM algorithm and belief propagation to jointly identify communities and semantic. A unified weakly-supervised framework was developed to capture the interaction between community structures and node attributes [40]. It's noteworthy that most of these methods involve matrix decomposition operations with high complexity and are hard to scale to large-scale attributed networks.

2.2. Network representation learning

Network representation learning as a new learning paradigm is typically proposed to learn latent and low dimensional representations of vertices by preserving structural features and node attributes, which has been proved to be effective in network analysis in recent years [8]. Earlier work related to NRL, such as Local Linear Embedding (LLE) and Laplacian Eigenmap (LE) [9], is mainly for dimensionality reduction and suffers from scalability issue in large-scale networks. Current NRL methods embed network nodes into a new representation space which facilitates the original complex network to be easily and efficiently handled for further analysis. Many representative algorithms have been developed to learn informative representations, and are scalable for various applications with promising performance.

More specifically, DeepWalk [11] as the pioneering work made use of random walks to capture the structural relationship between nodes and then employed a language model to learn node representations. Some improved versions including struct2vec [13], node2vec [41] and SNS [42] had been proposed. To leverage node attributes, some extended algorithms based on the random walk version were developed to harness representation learning, such as DDRW [43] and PPNE [44]. Besides, matrix factorization was used to embed high-dimensional node representation, algorithms like M-NMF [14] and GrapWave [45] performed eigen decompositions to learn indicative representations. TADW [15] imported node textual features in the inductive matrix factorization. Moreover, deep learning techniques were applied to NRL in order to capture highly nonlinear characteristics of complex networks [10]. Some algorithms like DNGR [16], SDNE [17], and ANRL [46] employed a deep autoencoder model on the high-dimensional matrix representations to extract non-linearity. Other end-to-end methods based on neural network, e.g., SNE [47] and DeepGL [48] incorporated network structure and attribute information in the inductive graph representation learning to get sufficient knowledge interactions.

There have also been efforts to develop semi-supervised methods for community detection based on network representation learning, where label information is combined with graph-based regularization to infer unlabeled nodes [8,9]. Instead of imposing regularization, Yang et al. [49] utilized node representations to jointly predict the context in a network and leveraged node labels to build both transductive and inductive variants. Recently, graph convolutional network (GCN) is introduced in network analysis tasks [50,51]. Different from most semi-supervised methods that focus on preserving network structure, GCN-based methods help to corporate network topology and attribute information, but rely on a large number of node labels to identify unlabeled nodes. Jin et al. [25] proposed to integrate GCN and Markov random field model to uncover communities. A network embedding framework based on graph convolutional autoencoder was been proposed by Sun et al. [26] for node clustering. Moreover, Some unsupervised approaches have been proposed recently. Jin et al. [27] proposed an unsupervised model for community detection via GCN embedding. He et al. [52] developed a community-centric GCN model for unsupervised community detection. In this work, we proposed a new unsupervised learning framework for community detection by integrating the local label sampling model and the GCN, which combines the advantage of GCN and local methods to make community discovery more effective.

3. Preliminaries

This section briefly introduces the preliminary knowledge of this work, including basic notations, problem statements, and the architecture of graph convolutional networks.

3.1. Notations and the problem

We are interested in the community detection task in an attributed network in which given an input network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ over a node set $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ with $|\mathcal{V}| = N$, an edge set $\mathcal{E}$ with $e_{ij} = (v_i, v_j) \in \mathcal{E}$, and a set of node attributes $\mathcal{W} = \{w_1, w_2, \ldots, w_M\}$ with M dimensions, a labeling $l : \mathcal{V} \to \{1, \cdots, C\}$ that denotes a partition P of all nodes into C communities is to be predicted for the network. For simple, we consider an undirected and attributed network $\mathcal{G}$ specified by an adjacent matrix $A = (a_{ij})_{N\times N}$ ($a_{ij} = 1$ if $e_{ij} \in \mathcal{E}$, or 0 otherwise) which encodes structural connectivity over all nodes, and an attribute matrix $X \in \mathbb{R}^{N\times M}$ which includes integrated attributes for all nodes. The resulting community structures in the attributed network generally show obvious local clustering characteristics in topology and dissimilarity in attributes.

If there are only a small number of labels that indicate some but not all of the nodes belong to some groups, these labels can be indexed by a set of k labels $\{l_1, l_2, \cdots, l_k\}$. The problem of community detection is then treated as a semi-supervised learning task where unlabeled nodes are assigned to correspond communities by training on partially labeled nodes. However, there is still a more extreme situation that no label information can be available in some complex networks. This means its more challenging to uncover underlying communities in these unlabeled networks. Here, we are interested in developing community detection methods that don't depend on the known label information. Therefore, we require that the method accepts networks of no labeled node, similar to unsupervised NRL models. Inspired by the message pass mechanism of GCN and the local self-organizing property of community structure, we integrate a label sampling model and GCN into an unsupervised learning framework to uncover community structures by fusing topology and attribute information. The label sampling model constructs a balanced training set by structural center location and neighbor node expansion without prior label information.

3.2. Convolutional operation on Graphs

As a type of graph-based neural networks, graph convolutional neural network (GCN) was first introduced to handle undirected graphs by a local operator with spectral Laplacians eigenbasis [24]. We first briefly review the generic GCN architecture and then describe some variants. Based on spectral graph theory, an input signal $\mathbf{x} \in \mathbb{R}^N$ on the nodes of a graph can be transformed into the spectral domain by graph Fourier transformation (GFT)$U^T\mathbf{x}$, where U is the matrix of eigenvectors of the normalized graph Laplacian L defined as $L = U\Lambda U^T$ (Λ is the diagonal matrix of eigenvalues). The transformed graph signal can be convolved with a function of the diagonal matrix $g_\theta(\Lambda)$ parameterized by $\theta \in \mathbb{R}^N$ in Fourier domain. The spectral convolutional operation of GCN can then be formulated as:

$$g_\theta * \mathbf{x} = U g_\theta(\Lambda) U^T \mathbf{x}, \tag{1}$$

where $\mathbf{s}$ is the vector representation of node attributes in the graph. In Eq. (1), $U^T\mathbf{x}$ and $U\mathbf{x}$ represent GFT and inverse transformation (IGFT) of $\mathbf{x}$, respectively. The function $g_\theta(\Lambda)$ of the eigenvalues is used as a filter for graph convolution, but it is computationally expensive for large graphs to the eigenvalue decompositions of L which requires as $\mathcal{O}\left(N^2\right)$ time. To tackle this problem, it was suggested to approximate the filter function $g_\theta(\Lambda)$ by K -order Chebyshev polynomial expression $T_K\left(\tilde{\Lambda}\right)$ where $\tilde{\Lambda} = (2/\lambda_{max}) - I_N$ and λ_{max}) represents the largest eigenvalue of L, i.e., $g_\theta\left(\tilde{\Lambda}\right) \approx \sum_{k=0}^{K} \theta'_k T_K\left(\tilde{\Lambda}\right)$ where $\theta' \in \mathbb{R}^K$ is the vector of Chebyshev coefficients with K orders. Therefore, the definition of the convolutional operator can be reformulated as:

$$g_{\theta'} * \mathbf{x} \approx \sum_{k=0}^{K} \theta'_k T_k\left(\widetilde{L}\right)\mathbf{x}, \tag{2}$$

where $\widetilde{L} = (2/\lambda_{max})L - I_N$. This expression shows graph convolution is a localized filter that depends on the K-order neighborhood of the central node. Note that the computational complexity of Eq. (2) is linear in the number of edges, so Defferrard et al. [50] defined a GCN architecture with K-localized convolution. Moreover, Kipf and Welling [51] further simplified the graph convolution model by setting $k = 1$ and $\lambda_{max} = 2$, and proposed a fast approximate expression:

$$g_\theta * \mathbf{x} \approx \theta \widetilde{D}^{-1/2} \widetilde{A} \widetilde{D}^{-1/2} \mathbf{x}, \tag{3}$$

where θ is the only Chebyshev coefficient left, and $\widetilde{D}$ denotes the renormalized degree matrix ($\widetilde{D}_{ij} = \sum_j \widetilde{A}_{ij}$) with $\widetilde{A} = A + I_N$. This definition can be generalized to a graph signal $\mathcal{G}$ with an attribute matrix X and be taken as a layer-wise convolution operation. A GCN model based on spectral convolutions can therefore be constructed by stacking multiple convolutional layers.

4. Methodology

4.1. Overview

Graph convolutional network naturally integrates the topological structures and node attributes of a graph in the convolution, which helps make the semi-supervised learning problem on complex networks easier. However, the graph convolutional operation is essentially a localized filter that requires a considerable number of labeled nodes for validation and model selection [53]. Therefore, it is unfeasible to apply the GCN model in networks with no or few labels. Additionally, we find that the number of labeled nodes required in GCN depends on the distribution of labels, that is, if most of the trainable nodes are topologically close to each other and far from the centers of clusters, which lead to GCN cannot effectively propagate the label information to the global network and require more training labels.

To address this problem, we propose to train a GCN with a label sampling model as the latter can construct a training set of planted labels, which complements the GCN model. Specifically, we first employ a node selection model to locate the structural centers which are distributed in various clusters and label them as the initial train set. Then, the nearest neighbors to the labeled structural centers are added to the train set to train the GCN model. Different from the original GCN model, we directly optimize the GCN model on the balanced training set and don't need additional data for validation. Moreover, we consider the connectivity of graphs, which is consistent with the nature of community structures. By doing so, we can develop a GCN-based method for community detection. Our method can be divided into two phases as summarized in Algorithm 1.

Algorithm 1: Framework of the proposed model

Require: A representation of a network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ specified by an adjacent matrix $A \in \mathbb{R}^{N\times N}$ and an attribute matrix $X \in \mathbb{R}^{N\times M}$;
Ensure: Community partition P;
1: //Phase 1: Label sampling
2: Consider the connectivity of the network based on the adjacent matrix A, and locate the structural centers with Algorithm 2;
3: Assign and construct a label set y_t with structural centers and their nearest neighbors;
4: //Phase 2: Train GCN model
5: Train the GCN model based on the sampled labels y_t which evenly distribute in different clusters, to predict unlabeled nodes for community detection;
6: **return** Community partition P with a labeling l.

4.2. Local label sampling

As mentioned above, a GCN model with multi-layers cannot effectively propagate the label information to the global network and require more training labels. This is mainly due to the uneven distribution of labeled nodes in the network. We propose to construct a trainable label set by a label sampling model, which can directly optimize the GCN based on a small number of labels without additional labeled nodes for validation. In particular, we first employ a structural center location method to find the structural centers which distribute in different communities and then add them to the train set as initially labeled nodes. Moreover, we sample the nearest neighbors to the train set to further ensure the GCN model converges as soon as possible with enough labels.

We propose to use the structural center location method as our label sampling model to select distinguishable labels. Different from other sampling algorithms, this label sampling model is based on a structural centrality that measures both local density of nodes and relative distance between nodes. It can find structural centers that are characterized by a higher density than their neighbors and by a relatively large distance from nodes with higher densities. Therefore, structural centers generally have high centrality and distribute in different groups, which implies that the number of communities arises intuitively. This model can actually be used to identify a local community in networks [54]. The identified structural centers determine the number of clusters and can be

used as the initial labels in our model. Additionally, the structural centers are closely connected to other core nodes as well as peripheral nodes, which greatly improve the efficiency of label propagation in communities.

The process of constructing the training set is described in Algorithm 2. First, we construct a distance matrix $D = (d_{ij})_{N\times N}$ between nodes where $d_{ij} = d(i,j)$ represents the shortest path length between node i and node j. The distance matrix can also be replaced by other matrix representations of graphs, like the adjacency matrix, similarity matrix. Second, we calculate the structural centrality of nodes by $sc_i = \rho_i \cdot \delta_i$. In this formulate, ρ_i measures the local density of node by $\rho_i = \sum_j \psi(d(i,j) - d_c)$ where $\psi(x) = 1$ if $x \leqslant 1$ and $\psi(x) = 0$ otherwise, d_c denotes a cutoff distance. Basically, ρ_i equals the number of nodes that are closer than d_c to node i. We find that the results of the algorithm are robust with respect to the choice of d_c. Here we set $d_c = 1$, and the node density is equivalent to degree centrality. δ_i further imposes restrictions on the distance between high-density nodes by $\delta_i = \min_{j:\rho_j>\rho_i} d(i,j)$. The definition of structural centrality ensures that the selected nodes are located in the center of communities and maintains a relatively large distance from each other. Assuming that the number of identified structure centers is K, we assign unique community labels to structural centers and construct an initial label set $l = \{1, 2, \ldots, C\}$ where C denotes the label of K-th community. Finally, we add the nearest neighbors of the structural center k with label $k \in l$ into the training set so as to accelerate the converge of the GCN model.

Algorithm 2: Label Sampling Model via Structural Center Location

Require: $A \in \mathbb{R}^{N\times N}$: an adjacent matrix;
Ensure: y_t : labeled nodes;
  $d_{ij} \leftarrow D(\mathscr{G})$;
2: $sc_i \leftarrow \rho_i \cdot \delta_i$;
  Select K structural centers as initial labels;
4: **for** each structure center k **do**
  $N(k) \leftarrow Neighbors(k)$;
6: Find the top t vertices in $N(k)$;
  Add them to the training set with label k;
8: **end for**
  return labeled node set.

Note that the selection of the structural centers is the core of the algorithm. To describe the process of label sampling model intuitively, we apply the Algorithm 2 on Zacharys karate club network with 34 nodes and 78 edges, which is split into two ground-truth communities because of the disagreement between the administrator (vertex 1) and the instructor (vertex 34). Fig. 1(a) shows the plot of ρ_i as a function δ_i for each node embedded in a two-dimensional space. We can find that node 1 (orange point) and node 34 (green point) both have the local maxima in density and relative distance, which are identified as the structural centers in the decision graph. The result is consistent with the description of the original network. In addition, we observe that the structural centrality of the whole network follows an approximate power-law distribution in logarithmic coordinates (Fig. 1(b)). It is worth noting that the identified structural centers (nodes 1 and 34) are distinguished from other nodes, which makes the target nodes arise automatically. This observation helps to choose the exact structural centers at the label sampling stage.

4.3. Community detection via GCN

Our method falls into the framework of GCN with a label sampling model. However, there are several differences from the original GCN model. First, we build a GCN-based model for community detection, which is essentially a graph clustering task rather than node classification. Second, we directly optimize GCN on the balanced training set constructed by the label sampling process without additional data for validation, which can remedy the disadvantage of the GCN model in an unsupervised learning task. Moreover, we only consider the connected component of graphs in the graph convolution, so as to meet the densely connected properties of community structures.

Additionally, we follow the shallow architecture of the GCN model based on the following reasons. On one hand, a multi-layer GCN is powerful in extracting network features although the graph convolution is a localized filter. On the other hand, a deep GCN is difficult to train and does not improve the performance of the model. Graph convolution has been proved to be essentially a Laplacian smoothing, and repeatedly applying this smoothing operation may mix the features of nodes from different clusters and make them indistinguishable [53]. Therefore, a GCN model with many convolutional layers reduces the clustering accuracy. The final output of the shallow GCN model with two layers is formalized as follows:

$$Z = softmax\left(\widehat{A}ReLU\left(\widehat{A}XW^{(0)}\right)W^{(1)}\right), \tag{4}$$

where $\widehat{A}$ denote the normalized adjacency matrix by $\widehat{A} = D^{-1/2}\widetilde{A}D^{-1/2}$, $softmax(\cdot)$ and $ReLu(\cdot)$ are two activation functions defined by $softmax(x_i) = \frac{1}{\mathscr{Z}}exp(x_i)$ with $\mathscr{Z} = \sum_i exp(x_i)$ and $ReLu(x) = max(0, x)$, respectively. $W^{(0)}$ and $W^{(1)}$ are weight matrices that can be trained by gradient descent. Adam optimizer is used to train the GCN model. Here, we still adopt the cross-entropy error over all labeled nodes to evaluate the loss:

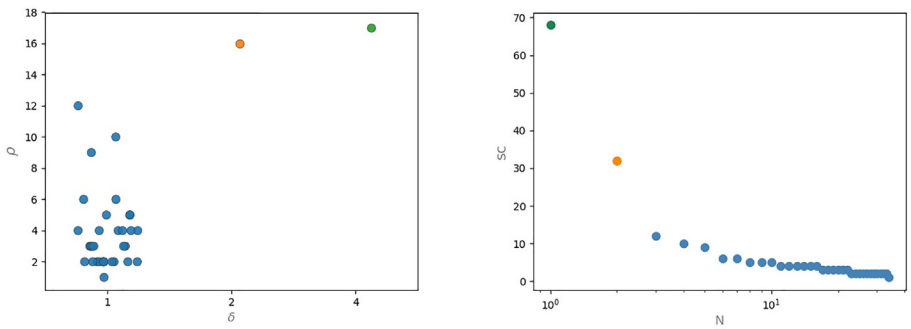

Fig. 1. Structural center location on the Karate network. (a) The decision graph for locating the structural centers. (b) The observation on the structural centrality on the real-world network.

$$\mathscr{L} = -\sum_{i\in V_l}\sum_{f=1}^{F} Y_{if} \ln Z_{if}, \tag{5}$$

where V_l denotes the set of indices corresponding to the labeled nodes, $Y \in \mathbb{R}^{|V_l|\times F}$ is the prior community label indicator matrix, and F is the dimension of the output features which actually equal to the number of communities. Then, we can train the GCN model with the constructed training set.

4.4. Computational cost

Our method combines the label sampling model with GCN into an unsupervised learning framework to uncover underlying community structures with topology and attribute information. It is hard to estimate the computational complexity of the proposed method, as it depends on the number of community structures, which essentially depends on the network currently being studied. Given a network $\mathscr{G} = (\mathscr{V}, \mathscr{X})$, we can analyze the computational cost of the algorithm from two stages: label sampling and GCN training. The overhead of the label sampling phase is the computational cost of the structural center location model by measuring the structural centrality of nodes, which requires approximately linear time by sparse representation. In experiments, the runtime is negligible on networks with a few thousand nodes. In the GCN training phase, we follow Kinf and Welling's setup with full-batch gradient descent, which scales linearly in the number of network edges. In our experiments, it generally converges fast with a few epochs as it propagates labels from the center of a community to the periphery. Moreover, the computation of GCN can be further accelerated with mini-batch stochastic gradient descent. Therefore, the running time of our algorithm is comparable to GCN.

5. Experiments

In this section, we conduct comparative experiments on various real-world networks to verify the performance of the proposed method in community detection. We also give some descriptions of network datasets, baseline methods, evolution criteria, and experimental setups.

5.1. Datasets and baseline methods

We conduct experiments on 12 real-world networks of different types to evaluate the proposed method from a comparison view. The basic information of these datasets is shown in Table 1. Five well-studied benchmark networks without attribute information, such as Karate, Dolphin, Polbooks, Adjnoun, and Football, are from Wang et al. [54] which show obvious community structures in topology. The datasets Cornell, Texas, Washington, and Wisconsin come from Wang et al. [40], which describe the connections of webpages generated by college students from several universities. Moreover, Three citation networks Cora, CiteSeer, and PubMed are from Kipf and Welling [51] which are commonly used for testing in network analysis. These benchmark networks have a lot of attribute information on nodes and show different distribution characteristics in topology. We follow Kipf and Welling on the proportion of available labeled information.

We also compare our method with various state-of-the-art community detection methods. The baseline methods include four types of methods. The first includes two traditional methods Lovain [29] and Infomap [34], which both cluster nodes based on network topological structure alone. Infomap is a multi-level network clustering based on the random walk and information theory. Lovain algorithm has been considered as one of the best modularity optimization methods in community detection. Especially, in order to show that GCN can achieve better performance based on the label sampling mechanism, we use a combined method (SCL + LEO) for community detection based on the structural center location (SCL) and a local expansion optimization (LEO) of a fitness function which is usually used to measure local community structures in community detection [33]. The second type includes two graph embedding methods DeepWalk [11] and MGAE [18] which both use graph neural networks for graph representation and a clustering algorithm for node clustering. The third type includes SCI [38] and NEM [39] which identify communities with both topology and attribute information of a network. The above three types of methods perform node clustering in an unsupervised manner, and all nodes will be considered in performance evaluation. The fourth type includes the original GCN [51], and DIG [53] and WSC [40], which are semi-supervised methods on attribute networks where test sets are usually used to evaluate their performance.

5.2. Evaluation Criteria and Experimental Setups

In this work, we focus on community detection on various real-world networks, all of which have known community structures and labels. The problem of community detection is often regarded as a clustering task that divides nodes into different clusters based on topology and attribute information. And thus detected results are evaluated by measuring the consistency with the ground-truth partitions. In our experiment, in addition to clustering Accuracy (AC) that is commonly used to evaluate the performance of various algorithms in machine learning, we also adopt the Normalized Mutual Information (NMI) [5] to evaluate the effectiveness of each method in community detection. The NMI index is an metric based on information theory and is widely used to measure the similarity between the ground-truth community partition of a net-

Table 1
The basic information on the real-world networks.

Dataset	Nodes	Edges	$\langle k \rangle$	cc	Communities	Attributes
Karate	34	78	4.58	0.57	2	\
Dolphin	62	159	5.13	0.25	2	\
Polbooks	105	441	8.40	0.49	3	\
Adjnoun	112	425	7.59	0.17	2	\
Football	115	613	10.66	0.40	11	\
Cornell	195	286	2.93	0.16	5	1,703
Texas	187	328	3.19	0.19	5	1,703
Washington	230	446	3.63	0.20	5	1,703
Wisconsin	265	530	3.62	0.21	5	1,703
Cora	2,708	5,429	3.89	0.24	7	1,433
Citeseer	3,312	4,732	2.81	0.14	6	3,703
PubMed	19,729	44,338	4.49	0.06	3	500

In this table,$\langle k \rangle$ is the average degree of nodes, "'cc" denotes the average clustering coefficient of the network, "Attributes" represents the dimension of attribute features, and "'\"indicates missing attribute information.

work and the detected partition from an algorithm. Given two partitions A and B, the metric NMI can be defined as:

$$NMI(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B}\mathcal{N}_{ij}\log\left(\frac{\mathcal{N}_{ij}N}{\mathcal{N}_{i.}\mathcal{N}_{j}}\right)}{\sum_{i=1}^{C_A}\mathcal{N}_{i.}\log\left(\frac{\mathcal{N}_{i.}}{N}\right)+\sum_{j=1}^{C_B}\mathcal{N}_{j}\log\left(\frac{\mathcal{N}_{j}}{N}\right)}, \quad (6)$$

where $\mathcal{N}$ is the confusion matrix whose element $\mathcal{N}_{ij}$ denotes the number of nodes in common between the community C_i and C_j, $\mathcal{N}_{i.}(\mathcal{N}_{j})$ represents the sum of the row i (column j) of matrix $\mathcal{N}$, and $C_A(C_B)$ is the number of communities in partition A (B). The index $NMI(A,B)$ ranges from 0 to 1 and it gets the maximum value 1 when A and B are exactly the same.

In our experiment, for these networks that don't have attribute information, we initialize the attribute matrix as $X = I$ where I is the identity matrix. For the citation networks including CiteSeer, Cora, and PubMed, we use the datasets provided by Yang et al. In training, we take a 2-layer GCN with randomly initialized weights, and follow the same hyper-parameters as Kipf and Welling with learning rate 0.01, maximum epochs 200, dropout rate 0.5, L2 regularization weight 5×10^{-4}, and 16 hidden units. For each run, we randomly divide labels into a small set for training, a large set for testing. We adopt Adam optimizer to train the GCN-based models and run experiments on TensorFlow. To ensure the stability of results, the proposed method and all compared algorithms have been independently run 10 times on each network. All the experiments are conducted on a PC with a 2.6 GHz, i7-5600, Quad-core CPU, and 12 GB of RAM.

5.3. Experiment results

We compare the performance of our method denoted by SGCN for short with other state-of-the-art algorithms on the twelve real-world networks. The experimental results are obtained based on the same experimental setups and are shown in Table 2 and 3. Additionally, the average performance of these methods is also shown in these tables. We choose different t for different networks with a criterion which will be discussed in detail later. Actually, the number of labels required by our method is far less than the number required by other GCN-based algorithms to achieve comparable performance, especially on small networks. For example, our method gets ideal results on the Karate network with only two structural centers as the training set. This shows that the GCN model can effectively propagate labels to the entire graph from structural centers with few training labels.

Measured in clustering accuracy AC, the proposed method SGCN outperforms other methods on 10 of 12 networks, obtains perfect accuracy like DIG on the Karate network, and performs the second-best on the remaining network Texas, as is shown in Table 2. The experimental comparison shows the methods that make use of both topological structures and attribute information perform better than those based on network topological information only. It further verifies that attribute information can help uncover underlying community information and improve algorithmic performance in community detection. Additionally, the semi-supervised learning methods generally outperform these methods that do not utilize prior label information. Based on label sampling by structural center location, SGCN is on average 9.7%, 6.1% more accurate than GCN and DIG, respectively. This verifies our analysis that the GCN model can effectively propagate labels to the entire graph based on the constructed training set with a small size. Moreover, it is worth noting that the combined method SCL + LEO, like other traditional methods, cannot perform well in the community detection task on attributed networks. It can be shown that SGCN improves over SCL + LEO by 39.4% on average.

In terms of partition similarity of communities, the NMI scores show a similar trend as in AC. The two topology-based methods Infomap and Lovain both obtain poor results, especially on the last seven attributed networks. They usually identify a large number of small communities on these networks which essentially show weak community structures. The results of SCI and NEM show obvious improvement due to the use of attribute information, but still inferior to semi-supervised methods. As shown in Table 3, SGCN performs the best on 9 of the 12 real-world networks and is also competitive on the remaining 3 networks. We highlight that SGCN outperforms GCN-based methods, which further validate the effectiveness of the label sampling strategy. Compared with the AC index, however, it shows the difference of performance in the NMI score aspect since most of these NMI values range between 40 and 100 points. By analyzing these networks, it is found that the deviation of performance in the NMI value is mainly due to the difference in degree distribution and clustering coefficient. As shown in Table 1, these networks with low degree distribution and low clustering coefficients, such as Cornell, Citeseer, and PubMed, often get low performance in NMI value.

5.4. Parameter analysis

As described in Algorithm 2, we optimize our GCN-based method on the expanded label set constructed by the label sampling model. The unsupervised learning framework has only one parameter t to control the numbers of labels and does not require any additional labels for validation. Generally, a good GCN model can be trained as long as there are enough labeled nodes. It's hard to get enough labels for this training task, but it's possible to estimate the lower bound of the number of labels. Here, we follow the criterion for choosing t proposed by Li et al. [53]. Given a network with the average degree $\langle k\rangle$ and a GCN with the number of layers τ, the lower bound of t is determined by $\langle k\rangle^{\tau} * t \approx N$. Its basis is to

Table 2
Comparison of algorithms on real-world networks in terms of clustering accuracy (AC: %).

Dataset	Infomap	Louvain	SCL + LEO	SCI	NEM	DeepWalk	MGAE	GCN	DIG	WSC	SGCN
Karate	79.4	50.0	88.1	85.0	89.1	88.2	91.3	87.5	**100**	92.8	**100**
Dolphin	43.5	21.0	79.0	81.3	92.5	88.7	92.7	93.8	94.5	94.3	**96.7**
Polbooks	46.7	41.0	40.8	68.4	76.7	43.8	79.6	81.8	84.1	85.7	**88.6**
Adjnoun	50.9	17.9	36.7	51.2	48.6	26.8	60.4	40.4	57.7	65.6	**71.9**
Football	73.9	67.4	70.3	78.7	81.2	66.5	76.7	88.6	85.7	90.8	**91.4**
Cornell	9.23	15.4	16.8	36.9	47.2	31.8	48.2	46.3	48.4	53.9	**62.5**
Texas	6.95	19.7	20.5	49.7	53.6	32.6	56.7	57.1	62.3	**77.5**	67.4
Washington	6.09	12.4	15.7	46.1	42.9	35.0	50.8	54.9	56.5	58.3	**64.3**
Wisconsin	5.9	15.9	16.6	46.4	63.4	28.7	58.8	55.6	57.7	61.9	**65.6**
Cora	0.44	7.16	7.34	41.7	57.6	46.7	63.4	81.5	81.7	53.7	**87.5**
Citeseer	1.29	4.57	6.03	34.4	49.5	36.2	63.6	70.3	71.2	47.6	**75.9**
PubMed	5.3	7.46	9.82	47.3	65.7	61.9	43.9	79.0	79.2	60.7	**81.2**
Avg.	27.5	23.3	40.0	55.6	64.0	48.9	65.5	69.7	73.3	70.2	**79.4**

The proposed method is denoted by SGCN for short. bold value in each row denotes the best result on the corresponding dataset.

Table 3
Comparison of algorithms on real-world networks in terms of partition similarity (NMI: %).

Dataset	Infomap	Louvain	SCL + LEO	SCI	NEM	DeepWalk	MGAE	GCN	DIG	WSC	SGCN
Karate	61.1	53.4	58.1	58.3	65.0	58.1	61.8	63.7	**100**	73.2	**100**
Dolphin	62.8	56.4	75.0	59.9	62.7	61.4	65.3	69.9	72.5	70.3	**77.7**
Polbooks	50.3	51.6	45.7	54.7	59.2	58.9	60.5	61.6	62.4	61.5	**65.8**
Adjnoun	0.43	0.41	18.1	25.7	31.0	25.4	32.2	34.7	36.3	33.4	**41.6**
Footall	90.4	89.1	88.0	81.5	87.4	86.2	88.3	91.6	92.5	92.7	**94.1**
Cornell	21.3	14.9	16.2	15.2	18.7	7.3	16.5	9.10	12.1	21.6	**33.7**
Texas	16.6	21.3	20.7	22.0	35.1	5.6	32.7	5.0	5.4	**53.7**	48.5
Washington	19.0	20.7	18.8	21.0	21.2	5.9	17.4	16.3	25.2	24.1	**40.3**
Wisconsin	18.2	19.1	19.8	18.5	38.0	4.3	21.8	17.8	25.7	31.4	**44.6**
Cora	47.4	39.8	42.5	17.8	44.1	32.7	48.9	54.5	**62.5**	52.5	61.4
Citeseer	39.9	34.7	31.3	9.2	24.3	9.7	41.6	42.3	45.4	35.3	**51.5**
PubMed	21.8	17.7	21.6	28.3	31.9	16.7	23.0	26.0	38.0	34.2	**40.8**
Avg.	37.4	34.9	37.9	34.3	43.2	31.0	42.5	41.0	48.2	48.7	**58.3**

estimate how many labels are needed for a GCN model to propagate them to cover the entire graph.

6. Conclusion

In this work, we proposed a task-specified network representation learning method based on GCN for the problem of community detection in attributed networks without prior knowledge. Inspired by the message pass mechanism of GCN and the local self-organizing property of community structure, we integrated a label sampling model and shallow GCN into an unsupervised learning framework to uncover underlying community structures by fusing topology and attribute information. The label sampling model constructed a balanced training set by structural center location and neighbor node expansion to train the GCN model when there is no available label information. This approach also leads to new solutions to unsupervised learning in graphs. Experimental results showed the proposed method achieved competitive performance on many real-world networks, compared with the state-of-the-art community detection algorithms.

We would like to emphasize that our method provides a general framework that yields a class of algorithms. For instance, one could choose a different method for label sampling or a different optimization procedure for the structural center location. It should be noted that the proposed method maybe not be expected to perform well when there are 'weak' communities (the characteristics of community structure are very vague) in networks, although almost all community detection algorithms face such a challenge. In such a case, the label sampling algorithm may not be able to find all the structural centers, resulting in the subsequent GCN model failing to identify all communities correctly.

In future work, we intend to study deep nonlinear on graphs and develop new convolutional filters to be compatible with deep GCN architectures. Moreover, the combination of community detection and deep learning helps to solve various problems such as anomaly detection, commercial recommendation, and epidemic spreading, so the graph-based application of community detection is a major concern in our further research.

CRediT authorship contribution statement

Xiaofeng Wang: Conceptualization, Methodology, Writing - original draft. **Jianhua Li:** Supervision. **Li Yang:** Software, Validation. **Hongmei Mi:** Investigation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the reviewers for their insightful comments and useful suggestions. This research was supported by Natural Science Foundation of Zhejiang Province (No. LQ20F020021 and No.LY19F030012) and NSAF Joint Fund (No. U20B2048).

References

[1] M.E.J. Newman, Networks, Second ed., Oxford University, Oxford, 2018.
[2] Y.Y. Liu, J.J. Slotine, A.L. Barabasi, Controllability of complex networks, Nature 473 (2011) 167–173, https://doi.org/10.1038/nature10011.
[3] K. He, Y. Li, S. Soundarajan, J.E. Hopcroft, Hidden community detection in social networks, Inf. Sci. 425 (2018) 92–106, https://doi.org/10.1016/j.ins.2017.10.019.
[4] T. Hoffmann, L. Peel, R. Lambiotte, N.S. Jones, Community detection in networks without observing edges, Sci. Adv. 6 (4) (2020) eaav1478, https://doi.org/10.1126/sciadv.aav1478.
[5] S. Fortunato, D. Hric, Community detection in networks: A user guide, Phys. Rep. 659 (2016) 1–44, https://doi.org/10.1016/j.physrep.2016.09.002.
[6] L. Peel, D.B. Larremore, A. Clauset, The ground truth about metadata and community detection in networks, Sci. Adv. 3 (5) (2017), https://doi.org/10.1126/sciadv.1602548, e1602548.
[7] F.D. Malliaros, M. Vazirgiannis, Clustering and community detection in directed networks: A survey, Phys. Rep. 533 (4) (2013) 95–142, https://doi.org/10.1016/j.physrep.2013.08.002.
[8] D. Zhang, J. Yin, X. Zhu, C. Zhang, Network representation learning: a survey, IEEE Trans. Big. Data 6 (2018) 3–28, https://doi.org/10.1109/TBDATA.2018.2850013.
[9] P. Cui, X. Wang, J. Pei, W. Zhu, A Survey on Network Embedding, IEEE Trans. Knowl. Data. Eng. 31 (5) (2018) 833–852, https://doi.org/10.1109/TKDE.2018.2849727.
[10] W. Zhu, X. Wang, and P. Cui, Deep Learning for Learning Graph Representations, in: W. Pedrycz, Shyi-Ming Chen (Eds.), Deep Learning: Concepts and Architectures, Springer, 2020, pp. 169–210.
[11] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of SIGKDD, 2014, pp. 701–710.
[12] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: Large-scale information network embedding, in: Proceedings of WWW, 2015, pp. 1067–1077, https://doi.org/10.1145/2736277.2741093.
[13] L.F. Ribeiro, P.H. Saverese, D.R. Figueiredo, struc2vec: Learning node representations from structural identity, in: Proceedings of SIGKDD, 2017, pp. 385–394, https://doi.org/10.1145/3097983.3098061.
[14] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, S. Yang, Community preserving network embedding, in: Proceedings of AAAI, 2017, pp. 203–209.
[15] C. Yang, Z. Liu, D. Zhao, M. Sun, E.Y. Chang, Network representation learning with rich text information, in: Proceedings of IJCAI, 2015, pp. 2111–2117.
[16] S. Cao, W. Lu, Q. Xu, Deep neural networks for learning graph representations, in: Proceedings of AAAI, 2016, pp. 1145–1152.
[17] D. Wang, P. Cui, W. Zhu, Structural deep network embedding, in: Proceedings of SIGKDD, 2016, pp. 1225–1234, https://doi.org/10.1145/2939672.2939753.
[18] C. Wang, S. Pan, G. Long, X. Zhu, J. Jiang, MGAE: Marginalized Graph Autoencoder for Graph Clustering, in: Proceedings of CIKM, 2017, pp. 889–898, https://doi.org/10.1145/3132847.3132967.
[19] J. Cao, D. Jin, L. Yang, J. Dang, Incorporating network structure with node contents for community detection on large networks using deep learning, Neurocomputing 297 (2018) 71–81, https://doi.org/10.1016/j.neucom.2018.01.065.
[20] W. Ding, C. Lin, P. Ishwar, Node embedding via word embedding for network community discovery, IEEE Trans. Signal. Inf. Process. Netw. 3 (3) (2017) 539–552, https://doi.org/10.1109/TSIPN.2017.2731163.

[21] Y. Xie, M. Gong, S. Wang, B. Yu, Community discovery in networks with deep sparse filtering, Pattern Recogn. 81 (2018) 50–59, https://doi.org/10.1016/j.patcog.2018.03.026.
[22] L. Yang, X. Cao, D. He, Modularity Based Community Detection with Deep Learning, in: Proceedings of IJCAI, 2016, pp. 2252–2258.
[23] F. Liu, S. Xue, J. Wu, C. Zhou, W. Hu, C. Paris, et al., Deep Learning for Community Detection: Progress, Challenges and Opportunities, in: Proceedings of IJCAI, 2020, pp.4981-4987. https://doi.org/10.24963/ijcai.2020/693.
[24] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, et al., Graph neural networks: A review of methods and applications, 2019. https://arxiv.org/abs/1812.08434.
[25] D. Jin, Z. Liu, W. Li, D. He, W. Zhang, Graph convolutional networks meet Markov random fields: Semi-supervised community detection in attribute networks, Proceedings of AAAI (2019) 152–159, https://doi.org/10.1609/aaai.v33i01.3301152.
[26] H. Sun, F. He, J. Huang, Y. Sun, Y. Li, C. Wang, et al., Network Embedding for Community Detection in Attributed Networks, ACM Trans. Knowl. Discov. Data 14 (2020) 1–25, https://doi.org/10.1145/3385415.
[27] D. Jin, B. Li, P. Jiao, D. He, H. Shan, Community Detection via Joint Graph Convolutional Network Embedding in Attribute Network, in: Proceedings of ICANN, 2019, pp. 594–606.
[28] J. Luo, Y. Du, Detecting community structure and structural hole spanner simultaneously by using graph convolutional network based Auto-Encoder, Neurocomputing 410 (2020) 138–150, https://doi.org/10.1016/j.neucom.2020.05.039.
[29] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. 10 (2008) P10008, https://doi.org/10.1088/1742-5468/2008/10/P10008.
[30] X. Wang, G. Liu, L. Pan, J. Li, Uncovering fuzzy communities in networks with structural similarity, Neurocomputing 210 (2016) 26–33, https://doi.org/10.1016/j.neucom.2016.01.109.
[31] N. Binesh, M. Rezghi, Fuzzy clustering in community detection based on nonnegative matrix factorization with two novel evaluation criteria, Appl. Soft. Comput. 69 (2018) 689–703, https://doi.org/10.1016/j.asoc.2016.12.019.
[32] E. Sara, Garza and Satu Elisa Schaeffer, Community detection with the Label Propagation Algorithm: A survey, Phys. A 534 (2019), 122058.
[33] W. Luo, N. Lu, L. Ni, et al., Local community detection by the nearest nodes with greater centrality, Inf. Sci. 517 (2020) 377–392, https://doi.org/10.1016/j.ins.2020.01.001.
[34] M. Rosvall, C.T. Bergstrom, Maps of random walks on complex networks reveal community structure, PNAS 105 (4) (2008) 1118–1123, https://doi.org/10.1073/pnas.0706851105.
[35] A. Saxena, S. Iyengar, Centrality Measures in Complex Networks, A Survey (2020), https://arxiv.org/abs/2011.07190..
[36] J. Yang, J. McAuley, J. Leskovec, Community detection in networks with node attributes, in: Proceedings of ICDM, 2013, pp. 1151–1156.
[37] M. Azaoouzi, D. Rhouma, L.B. Romdhane, Community detection in large-scale social networks: state-of-the-art and future directions, Social Network Anal. Mining 23 (2019) 9.
[38] X. Wang, D. Jin, X. Cao, L. Yang, W. Zhang, Semantic community identification in large attribute networks, in: Proceedings of AAAI, 2016, pp. 265–271.
[39] D. He, Z. Feng, D. Jin, X. Wang, W. Zhang, Joint Identification of Network Communities and Semantics via Integrative Modeling of Network Topologies and Node Contents, in: Proceedings of AAAI, 2017, pp. 116–124.
[40] W. Wang, X. Liu, P. Jiao, X. Chen, D. Jin, A Unified Weakly Supervised Framework for Community Detection and Semantic Matching, in: in: Proceedings of PAKDD, 2018, pp. 218–230.
[41] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of SIGKDD, 2016, pp. 855–864, https://doi.org/10.1145/2939672.2939754.
[42] T. Lyu, Y. Zhang, Y. Zhang, Enhancing the network embedding quality with structural similarity, in: Proceedings of CIKM, 2017, pp. 147–156, https://doi.org/10.1145/3132847.3132900.
[43] J. Li, J. Zhu, B. Zhang, Discriminative deep random walk for network classification, in: Proceedings of ACL, 2016, pp. 1004–1013, https://doi.org/10.18653/v1/P16-1095.
[44] C. Li, S. Wang, D. Yang, Z. Li, Y. Yang, X. Zhang, J. Zhou, PPNE: Property preserving network embedding, in: Proceedings of DASFAA, 2017, pp. 163–179.
[45] C. Donnat, M. Zitnik, D. Hallac, J. Leskovec, Spectral graph wavelets for structural role similarity in networks, in: Proceedings of ICLR, 2017.
[46] Z. Zhang, H. Yang, J. Bu, S. Zhou, P. Yu, J. Zhang, M. Ester, C. Wang, ANRL: Attributed Network Representation Learning via Deep Neural Networks, in: Proceedings of IJCAI, 2018, pp. 3155–3161.
[47] L. Liao, X. He, H. Zhang, T.-S. Chua, Attributed Social Network Embedding, IEEE Trans. Knowl. Data. Eng. 30 (12) (2018) 2257–2270, https://doi.org/10.1109/TKDE.2018.2819980.
[48] R.A. Rossi, R. Zhou, N. Ahmed, Deep inductive graph representation learning, IEEE Trans. Knowl. Data. Eng. 32 (3) (2018) 438–452, https://doi.org/10.1109/TKDE.2018.2878247.
[49] Z. Yang, W-W. Cohen, R Salakhutdinov, Revisiting semi-supervised learning with graph embeddings, 2016, pp.40-48, https://arxiv.org/abs/1603.08861.
[50] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Proceedings of NIPS, 2016, pp. 3844–3852, https://arxiv.org/abs/1606.09375.
[51] T.N. Kipf and M. Welling, Semi-supervised classification with graph convolutional networks, in: Proceedings of ICLR, 2017. https://arxiv.org/abs/1609.02907.
[52] D. He, Y Song, Jin D, et al., Community-Centric Graph Convolutional Network for Unsupervised Community Detection. In: Proceedings of IJCAI, 2020, PP.3515-3521. https://doi.org/10.24963/ijcai.2020/486.
[53] Q. Li, Z. Han, X.-M. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, in: Proceedings of AAAI, 2018, pp. 3538–3545, https://arxiv.org/abs/1801.07606.
[54] X. Wang, G. Liu, J. Li, J.P. Nees, Locating Structural Centers: A Density-Based Clustering Method for Community Detection, PLoS ONE 12 (1) (2017), https://doi.org/10.1371/journal.pone.0169355, e0169355.

Xiaofeng Wang received his M.S. degree in communication and information system from Yunnan University, Kunming, China in 2012, and his Ph.D. in information and communication engineering from Shanghai Jiao Tong University (SJTU), Shanghai, China in 2017. He is currently working as a Lecturer in the School of Information Engineering, at China Jiliang University, Hangzhou, China, a Postdoctoral Research Fellow in Computer Science at SJTU, and is a visiting scholar in Social Network Analysis from Concordia University, Canada. His research interests include network science, Data Mining, and artificial intelligence, and machine learning.

Jianhua Li is currently a full Professor with the Department of Electrical Engineering and Cyber Security, Shanghai Jiao Tong University, Shanghai, China. He is also the director of the National Engineering Laboratory of Information Content Analysis Technology. He received his Ph.D. from the Department of Electrical Engineering, Shanghai Jiao Tong University in 1998. He has published nearly 300 research articles in major peer-reviewed international journals and international conferences. His research interests include cyberspace security, big data analysis, information security, and privacy.

Li Yang received his M.S. degree in pattern recognition and intelligent systems from Wuhan University, Hubei, China in 2004, and his Ph.D. from Zhejiang University, Hangzhou, China in 2013. He is currently working as an Associate Professor with the College of Information Engineering, China Jiliang University, Hangzhou, China. His research interests include computer vision, data mining, and robotics.

Hongmei Mi received her M.S. degree in signal and information processing from Dalian University of Technology, Dalian, China in 2011, and her Ph.D. in computer science from Rouen University, Rouen, France in 2016. She is currently working as a Lecturer in the School of Information Engineering, at China Jiliang University, Hangzhou, China. Her research interests include signal and image processing, medical image processing, and machine learning.