A Community Detection Method for Social Network Based on Community Embedding

Meizi Li, Shuyi Lu, Lele Zhang, Yuping Zhang, and Bo Zhang, *Member, IEEE*

***Abstract*— Most community detection methods focus on the similarities between detection nodes to achieve community partitioning. Traditional network representation learning methods are also limited to the local context of the central nodes, which results in less truly representative results. This article examines nodes' influence information, nodes' community affiliating information, and similarity of community topologies and proposes a more effective node representation strategy. According to the local node information and global topology in the social network graph, a method of combining local node embedding and global community embedding is also designed. The effectiveness of learning node representation and community representation is improved by our approach. The proposed model can also effectively detect overlapping communities.**

***Index Terms*— Community embedding, node belonging community information, node embedding, node influence information, overlapping community detection.**

I. Introduction

THE quality of the corpus is of interest because it directly affects the representation learning results of the nodes. The representation of a node is determined by its neighbors and its community. Only node embedding cannot address similar attributes between nodes in the same community; community detection, the representation, and the distribution of nodes and community can help tackle this problem.

This study intends to: 1) study the influence of node information fusion; 2) improve the quality of node learning by embedding the community information of nodes; and 3) obtain a stable community distribution of each node, that is, to finally complete community detecting. To achieve the three aims, we design a model in three steps. First, a new, high-quality, and semantically rich corpus is constructed based on a social network graph. Second, the attribute information and the community attribute information of the nodes are learned at the same time. Third, when the model reaches convergence, the community distribution of each node is obtained.

Traditional network representation learning (NRL)-based community detection methods use depth-first and breadth-first searching strategies when generating corpora. These strategies will incur random selection errors during traversing selected nodes, resulting in poor-quality corpora. In addition, these NRL-based methods simply learn a single index of the topology of the nodes without considering the relationship between the nodes and the community, and their final detected communities can hardly be called ideal.

In this article, we propose a new NRL-based community detection method to provide a high-quality corpus and a satisfying detected community. First, a corpus is constructed. Second, the many-to-many relationships between nodes and communities are incorporated into representation learning of the nodes. Third, when the model reaches convergence, all vector representations and the community distribution of the nodes can be learned to detect the community. The framework can also learn the vector representation of the community.

The main contributions of this study are summarized as follows.

1) Considering the local topological structure of the network and the influence of the nodes' attribute information, we propose a more effective node searching strategy that can avoid reconstructing a new corpus due to the errors.
2) Combining the information of many-to-many relationships between nodes and communities, we design a method that combines local node and global community embedding so that the learned nodes can indicate that they have community information.
3) The novel NPL-based model can detect communities in a more effective way and identify overlapping communities as well.

II. Related Work

A. Traditional Community Detection Algorithms

A social network's structure consists of users and relationships. Some relationships are dense, while others are sparse. A tightly connected part can be seen as a community, and overlapping communities have common nodes. For a given social network, community detection can be seen as a process of clustering.

Algorithms for community detection include the GN algorithms and the label propagation (LC) algorithms [1], [2].

Manuscript received May 16, 2020; revised October 20, 2020 and December 26, 2020; accepted December 29, 2020. Date of publication February 8, 2021; date of current version April 1, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61802258, Grant 61572326, and Grant 61702333; in part by the Natural Science Foundation of Shanghai under Grant 18ZR1428300; and in part by the Shanghai Committee of Science and Technology under Grant 17070502800. *(Corresponding author: Bo Zhang.)*

Meizi Li, Shuyi Lu, Lele Zhang, and Yuping Zhang are with the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 200234, China (e-mail: limeizi@shnu.edu.cn; shuyi_lu@126.com; claire_zll@163.com; yp_zhang@shnu.edu.cn).

Bo Zhang is with the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 200234, China, also with the Institute of Artificial Intelligence on Education, Shanghai Normal University, Shanghai 200234, China, and also with the Shanghai Engineering Research Center of Intelligent Education and Bigdata, Shanghai Normal University, Shanghai 200234, China (e-mail: zhangbo@shnu.edu.cn).

Digital Object Identifier 10.1109/TCSS.2021.3050397

In the field of module-based community detection algorithms, Newman *et al.* [1] use modularity to evaluate the partitioning results of a community network. The larger the value of the modularity, the better the results of the community detection. The Leuven algorithm is a graph model based on modularity. It divides the communities continuously to increase the modularity [3]. These traditional algorithms have been applied to different scenarios, only with different ratios of merging surrounding nodes [4].

Most researchers who study nonoverlapping community detection have used the LC algorithm [5]–[9]. Its basic idea is that the label of a node itself depends on the largest number of labels in its neighbor nodes. The algorithm does not need any prior parameters and community indicators, and it has a short convergence period. However, it is difficult to estimate its iteration times.

Given the unstable results that the LC algorithm produces, researchers have designated an improved version with better stability and accuracy. The LabelRank algorithm is based on the LC and the Markov random walk [5], [10]. Despite its higher computational complexity and probability for each label, this algorithm can provide more accurate results than the LC algorithm [11], [12].

The above studies target the detection of nonoverlapping communities and have provided satisfying results. However, real social networks contain more diversified communities, such as overlapped ones, and researchers have veered their attention to overlapping community detection problems [13]–[15].

Spectral clustering, a widely used algorithm in clustering, is evolved from graph theory. Its main idea is that the edge weight between two nodes is lower if they are farther away from each other. Conversely, the edge weight is higher if the two nodes are closer. Based on the evolution of the clustering algorithm, the SAEC algorithm transforms the topology of nodes into a set of edges [13]. By calculating the similarity matrix between nodes, the algorithm obtains the probability transfer matrix, then classifies the edges into corresponding communities by spectral clustering, and, finally, achieves the goal of overlapping community detection.

Earlier studies ignored the problem of the exponential growth of label space. To solve the problem, an algorithm based on latent features has emerged [16], [17]. Based on the network generation model in overlapping communities, the algorithm maximizes the generation probability of the entire network. It then proposes an optimal objective function to infer the potential features of each node. In addition, the network is introduced into the bipartite graph. To optimize the objective function, the algorithm analyzes the lower limit of the feature number [14]. Subsequently, an overlapping detection algorithm with hierarchical cohesion clustering appeared based on the local optimal expansion cohesion. The gist of this algorithm is that initialization should first construct the most important node and its neighbors and then the node attribution, until the termination condition of the algorithm is satisfied. However, there are several drawbacks to these algorithms. First, they are usually too large, excessively overlapped and cannot guarantee the stability of multiple operations [15]. Second, they have to consume a large space to store and deal with the network topology before they could present the final results of community detection. NRL-based community discovery methods have come into being to address these limits.

B. Network Representation Learning

Traditional machine learning-based classification methods learn how to map the samples' attributions to the classification labels. It is not fit for social networks, given fewer attributions within them. The Word2vec model proposed by Mikolov *et al.* [18] has become a popular tool in the field of natural language processing (NLP). It vectorizes all words so that the relationships between words can be quantitatively measured. It can visually represent the nodes and the relationships by the low-dimensional vectors. It can also be easily inputted into the machine learning model. The obtained vector representation can be performed for tasks, such as community detection [22]–[24].

With the development of the NLP, an increasing amount of NLP methods [18]–[21] have been proposed. For example, after the word embedding technology was proposed, Perozzi *et al.* [27] formulated the DeepWalk algorithm. The DeepWalk algorithm was the first method based on Word2vec to vectorize nodes in a social network. The main idea is to use the random walk path of nodes to simulate the text generation process. It generates a random walk sequence that only depends on the local information. It can save computation time and space consumption.

However, the completely random walk in the traversal strategy cannot adequately reflect the neighbor information of a node. The existing literature on NRL has gradually become more extensive. For example, the LINE algorithm [25] improves the first-order similarity sparse problem of DeepWalk. It uses the second-order similarity to compensate for the sparse problem of the first-order similarity. To optimize the learning of node features, it combines the first-order similarity and the second-order similarity in the objective function. The LINE algorithm can obtain better results of node label prediction than DeepWalk. To refine the random walk strategy in DeepWalk, Grover and Leskovec [26] proposed the node2vec algorithm that controls the direction of random walk by defining the offset function, so as to achieve a balance between the depth-first search and breadth-first search. It also considers the local and macrostructures information of the nodes, thereby ensuring the proximity and isomorphism in the social network structure [28]–[30].

The DeepWalk algorithm traverses the nodes randomly and deviates from the evolution of the actual social relationship. As a result, it fails to integrate the total network topology information [31]. By introducing the community information of the nodes, researchers proposed an improved version of DeepWalk, a unified NRL framework called the enhanced community embedded NRL model. The model is based on NRL and text modeling. With its inner structure, it can simultaneously detect the community distribution of each node, and it learns the embedding of the nodes together with communities.

It can consider local and global information effectively in social networks.

Although NRL has broad application, existing algorithms still face challenges in retaining information. In order to learn more information about the nodes, the algorithm should retain not only the topology information of the network but also the attributions of the nodes. However, the two factors are heterogeneous, and combining them effectively is a huge challenge. In this article, we propose a new algorithm that can effectively combine the attribution and the topology information. It improves the quality of the learning corpus and the quality of the learning results of the nodes' representation. Furthermore, it optimizes the node vector representation by combining the attribution of the community and can achieve the goal of community detection.

III. Formulation and Method

The nomenclature in community detection and NRL is introduced as follows. For a given graph $G = (V, E)$ representing a social network, the nodes and edges are denoted as v and e, respectively, and the sets of nodes and edges are denoted as V and E, respectively. The probability map is a probability distribution represented by a graph, and $P = (u, v)$ represents the joint distribution of the node u and the node v in the $V \times V$ vector space. The degree distribution of the network map refers to the probability distribution of the number of edges connected to the node when the node is randomly extracted from the network. The degree distribution of the scale-free network satisfies the power-law distribution, i.e., the probability is proportional to the power defined as follows:

$$P(d = k) \propto k^{-\alpha}. \tag{1}$$

The degree distribution of a scale-free network is discretely distributed: most nodes have fewer connections, while fewer nodes have more connections.

This distribution is similar to the word frequency in linguistics where only a few words are frequently used, while most of the words are not. Therefore, we can treat nodes in the network as words in the text and traverse the sequences in the network as sentences in the text. In word prediction, the Word2vec model optimizes the training process according to a specific corpus and outputs the words in an expression of the vector. The goal of the Skip-Gram model is to provide a training sample that predicts the probability of the next word based on the central word. These can be adapted to the network graph where the probability of the next node is predicted based on the central node.

The NRL method proposed in this article is based on the undirected and unweighted network graph. Our goal is to convert the node into a low-latitude feature representation according to the mapping function $f(u)$. Let $f(u)$ be a mapping function that maps the central node to a low-latitude vector and define $Ns(u)$ as the set of the neighbor nodes of node u. A set of the neighbor nodes of vertex u is obtained by sampling the sampling strategy S.

A. Node2vec

In the network $G = (V, E)$, based on the idea of the Word2vec model, the node2vec algorithm uses the Skip-Gram model to process texts in the NLP domain, extracting continuous feature representations. The algorithm compares the network to the text. The nodes in the network are similar to the words in the text, and the edges between the nodes are similar to the sentences in the text. The DeepWalk algorithm uses a random walk method to artificially define the sequence length of the walk access. It is used as a graph-based corpus to obtain different node access sequences. The Skip-Gram model predicts the context of the central word and learns the vector representation by maximizing the co-occurrence probability between words within the window. After the extension, the main purpose of the node2vec algorithm is to obtain the probability of neighboring nodes through the central node. Nodes with the same context represent similarities. In the training process of the model, the gradient is the basis of the training parameter update. The gradient formula can be obtained by constructing the objective function of the training model. Thus, the goal of node2vec optimization is to maximize the probability of occurrence of its neighbor nodes, given the conditions of each central node

$$\max \sum_{u \in V} \log P(Ns(u)|f(u)). \tag{2}$$

The conditions of (2) are based on two assumptions.

1) *Conditional Independence Assumption:* Given a central node, if the probability of its neighbors' existence is independent of other nodes in the set of neighbor nodes, the probability of the neighbors' existence is equal to the product of the probability of occurrence of all its independent neighbor nodes, as defined in the following equation:

$$P(Ns(u)|f(u)) = \prod_{v_i \in Ns(u)} P(v_i|f(u)). \tag{3}$$

2) *Feature Space Symmetry Assumption:* Space symmetry means that a node is the same as the neighbor node if it is the central node, which is different from the second-order similarity in the LINE model. Therefore, under this assumption, the above probability formula can be rewritten as

$$P(v_i \mid f(u)) = \prod_{v_i \in Ns(u)} \frac{\exp f(v_i) \cdot f(u)}{\sum_{v \in V} \exp f(v) \cdot f(u)}. \tag{4}$$

The negative sampling method is used for optimization. All the above statements lead to the idea that the network node represents learning. It has to be noted that the probability of occurrence between nodes is maximized in a fixed sliding window. Although the representation of the node vector can be obtained through (1)–(3), random walking is still required. The way in node2vec to obtain the neighbor sequence of the central node is different from that in the DeepWalk model: it uses a flexible randomization strategy to ensure a balance between the depth-first and breadth-first search strategies.

Fig. 1. Example of the social network graph structure search strategy. (a) BFS. u and s_1, s_2, s_3, and s_4 are in the same community. s_6 and s_5, s_7, s_8, and s_9 also are in another community. u and are central nodes in the two communities respectively. (b) DFS. Performing a depth-first search with node u as the center node can obtain the sequence u, s_4, s_5, s_6, and s_8. The small communities formed by u and s_6 as the center in the graph (a) have edge connections, indicating that the two are similar in some structural properties.

Fig. 2. Example of the social network graph structure search strategy. (a) Search strategy in Node2vec. (b) Search strategy in our algorithm.

B. Strategy-Enhanced Node2vec

1) Breadth-First Search Strategy (BFS): Traversing graphs is commonly used in breadth-first and depth-first search strategies. The DeepWalk model utilizes a random walking strategy where the neighbors are randomly selected for a given central node and then form a sequence of walks. The node2vec model improves the random walk strategy into a biased random walk strategy that can achieve a delicate balance between the depth-first and breadth-first strategies [35].

In a simple social network, as shown in Fig. 1(a), the nodes u and s_1, s_2, s_3, and s_4 are direct neighbors. The nodes s_6, s_5, s_7, s_8, and s_9 are also direct neighbors. This is called homogeneity, and we can call it the center. Nodes and their immediate neighbors are similar in nature, which is a characteristic of the BFS.

2) Depth-First Search Strategy (DFS): As shown in Fig. 1(b), a connected edge between two communities means that their structural properties are similar. The DFS is written into the walk sequence according to indirect neighbor nodes with a similar structure. The BFS focuses on finding the immediate neighbors around the central node. Compared to the breadth-first strategy, the depth-first can jump out of the breadth-first loop, find similar structural nodes more quickly, and achieve similar structures. Furthermore, the DFS is more suitable for traversing large data sets.

3) Random Walk Search Strategy Proposed by Node2vec: Combining two extreme search strategies, the DeepWalk model proposes a random walk search strategy, that is, given a central node, its neighbors are randomly selected to form a random walk sequence. After all the central nodes are traversed, results in a complete corpus for nodes are ready for learning vector representation. The node2vec model utilizes a biased random walk strategy, which combines the advantages of the depth-first and the breadth-first. Fig. 2 depicts an example of the biased random walk strategy. That is, given a center node u, p is the probability of returning, which determines whether or not to go back, and q is a parameter that reflects the feature of wander with DFS or BFS. Assuming that a random walk sequence is generated from node t, and node u is reached, the rules for traversing the next node are illustrated as follows.

If a BFS is adopted, the next step is to select the direct neighbor node of node t, node x_1. If a DFS is used, the next step is to select the indirect neighbor nodes of node t, x_2 or x_3, because the two are separated from the node by one step. There is also a case where it is returned to node t.

In the node2vec model, a second-order transition probability algorithm is used to record the transition probability between two nodes

$$\pi_{ux} = \alpha_{p,q}(t, x) \cdot w_{ux} \tag{5}$$

where w_{ux} represents the weight between the two nodes, which is denoted as $w_{ux} = 1$ in the undirected graph, and α is defined as

$$\alpha_{p,q}(t, x) = \begin{cases} \dfrac{1}{p}, & \text{if } d_{tx} = 0 \\ 1, & \text{if } d_{tx} = 1 \\ \dfrac{1}{q}, & \text{if } d_{tx} = 2 \end{cases} \tag{6}$$

where u indicates the current node, t is the previous node of u, x is the next node of u, and d_{tx} indicates the distance between t and x. It can be seen that, given a central node, the next node is determined, depending on the relationship between the previous step and the next step.

When $d = 0$, it is returned from node u to t. At this time, $a = 1/p$, meaning that the probability of $1/p$ has u returns to the previous node.

When $d = 1$, x is a direct neighbor node of t, which is equivalent to the result of the breadth-first search and here at $\alpha = 1$.

When $d = 2$, x and t are not directly connected, and the shortest distance between them is 2, which is equivalent to the result of the depth-first search. At this time, $\alpha = 1/q$, which means that there is a probability of $1/q$ at which the neighbor nodes of its neighbors are selected.

4) Strategy-Enhanced Node2vec:

a) NRL that integrates node influence: NRL means network representation learning. First, the nodes are added to a sequence, indicating that the nodes in the sequence are similar in some sense. For example, the BFS writes homogenous direct neighbor nodes into the walk sequence. The idea is to write a structurally similar indirect neighbor node into the walk sequence. The written sequence is used for subsequent learning of the vector representation of the nodes, in order to fully retain the network information, such as the network topology information and the influence attribute information of the nodes themselves. The importance of the node in the community can be measured by the attribute of influence. Therefore, the value of the influence of the node has a greater impact on the learning effect, and the value of the node influence is normalized here. There are many factors pertaining to node influence. This article quantifies node influence from the indices of degree centrality and agglomeration coefficient.

Degree Centrality: Degree is defined as the number of neighbor nodes of a node in a social network. The greater the degree of a node, the higher the degree of centrality of the node, or in other words, the greater the influence of the node in the network. In an undirected graph, the degree of centrality represents the sum of the direct connection of a node to all other nodes. Specifically, for an undirected graph with n nodes, the degree of centrality of the node v_i is defined as follows:

$$C_{\text{deg}} = \sum_{i=1}^{n} e_{v_i v_j} \quad (v_i \neq v_j) \tag{7}$$

where C_{deg} indicates the number of nodes directly connected to node v_i. The measure of node degree centrality not only reflects the relationship between each node and other nodes in the network but also depends on the complexity of the network. In other words, the larger the network, the greater the possible value and centrality. In order to eliminate the influence of the scale of the complex network on the centrality, the above formula is normalized as follows:

$$C'_{\text{deg}} = \frac{C_{\text{deg}}}{n-1}. \tag{8}$$

In the above formula, the ratio of the number of nodes directly connected to node i is obtained by dividing the degree centrality value of the node by the maximum possible number $n-1$ connected to other nodes, and the ratio range is 0.0–1.0, where 0.0 means that there is no direct contact with any other node in the network. The point can be considered as a lone point, and 1.0 means that the point is directly connected to every other node in the network. It can be observed that the closer the degree of centrality value after normalization is to 1.0, the greater the influence of the point in the network.

Agglomeration Coefficient: In social networks, the ternary closure principle is that, in a complex social network, if two unconnected nodes have a common neighbor node, the probability of connecting between the two nodes in the future will increase. As reflected in a specific social network diagram, the aggregation coefficient of a node is expressed as the probability that there are connected edges between any two adjacent nodes of the node, that is, the more the ternary closures near a node, the larger its aggregation coefficient. Specifically, the agglomeration coefficient of a node v in an undirected graph is expressed as

$$CC_v = \frac{n}{C_k^2} = \frac{2n}{k(k-1)} \tag{9}$$

where k is the number of all neighbor nodes of node v. n represents the number of interconnections between all adjacent nodes of node v. It can be obtained from the above formula that, for node v to have the number of neighbors k, the more the edges k between the neighbor nodes, the larger the aggregation coefficient.

Define the node influence value according to the two measures of the node influence value

$$f_v = \beta C_{\text{deg}} + \gamma CC_v \tag{10}$$

where β and γ are constants, determining the weight of the two factors above. In the experiment, we usually set it to 1/2. To ensure that the node influence is determined by only two indicators, those of degree centrality and aggregation coefficient, in this article, they should satisfy $\beta + \gamma = 1$, and f_v is the mean value of the influence of the node v. In other words, the greater the influence of the node, the closer f_v is to 1.

In order to be able to combine the two at the same time, the algorithm proposed in this article tells the model that network information and node attribute information to keep by defining a clear objective function

$$L(s) = \frac{1}{|s|} \sum_{i=1}^{|s|} \sum_{i-l \leq j \leq i+l} \log P\left(v_j | v_i\right) \tag{11}$$

where s denotes the length of the sequence. l denotes the size of the slide windows. According to the above objective function, we find that the idea is to maximize the objective function, that is, given a central node, the probability of context occurrence is maximized. Among them, by improving the random walk strategy, we redefine the probability function

$$P\left(v_j \mid v_i\right) = \begin{cases} \dfrac{\alpha\left(v_i, v_j\right)}{Z}, & \text{if } \left(v_i, v_j\right) \in E \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where the node $v_j \in N(v_i)$. $N(v_i)$ is the set of all neighbor nodes of the node v_i. $\alpha(v_i, v_j)$ is the unnormalized transition probability between nodes v_i and v_j, Z is the normalization constant, and the normalized probability $P(v_j \mid v_i)$ ranges from 0.0 to 1.0. On the basis of retaining the network topology information, the influence information of the nodes themselves is merged. The influence value f of each node is introduced, and α is redefined

$$\alpha_{p,q}\left(v_j, v_i\right) = \begin{cases} \dfrac{f}{p}, & \text{if } d_{v_i v_j} = 0 \\ f, & \text{if } d_{v_i v_j} = 1 \\ \dfrac{f}{q}, & \text{if } d_{v_i v_j} = 2 \end{cases} \tag{13}$$

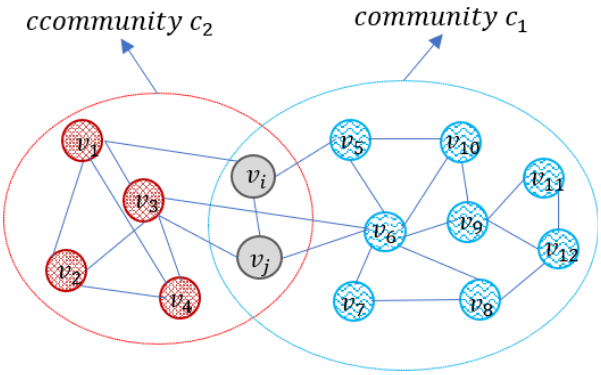


Fig. 3. Community detection principle diagram.

where f is the influence information of the node v_i. As shown in the network graph in Fig. 2, starting from the previous node t, it now comes to node u. The next step to traverse is as follows:

When $d = 1$, there are no changes for the directly connected node. There is still a probability that the value is 1 to select the node directly connected to it.

When $d = 0$, it indicates that there is a case where the probability of f/p is returned to t by the node u, where f represents the influence value of each node. Because the influence of each node is different, the probability of returning to the node t by the node u is different for each node.

Similarly, when $d = 2$, it means that there is a probability that f/p has a node that is not directly connected to it. Because the influence value of each node is different, the probability of selecting nodes is also different, which can improve the representation quality of nodes.

Since the calculation cost of the normalization factor $Zu = \sum_{v_i \in Ns(u)} \exp(f(v_i) \cdot f(u))$ is relatively high, the negative sampling method is used for optimization. When the objective function converges, a vector representation of each node is obtained, where v_j is the neighbor node of v_i, and the probability $\Pr(v_j \mid v_i)$ is defined by the softmax function

$$\Pr(v_j \mid v_i) = \frac{\exp\left(v'_j \cdot v_i\right)}{\sum_{v \in V} \exp(v' \cdot v_i)}. \tag{14}$$

After the above process of training, we can obtain the embedding of node v_f, which retains both the network topology information and the node's influence attribute information.

5) Community Embedding: Given a social network graph $G = (V, E)$, use V and E to represent the set of nodes and edges in the network, respectively, with $E \subseteq (V \times V)$. A community is composed of a series of closely connected nodes, while the connections between two different communities are relatively sparse. As shown in Fig. 3, the goal of community detection is to detect these more closely connected community structures in a complex social network. In other words, the purpose of community detection is to divide the nodes in the network graph $G = (V, E)$ into K subcommunities $C = \{c_1, \ldots, c_i, \ldots, c_k\}$ according to certain rules, taking into consideration the degree of similarity of the network characteristic structure and the attribute information of the nodes.

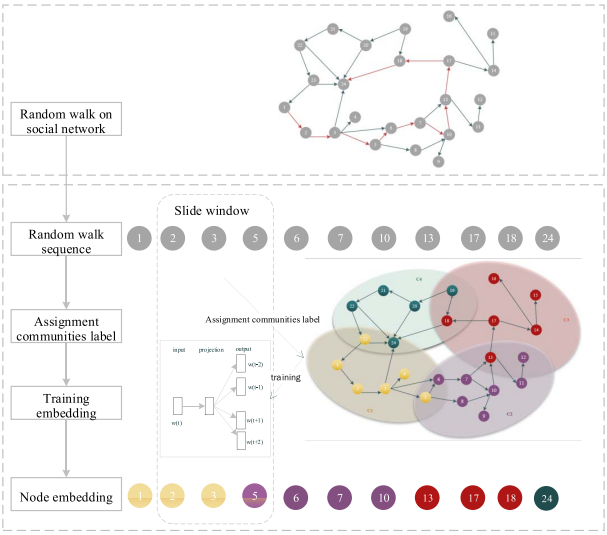
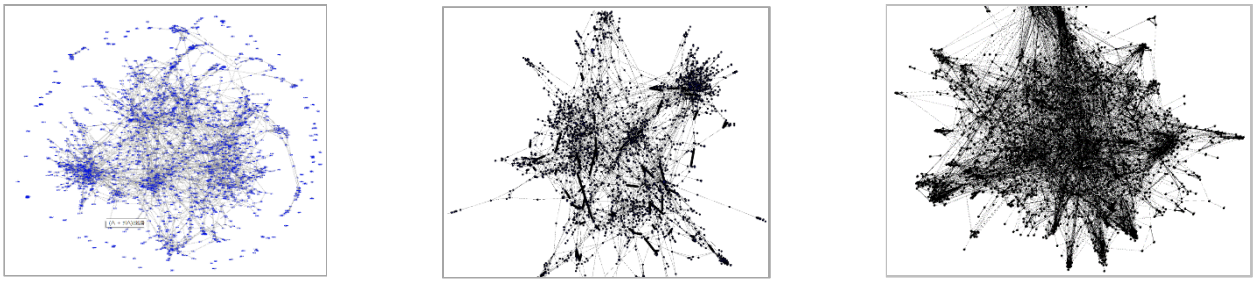


Fig. 4. Framework of training that integrates node and community information.

The community information is embodied in the network diagram as the vector feature between the nodes. For example, the blue line represents the community c_1, and the red line represents the community c_2. When the model reaches convergence, it can detect that the community c_2 contains nodes v_1, v_2, v_3, v_4, v_i, and v_j. Community c_1 contains nodes $v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_i$, and v_j. At the same time, node v_i and v_j can be identified as belonging to community c_1 and c_2, that is, overlapping communities between communities can be detected. In this article, each node is represented by a vector consisting of 256 features. We can find that the observed two nodes with strong connection strength have similar vector representations. For example, they have a lot of common neighbors and so on. Thus, the two vectors with strong similarities are more likely to exist in the same community. However, the community can also be seen as an abstract supernode, also represented by a vector form similar to the node vector. In addition, nodes with common neighbors are more likely to be similar, which is determined by the shared neighborhood between nodes and also reflects the global structure of the network graph. Therefore, in order to effectively consider the attribute information of the community, this article proposes a new network node representation learning method that integrates community information. The method is based on the node vector of the network topology information learned from the random walk sequence corpus. It can also adjust the community attribute information and obtain the node vector that considers both the network topology and the community attribute information tendency, as shown in Fig. 4.

In a network diagram, a node belongs to multiple communities that overlap. Our main goal is to determine which community the nodes in a sequence belong to, inspired by the language model, in a specific sequence. Which community a node belongs to depends on the distribution of the nodes and the distribution of the communities in which the nodes are located. Therefore, each node and the community to which the node is assigned are used to predict the node content in the sequence. Also, in the process of learning representation,



(a) (b) (c)

Fig. 5. Network diagram of three data sets. (a) Cora. (b) Wiki. (c) BlogCatalog.

the community distribution of nodes is also iteratively updated. In the process of vector representation of learning nodes, in order to fully consider the community attribute information, two assumptions are made.

Assumption 1: Each node in a network can belong to multiple communities with different probabilities.

Assumption 2: Given a central node v_i, v_i is known to belong to multiple communities $C = \{c_1, \ldots, c_i, \ldots, c_k\}$. This article states that, in a particular sequence s_i, v_i belongs to only one c_i. Furthermore, on the condition that the known central node v_i belongs to the community c_i, whether the neighbor node v_j of the central node v_i belongs to the community is unknown, which is the key target that we want to obtain $P(v_j \mid c_i)$.

Therefore, our objective function is defined as

$$O_2 = \frac{1}{|s|} \sum_{i=1}^{|s|} \sum_{i-l \le j \le i+l} \log P\left(v_j | c_i\right). \tag{15}$$

In formula (15), c_i denotes the community to which the central node v_i belongs, v_j is the neighbor node of the central node v_i, and $P(v_j \mid c_i)$ denotes the probability that the neighbor node v_j of v_i belongs to the community, under the condition that the known central node v_i belongs to the community c_i. As was done in calculating $P(v_j \mid v_i)$, use the softmax function to calculate $P(v_j \mid c_i)$ as follows:

$$P\left(v_j \mid c_i\right) = \frac{\exp\left(v_j \cdot c_i\right)}{\sum_{v \in V} \exp(v \cdot c_i)}. \tag{16}$$

In the actual optimization phase, we use a negative sampling strategy to optimize the probability $P(v_j \mid c_i)$. After the above representation learning, we can obtain both the vector representation of the node and the vector representation of the community. Once the learning process is completed, the node vector representation v_c containing the community information can be obtained. Possessing both global community information and local structure information, we can obtain a new node vector v' that combines the two pieces of information

$$v' = v_f \oplus v_c. \tag{17}$$

At the same time, after learning all the processes, the community distribution of each node is obvious and can be displayed in a visual form to achieve the purpose of community discovery. We can see that the model is better at finding overlapping communities because the premise is that each node in the network can belong to multiple communities.

C. Overview of the Proposed Framework

The specific pseudocode code is shown in Algorithm 1.

Algorithm 1 Training Process of This Model

Input: graph $G(V, E)$
  community size K
  embedding size d
  window size t
  sequence length L
Output: vertex embedding $\boldsymbol{v}$
  community embedding $\boldsymbol{c}$
$S = InfluenceSamplePath(G)$
Initialize $\boldsymbol{v}$ and $\boldsymbol{c}$
assign a community for each vertex in S randomly
for $iter = 1 : L$ do
  for each vertex v_i in each sequence $s \in S$ do
    calculate statistic-based $P(v|c)$
    assign a community c_i for v_i by $P(c|v, s)$
    for each $j \in [i - t : i + t]$ do
      $L(\mathrm{s}) = \frac{1}{|s|} \sum (\log P(v_j|v_i) + \log P(v_j|c_i))$
      $\boldsymbol{v}_i = \boldsymbol{v}_i - \alpha * \frac{\partial L}{\partial \boldsymbol{v}_i}$
      $\boldsymbol{c}_j = \boldsymbol{c}_j - \alpha * \frac{\partial L}{\partial \boldsymbol{c}_j}$
    end for
  end for
end for
repeat steps 10-21 until convergence

The function $InfluenceSamplePath()$ is to construct a random walk corpus, which integrates the influence information of nodes. We assign a community for each node randomly. We updated the parameters by calculating backpropagation until the objective function converges.

IV. Results and Discussion

In the experiments, we evaluated the effectiveness of the proposed model through two tasks: link prediction and community discovery. All tasks were performed on four real-world data sets, and typical algorithms were selected for comparison. We analyzed the experimental results and drew some conclusions.

A. Data Sets

The experiments were conducted on three widely adopted network data sets: Cora, Wiki, and BlogCatalog. We also

TABLE I
INFORMATION OF DATA SETS

Database	Nodes	Edges	Labels
Cora	2,708	5,429	7
Wiki	2,405	17,981	19
BlogCatelog	10,312	333,983	39

conducted community detection on a social network named Karate [1], [32]. The detailed information of the data sets for real-world social networks is shown in Table I.

Cora: The Cora data set consists of machine learning papers. The network contains 2708 nodes and 5429 edges, which can form seven classifications.

Wiki: Wikipedia is an online collaboration encyclopedia written by its users and is widely used in social network analysis. The data set consists of 2405 web pages and 17 981 articles, which are divided into 19 categories. It is obvious that the social relationship in this data set is relatively dense.

BlogCatelog: The BlogCatalog data set is taken from the BlogCatalog management blog and the blogger's social blog directory, which forms a social friendship network between the blogs and the bloggers, including 10 312 nodes and 333 983 edges. The topics published by the bloggers are presented as the label values, forming 39 categories.

B. Baseline Methods (Evaluation Standard)

We used four classic models as baseline methods: LINE, node2vec, DeepWalk, and GraRep. Their working procedures are described as follows.

LINE [33]: First, the first-order similarity and the empirical probability, as well as the second-order similarity and the empirical probability of the nodes in the network, are minimized, and their KL divergences are the primary target of the neural network training. The output is then merged as a low-dimensional vector representation of the node. The method retains the first- and the second-order neighbor information of the node.

Node2vec [34]: First, the probability transfer matrix is initialized according to the parameters p and q. Next, several random walks are used for each node (the probability is not equal, and the next neighbor is selected to use the alias sampling method) to obtain the local sequence information with a fixed length. The SkipGram method is used to train the neural network to obtain the vector representation of the nodes while retaining the local and global characteristics of the nodes. In other words, it includes structural similarity and content similarity.

DeepWalk [35]: The DeepWalk algorithm is the first application of SkipGram technology in the field of natural language in social networks. Random walk sequences are generated by the random walk sampling method for all nodes. In order to obtain the characteristics of the node, a large corpus is formed by learning and using the SkipGram model.

GraRep [36]: GraRep solves the problem of network embedding by matrix decomposition. GraRep can deal with weighted networks. However, due to a large amount of computation, this method is particularly time-consuming.

C. Link Prediction

The task of link prediction is to predict possible missing edges based on the existing network structure. The structural similarity of the nodes in the network, the influence attribute information of the nodes, and the community attribute information of the nodes can be used to predict the two nodes and the potential relationship between them. Therefore, the algorithm proposed in this article takes into account both the topology of the network and the influence attributes of the nodes, and the attribute information of the community they belong to.

The link prediction algorithm uses Micro-F1 and Macro-F1 as our evaluation metrics, which can measure the accuracy of the algorithm.

For the three data set networks, some of the connected edges are randomly removed. However, the overall connectivity of the network is unchanged after the connected edges are removed. Use the remaining network structure after culling to train the model and learn to obtain 128-D features of all nodes in the network. Calculate the similarity distance between any two nodes, and use the evaluation index to measure and analyze the comparative experimental results. Moreover, when experiments are performed on each data set, regardless of whether there are connected edges between nodes, in reality, the distances between all nodes excluding the training set are calculated, and the predicted structure and the true condition of edges are used to calculate the value of Micro-F1 and Macro-F1.

The results on different data are shown in Figs. 6 and 7. From Figs. 6 and 7, we can draw the conclusions: the algorithm proposed in this article achieves better results than the other four algorithms in general, which means that it can retain the network topology information and the influence attribute information of the node.

D. Overlapping Community Detection

In the community detection experiments, the Edge Betweenness and Community Structure (EC), Fast-greedy (FC), and LC algorithms were carried out in four network data sets to verify our algorithm's effectiveness in community detection. More analyses were carried out to ascertain the pros and cons of the four algorithms.

With the EC algorithm, researchers paid particular attention to some attributes that many networks seem to share, such as small-world transitivity, power-law degree distribution, and network transitivity. This article focuses on another attribute that is detected in many networks: the attribute of community structure, in which network nodes are connected together in tightly woven groups, and there are only loose connections between them. A method for detecting such communities is proposed. This method is based on the idea of using the

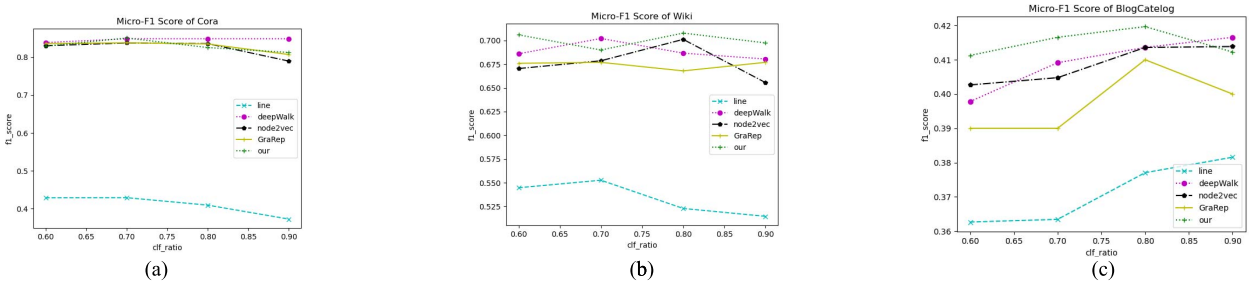


Fig. 6. Micro-F1 of the methods. (a) Cora network. (b) Wiki network. (c) BlogCatelog network.

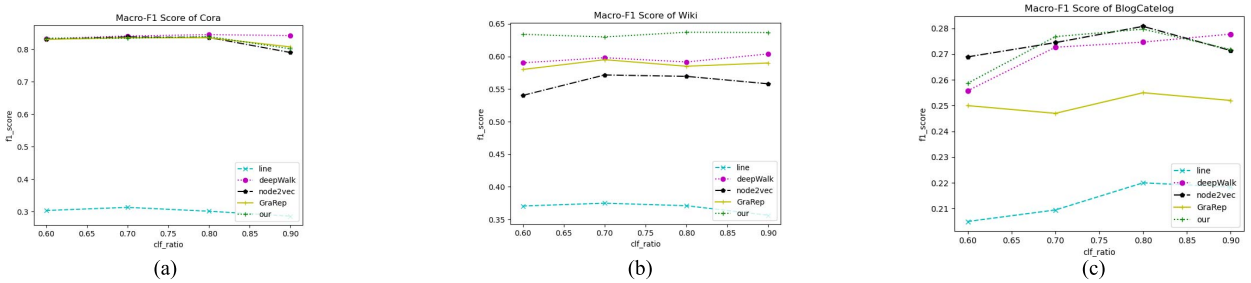


Fig. 7. Macro-F1 of the methods. (a) Cora network. (b) Wiki network. (c) BlogCatelog network.

concentration index to find the community boundary, and it can detect the actual community division.

The FC algorithm is a hierarchical aggregation algorithm for detecting community structures. This algorithm is faster than many competing algorithms and is suitable for large networks. For instance, it can analyze the network of goods for sale on the website of a large online retailer. If the same buyer purchases goods from the website frequently, the goods in the network will be linked together, and the algorithm can extract meaningful communities from the network to reveal a large-scale model that exists in customer buying habits.

The LC algorithm is a simple LC algorithm that considers only the network structure. It does not need to optimize the predefined objective function or the prior information about the community. In this algorithm, each node is initialized with a unique label, and each node uses the label currently owned by most of its neighbors at each step. During this iteration, densely connected node groups are formed on the unique label consensus to form a community.

We used two indexes of modularity and ARI to measure the performance of the algorithm in this article, and a reasonable analysis and explanation were also conducted based on the experimental results.

Modularity is one of the most commonly used indicators in the community detection field for measuring the effectiveness of community division. The main idea of modularity is that: when the similarity of the nodes inside the community is relatively high, and the similarity of the nodes outside the community is relatively low, it is regarded as an ideal community detection result. The calculation of the modularity is as follows:

$$Q = \sum_c \left[\frac{\sum \text{in}}{2m} - \left(\frac{\sum \text{out}}{2m} \right)^2 \right] = \sum_c [e_c - a_c^2]. \quad (18)$$

Among them, $\sum$in represents the sum of the weights of the edges in the community c, $\sum$out represents the sum of the weights of the edges connected to the nodes in the community c, and m represents the sum of the weights of all edges in the community. The weight of all edges in the unweighted network graph can be regarded as 1. Therefore, the greater the value of the modularity, the better the effect of community detection.

The adjusted rand index (ARI) is one of the more commonly used evaluation indicators in clustering algorithms and can also be used in the field of community detection.

The rand index (RI) is calculated as follows:

$$\text{RI} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \quad (19)$$

where the following holds.

True Positive (TP): The points of the same category that are divided into the same community.

True Negative (TN): The different types of nodes that are divided into different communities.

False Positive (FP): The nodes of the different classes that are divided into the same community.

False Negative (FN): The nodes of the same category that are divided into different communities.

ARI shows better differentiation on the basis of RI. The value range of RI is [0, 1], but the value range of ARI is [−1, 1]. The larger the value of ARI, the more the community detection structure matches the real situation. The ARI is calculated as follows:

$$\text{ARI} = \frac{\text{RI} - E[\text{RI}]}{\max(\text{RI}) - E[\text{RI}]}. \quad (20)$$

The results of comparison experiments on the different data sets are as follows.

1) Cora Network: Our algorithm performed better than the others in terms of modularity. For the ARI, the performance of

TABLE II
RESULTS ON THE CORA NETWORK

	EC	FC	LC	Our algorithm
Modularity	0.401298	0.380670	0.371488	**0.418803**
ARI	0.391571	0.568439	**0.882257**	0.583075
Community size	**10**	8	7	7

TABLE III
RESULTS ON THE WIKI NETWORK

	EC	FC	LC	Our algorithm
Modularity	0.599625	0.549740	0.601008	**0.604569**
ARI	0.778102	0.543118	**0.899062**	0.820829
Community size	17	8	19	**19**

TABLE IV
NMI OF BASELINE METHODS ON EXPERIMENTAL NETWORKS

	DeepWalk	Node2vec	LINE	Our algorithm
LFR ($\mu = 0.6$)	0. 962	0. 981	0. 669	0. 975

our algorithm was very different from other algorithms. It is found that the LC algorithm performed the best with the largest value. Our algorithm was close to the FC algorithm. It may be because of the community size divided by the algorithm. When the number of communities is not equal to the number of the real data set labels, the two indicators, modularity and ARI, will affect the final results.

2) Wiki Network: Our algorithm performed significantly better than the other algorithms on two indicators of modularity and community size. As for community size, the LC algorithm and our algorithm both precisely detected the number of communities in the network.

In order to better evaluate the accuracy, we also adopted normalized mutual trust interest (NMI) as an evaluation standard. The closer the value is to 1, the closer the community detection result is to the real result, and the more accurate the algorithm result is.

According to Table IV, the NMI value of the proposed algorithm is similar to those of the other algorithms. It indicates that its effect on community detection is effective.

V. CONCLUSION

In recent years, NRL has been widely adopted in network analysis for several reasons. It performs low-dimensional vectors based on the role of nodes in the network. Based on the low representation of real values, NRL can measure the semantics between nodes. It also reduces the traditional graph-based representation. The conclusions of our study can be drawn as follows.

1) Based on the local information of the nodes, we use a stochastic strategy to improve the node2vec model. We propose a specific global community label training method that can improve the quality of the prediction node representation.
2) In general, the representation of a node is determined by its neighbors and the communities. We propose a community detection method based on NRL that combines node embedding, community embedding, and clustering methods.
3) Our experiments on three prevailed data sets demonstrate the effectiveness of our model in detecting overlapping communities and node representation learning.

We will examine and test the generalization of our method on larger data sets and explore the impact of dynamic communities on community embedding.

REFERENCES

[1] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, 2002.
[2] T. Opsahl, "Triadic closure in two-mode networks: Redefining the global and local clustering coefficients," *Social Netw.*, vol. 35, no. 2, pp. 159–167, May 2013.
[3] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech.: Theory Exp.*, vol. 2008, no. 10, 2008, Art. no. P10008.
[4] I. Kirianovskii, O. Granichin, and A. Proskurnikov, "A new randomized algorithm for community detection in large networks," *IFAC-PapersOnLine*, vol. 49, no. 13, pp. 31–35, 2016.
[5] M. Shi, Y. Zhou, and Y. Xing, "Community detection by label propagation with LeaderRank method," *J. Comput. Appl.*, vol. 35, no. 2, pp. 448–451, 2015.
[6] A. Lakhdari, A. Chorana, H. Cherroun, and A. Rezgui, "A link strength based label propagation algorithm for community detection," in *Proc. IEEE Int. Conf. Big Data Cloud Comput. (BDCloud), Social Comput. Netw. (SocialCom), Sustain. Comput. Commun. (SustainCom) (BDCloud-SocialCom-SustainCom)*, Oct. 2016, pp. 362–369.
[7] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, "Learning with local and global consistency," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 321–328.
[8] X. Liu and T. Murata, "Advanced modularity-specialized label propagation algorithm for detecting communities in networks," *Phys. A, Stat. Mech. Appl.*, vol. 389, no. 7, pp. 1493–1500, 2010.
[9] J. Xie and B. K. Szymanski, "Community detection using a neighborhood strength driven label propagation algorithm," in *Proc. IEEE Netw. Sci. Workshop*, Jun. 2011, pp. 188–195.
[10] J. Xie and B. K. Szymanski, "LabelRank: A stabilized label propagation algorithm for community detection in networks," in *Proc. IEEE 2nd Netw. Sci. Workshop (NSW)*, Apr. 2013, pp. 138–143.
[11] T. Luan, Z. Yan, S. Zhang, and Y. Zheng, "Fraudster detection based on label propagation algorithm," in *Proc. IEEE 20th Int. Conf. High Perform. Comput. Commun.; IEEE 16th Int. Conf. Smart City; IEEE 4th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Jun. 2018, pp. 346–353.
[12] T. Ma, M. Yue, J. Qu, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan, "PSPLPA: Probability and similarity based parallel label propagation algorithm on spark," *Phys. A, Stat. Mech. Appl.*, vol. 503, pp. 366–378, Aug. 2018.
[13] Z. Zhang, Z. Zhang, W. Yang, and X. Wu, "An overlapped community partition algorithm based on line graph," in *Proc. 14th Int. Conf. Web-Age Inf. Manage.*, 2013, pp. 277–281.
[14] S. Huixia and L. I. Yuexin, "Overlapping community discovering algorithm based on latent features," *J. Comput. Appl.*, vol. 35, no. 2, pp. 3477–3480, 2015.

[15] H. Liu, L. Fen, J. Jian, and L. Chen, "Overlapping community discovery algorithm based on hierarchical agglomerative clustering," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 32, no. 03, Mar. 2018, Art. no. 1850008.
[16] M. Huang, G. Zou, B. Zhang, Y. Liu, Y. Gu, and K. Jiang, "Overlapping community detection in heterogeneous social networks via the user model," *Inf. Sci.*, vol. 432, pp. 164–184, Mar. 2018.
[17] C. Pizzuti, "Evolutionary computation for community detection in networks: A review," *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 464–483, Jun. 2018.
[18] T. Mikolov, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[19] D. Liang, J. Altosaar, L. Charlin, and D. M. Blei, "Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 59–66.
[20] C. Van Gysel, M. de Rijke, and E. Kanoulas, "Learning latent vector spaces for product search," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 165–174.
[21] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.
[22] L. Ni, P. ManMan, J. Wenjun, and L. Kenli, "A community detection algorithm based on multi-similarity method," *Cluster Comput.*, vol. 22, no. 2, pp. 2865–2874, 2019.
[23] Z. Liu, B. Xiang, W. Guo, Y. Chen, K. Guo, and J. Zheng, "Overlapping community detection algorithm based on coarsening and local overlapping modularity," *IEEE Access*, vol. 7, pp. 57943–57955, 2019.
[24] Y. Bian, D. Luo, Y. Yan, W. Cheng, W. Wang, and X. Zhang, "Memory-based random walk for multi-query local community detection," *Knowl. Inf. Syst.*, vol. 62, no. 5, pp. 2067–2101, May 2020.
[25] J. Tang, M. Qu, M. Wang, M. Zhang, and J. Yan, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 1067–1077.
[26] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, vol. 2016, pp. 855–864.
[27] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 701–710.
[28] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, "Network embedding as matrix factorization: Unifying deepwalk, LINE, PTE, and node2vec," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, 2018, pp. 459–467.
[29] R. A. Rossi, R. Zhou, and N. K. Ahmed, "Deep inductive network representation learning," in *Proc. Companion The Web Conf. Web Conf. (WWW)*, 2018, pp. 953–960.
[30] K.-H. Lai, C.-M. Chen, M.-F. Tsai, and C.-J. Wang, "NavWalker: Information augmented network embedding," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Dec. 2018, pp. 9–16.
[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Represent.*, 2013.
[32] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropological Res.*, vol. 33, no. 4, pp. 452–473, Dec. 1977.
[33] H. Sun *et al.*, "IncOrder: Incremental density-based community detection in dynamic networks," *Knowl.-Based Syst.*, vol. 72, pp. 1–12, Dec. 2014.
[34] Y. Luo, L. Wang, S. Sun, and C. Xia, "Community detection based on local information and dynamic expansion," *IEEE Access*, vol. 7, pp. 142773–142786, 2019.
[35] M. Cordeiro, R. P. Sarmento, and J. Gama, "Dynamic community detection in evolving networks using locality modularity optimization," *Social Netw. Anal. Mining*, vol. 6, no. 1, pp. 1–20, Dec. 2016.
[36] S. Cao, W. Lu, and Q. Xu, "GraRep: Learning graph representations with global structural information," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 891–900.

Meizi Li is currently an Associate Professor with the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China. Her current research interests include social network analysis, trust, and reputation computation.

Shuyi Lu is currently pursuing the master's degree with the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China.

Her current research interests include link prediction in social networks and community discovery.

Lele Zhang is currently pursuing the master's degree with the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China.

Her current research interest includes social network community discovery.

Yuping Zhang was born in 1963. She received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2004.

She is currently a Professor with Shanghai Normal University. She is also the Director of the Shanghai Graphics Society in China. Her research interests include CAD and optimization algorithms.

Bo Zhang (Member, IEEE) received the Ph.D. degree in computer science from the College of Electronics and Information Engineering, Tongji University, Shanghai, China, in 2009.

He finished his post-doctoral research work at Tongji University in 2012. He is currently a Professor with the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai. He is also the Director of the Social Network Analysis Project, which is funded by the National Nature Science Foundation of China. His current research interests include intelligent information processing, trust computation, and social network analysis.