Hindawi
Complexity
Volume 2019, Article ID 8292485, 16 pages
https://doi.org/10.1155/2019/8292485

Research Article

Neighbor Similarity Based Agglomerative Method for Community Detection in Networks

Jianjun Cheng[1], **Xing Su**[1], **Haijuan Yang**[1,2], **Longjie Li**[1], **Jingming Zhang**[1], **Shiyan Zhao**[1] and **Xiaoyun Chen**[1]

[1] *School of Information Science and Engineering, Lanzhou University, China*
[2] *Department of Electronic Information Engineering, Lanzhou Vocational Technical College, China*

Correspondence should be addressed to Jianjun Cheng; chengjianjun@lzu.edu.cn

Received 27 December 2018; Revised 15 March 2019; Accepted 11 April 2019; Published 2 May 2019

Academic Editor: Guang Li

Community structures can reveal organizations and functional properties of complex networks; hence, detecting communities from networks is of great importance. With the surge of large networks in recent years, the efficiency of community detection is demanded critically. Therefore, many local methods have emerged. In this paper, we propose a node similarity based community detection method, which is also a local one consisted of two phases. In the first phase, we first take out the node with the largest degree from the network to take it as an exemplar of the first community and insert its most similar neighbor node into the community as well. Then, the one with the largest degree in the remainder nodes is selected; if its most similar neighbor has not been classified into any community yet, we create a new community for the selected node and its most similar neighbor. Otherwise, if its most similar neighbor has been classified into a certain community, we insert the selected node into the community to which its most similar neighbor belongs. This procedure is repeated until every node in the network is assigned to a community; at that time, we obtain a series of preliminary communities. However, some of them might be too small or too sparse; edges connecting to outside of them might go beyond the ones inside them. Keeping them as the final ones will lead to a low-quality community structure. Therefore, we merge some of them in an efficient approach in the second phase to improve the quality of the resulting community structure. To testify the performance of our proposed method, extensive experiments are performed on both some artificial networks and some real-world networks. The results show that the proposed method can detect high-quality community structures from networks steadily and efficiently and outperform the comparison algorithms significantly.

1. Introduction

Many real-world systems can be abstracted as complex networks, in which nodes represent entities in the systems, and edges correspond to interactions between the entities. One of the most significant characteristics observed in these complex networks is the "community structure," which means that nodes in the network can be divided into groups naturally; nodes in the same group are connected densely, and connections across different groups are relatively sparse; each of the node groups is a so-called "community."

The communities are always related to functional modules of networks. For instance, communities can be groups of web pages in WWW networks [1] or scientific papers in citation networks [2] sharing same topics, books with the same political orientations copurchased from the online bookseller, Amazon.com [3], pathways or complexes in metabolic networks, or protein-protein interaction networks [4, 5]. In social networks, communities often correspond to real social groupings having the same interests or professional occupations, e.g., scientist groups classified according to the scientists' specialties in the coauthor relationship collaboration networks [6, 7], jazz musician groups divided according to the locations and race [8], or affiliations of gang members in the policing area of Hollenbeck, Los Angeles [9]. Besides this, some researches have indicated that networks can present quite different properties when being considered at the community level, rather than from the perspective of entire network or the individual node [10].

Therefore, analyzing the community structures in networks can facilitate the recognition of the characteristics of networks and make prediction further about the functional properties of the corresponding systems. That is to say, community detection provides us with an effective means for studying the functional properties of networks via dipping into structural characteristics, which really make sense in practical applications. Therefore, a multitude of methods [11, 12] have been proposed for detecting communities in complex networks; we will review some related literature in Section 2.

In this paper, we propose a community detection method as well, which is based on node similarity and consists of two phases. The first phase repeatedly selects the node with the largest degree in the remainder of the network and either takes it as the exemplar of a new community or inserts it into the community to which its most similar neighbor belongs, according to its most similar neighbor's community affiliation. At the end of this phase, we get a series of communities. However, they are only the preliminary communities; some of them might be too small or too sparse; edges connecting to outside of them might go far beyond the ones inside them. Accepting them as the final ones will lead to a low-quality community structure. Therefore, the second phase merges some of the preliminary communities to improve the quality of the resulting community structure.

The main contributions of this work can be summarized as follows.

(i) We propose a node similarity based local algorithm, shortened as *NSA*, for community detection, which is a two-phase method. The first phase is used to get the preliminary communities, and the second phase is to merge some of the preliminary communities to improve the quality of the resulting community structure.

(ii) We propose an index, *community metric*, to measure the sparsity or smallness of a community. In the second phase, we use the index as a criterion to determine which preliminary communities need to be merged.

(iii) Extensive experiments on some artificial networks and real-world networks are carried out to testify the performance of the proposed method. The experimental results show that the performance and the time complexity of the proposed method are steadily promising and outperform its competitors.

The remainder of this paper is organized as follows. Section 2 reviews some literature about community detection. The details of the proposed algorithm are elaborated in Section 3. The experimental results and analysis on both artificial networks and real-world networks are presented in Section 4. In Section 5, we discuss how to set the optimal value for a parameter introduced in our proposed method, and the paper ends with a conclusion in Section 6.

2. Related Work

A great deal of community detection methods have been proposed in the last decade; these methods try to explore communities in networks from various perspectives. The graph theory-based methods take the problem of community detection as the traditional task of graph partitioning and divide the network into subnetworks. Kernighan-Lin [13] is a representative method of this kind, which partitions the network into two arbitrary subnetworks first and then repeatedly swaps some nodes between the two subnetworks to maximize a predefined gain function.

The hierarchical clustering methods reveal multilevel community structures either in divisive ways or in agglomerative approaches or in hybrid ways; e.g., GN algorithm [6, 7] detects communities by repeatedly removing the edge with the largest betweenness from the networks, its output is a dendrogram representing the nested hierarchy of possible community structures of the network, and the level corresponding to the largest value of a measure, modularity[7], is taken as the final result. Fast$Q$ algorithm [23, 24] takes each node in the network as a community first and then repeatedly merges two of them into one. Its output is also a dendrogram depicting the merge procedure of possible community hierarchies. Zarandi et al. [25] randomly removed some edges with low similarity to obtain some disconnected components as the primary communities, and then some of them are merged to get the resulting community structure.

The modularity optimization-based algorithms detect community structures from networks by utilizing the physical meaning of modularity—the higher the value of modularity, the better the community structure—and taking the modularity as the objective to optimize. For instance, in order to maximize the modularity of the community structure, Fast$Q$[23, 24] joins a pair of communities whose merge can lead to the largest modularity increment in each iteration. Louvain algorithm [26] uses the node-moving strategy to extract community structure with the optimized modularity from the network, which begins with an initial partition of each node being a community as well; then for each node, the algorithm evaluates the modularity gain of moving it into the community to which each of its neighbors belongs and moves that node into the community with the largest positive modularity gain consequently. SLM (short for Smart Local Moving) algorithm [27] searches for possibilities of increasing modularity with respect to both splitting communities and moving sets of nodes from one community to another.

LPA (Label Propagation Algorithm) [28] makes utilization of information propagation mechanism to detect communities from networks. Every node in the network is initialized with a unique label and all nodes in the network are arranged in a random order first; then each node in that specific order updates its label to the one occurred most frequently among its neighbors. This label update procedure is ended with the status that every node in the network has a label which is the majority one among neighbors, and nodes with the same labels form a community. Owing to its simplicity and high efficiency, several variants have

been derived from LPA. Barber et al. [29] proposed a series of algorithms that propagate labels under some constraints; LPAm is the most famous one, which tries to maximize the modularity during the label propagation procedure. Chin et al. [30] identified the main communities using the number of mutual neighboring nodes first; then they attached some independent constraints to the basic LPA and used the constrained LPA to add the remainder nodes into communities; finally, they used a node-moving strategy like that is employed in Louvain to refine the quality of the resulting community structure. Ding et al. [31] yielded a modified version of LPA, which exploits the idea of density peak clustering [32] and Chebyshev inequality to choose community centers from the network, and then propagates labels of the selected centers to the whole network with the proposed multistrategy of label propagation.

Density-based methods define and utilize the concept of *density* in networks for nodes or communities to uncover community structures. SCAN [33] borrows the idea from the classical density-based clustering algorithm, DBSCAN [34], to reveal communities, hubs, and outliers from networks. SCAN++ [35] is a derivative of SCAN; it reduces time consumption via introducing a new data structure and reducing the number of density evaluations in the detecting procedure. IsoFdp [36] maps the network nodes as data points into a low-dimensional manifold and then exploits the density peak clustering algorithm [32] to extract the final community structure. LCCD algorithm [37] also practices on the way proposed in the density peak clustering algorithm [32] to locate the structural centers from networks and then expands communities from the identified centers to the borders using a local search procedure.

Network dynamic-based methods explore community structures by simulating the dynamic processes in networks. Random walk is a typical dynamic procedure carried out in networks; random walk-based methods utilize the tendency of the walker being trapped into a community during a short walk, rather than walking across the community border into another community, to detect communities from networks. WalkTrap [38] makes use of random walk to calculate the probability of going from one node to another during a short-length walk and then calculates the distance to measure nodes' similarities and community similarities. PPC algorithm [39] considers the network as a single community initially and recursively partitions each community utilizing node similarities computed using random walks until further partitioning cannot acquire a better value of modularity. RWA [40] employs random walks to calculate the probability of a node belonging to a community, and each community is expanded by repeatedly attracting the node which is most likely to belong to that community to join. Besides this, Attractor [41] utilizes distance dynamics to explore communities from networks, node interactions might change the distances among nodes, and the distance change will make an impact on the interaction in reverse. Members of the same community will gradually move together under such interplays, and nodes in different communities will keep far away from each other steadily. BiAttractor [42] extends the concept of distance dynamics and the idea of Attractor to bipartite networks, which is used to detect two-mode communities of bipartite networks.

Spectral methods engage eigenspectra of various network-associated matrices to extract communities. For example, Amini et al. [43] found the initial node partitions using the spectral clustering method based on the normalized Laplacian matrix derived from a regularized adjacency matrix; those partitions were used for fitting a stochastic block model by a pseudolikelihood algorithm to detect the resulting community structure. Siemon C. de Lange et al. [44] identified an integrative community structure in the macroscopic anatomical neural networks of the macaque and cat and the microscopic network of the *C. elegans* by examining the spectra of their normalized Laplacian matrices. Krzakala et al. [45] produced a class of spectral algorithms to detect communities based on the nonbacktracking matrix, which depicts a nonbacktracking walk on the directed edges of the network. Shi et al. [46] proposed a spectral community detection method, LLSA, which employs Lanczos method to obtain the approximated eigenvector of the transition matrix with the largest eigenvalue, and the elements of this eigenvector approximately indicate the affiliation probability of the corresponding nodes to the communities.

Most of the methods mentioned above are global ones; they detect communities often depending on some global information, such as the number of communities, information about eigenvalues or eigenvectors, as prior knowledge, but they are hard to acquire due to the size of networks involved getting larger and larger. Moreover, most of them are computationally demanding, leading to high time complexity. These limitations prevent them from being applied to large-scale applications. To overcome the deficiency of the global algorithms, many local methods have been proposed, including some of the aforementioned methods. For example, LPA and most of its variations determine which label should be adopted by a node according to its neighborhood only; LCCD takes into account both the local density of nodes and the relative distance between nodes to locate the local structural centers and expands communities from the structural centers with a local search procedure; LLSA applies a fast heat kernel diffusing to sample a small subnetwork including almost all members of a community, and the eigenvector whose elements suggest nodes for their memberships of communities is obtained by performing Lanczos method on the sampled subnetwork.

Besides this, ComSim algorithm [47] identifies cores of communities from bipartite networks by seeking for cycles which are node chains formed by following outgoing links and reaching a node already visited and then allocates the remaining nodes to the communities that maximize the similarity between the node and the community. In BLI algorithm [48], local clustering information and local structural similarity are employed to establish the primary community structure; then some small-scale communities whose sizes are smaller than a given threshold, λ, are absorbed by some larger ones. *k*SIM [49] is also a local method that works in a bottom-up way. At the beginning, each node is taken as a community; then the preliminary communities are formed by identifying for each node the neighbor community to

Input: $G(V, E)$, the network; δ, the community metric threshold
Output: CS, the detected community structure
/* form the preliminary community structure CS_pre */
1 $CS\_pre \longleftarrow$ **FPC**(G)
/* merge small or sparse communities in CS_pre */
2 $CS \longleftarrow$ **PCM**(CS_pre, δ)
3 **return** CS

Algorithm 1: The framework of our proposed method, NSA.

which one of its k most similar neighbors with the lowest degree belongs and assigning the node to that community. In this procedure, common neighbor index is employed as the similarity measure for each pair of nodes.

Compared to those global ones, these local methods show good performance in large-scale networks. Inspired by this, we also propose a local method to extract communities from networks. The proposed method is based on node similarity and is termed as *NSA* (Node Similarity based Algorithm) for short; it comprises of two phases: the first phase aims at constructing the preliminary community structure; the second phase tries to improve the quality of the final result by merging some small or sparse communities. To do so, we also propose a measure, *community metric*, to evaluate the sparsity or smallness of communities. The details of the proposed method are elaborated in the next section.

3. The Proposed Method

3.1. The Framework of the Proposed Method. The framework of the proposed method is outlined by the pseudocode listed in Algorithm 1.

As mentioned previously, the proposed method consists of two phases. Function calls **FPC()** and **PCM()** implement the two phases, respectively. The former establishes the preliminary community structure based on a node selection strategy and the node similarity; the latter merges some small or sparse communities to improve the quality of the resulting community structure. The inputs of this algorithm are the network and a threshold δ; the network involved in this paper is the undirected and unweighted graph, which is always represented as $G(V, E)$ as in Algorithm 1, where V and E are the node set and edge set, respectively; $|V| = n$ and $|E| = m$ are the number of nodes and edges in the network, individually. The threshold δ is used in the second phase of the proposed method to identify communities to be merged—a community whose community metric is smaller than δ should be merged into another one. The output of this algorithm is the detected community structure.

The next two subsections describe the two procedures concretely and deliberately.

3.2. Formation of the Preliminary Community Structure. The function **FPC()** implements the first phase of the proposed method, whose purpose is to construct the preliminary community structure from the network. We first pick out the node with the largest degree from the network, take it as the exemplar of the first community, and insert its most similar neighbor into the community as well (if there are more than one node with the largest degree in the network, we arbitrarily select any one of them to take it as the exemplar; and if the exemplar has more than one most similar neighbors, the one with the smallest degree is selected). Afterwards, the next largest-degree node in the remainder of network is selected; if its most similar neighbor has not been classified into any community yet, we create a new community for it and its most similar neighbor. Otherwise, if its most similar neighbor has been assigned to a certain community (e.g., the one denoted as C_k), we insert the selected node into that community (i.e., C_k) as well. We repeat this process until every node is classified into a community. In this procedure, densely connected nodes can quickly gather together around the exemplars to form communities. At the end of this procedure, we get a series of communities, which constitute the preliminary community structure of the network. The pseudocode describing the entire procedure is listed in Algorithm 2.

In this algorithm, the degree of node u is the number of u's neighbors and is denoted as d_u, i.e.,

$$d_u = |\Gamma(u)|, \tag{1}$$

where

$$\Gamma(u) = \{v \mid (u, v) \in E, v \in V\} \tag{2}$$

is the set of neighbors of node u. $sim(u, v)$ stands for the similarity between nodes u and v. There are abundant ways to calculate the similarity between nodes in the network; any one of them can be employed in principle. However, to pursue the efficiency, we calculate it here as in the following equation, which involves only the neighborhoods of nodes u and v themselves.

$$sim(u, v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}. \tag{3}$$

The variables U and CS_pre are used to record the unclassified nodes and the preliminary community structure; they are naturally initialized to be the original node set V of network G and an empty set ϕ in step 1. Steps 2 and 3 select the node with the largest degree from the remainder of the network and its most similar neighbors and denote them as v and w, respectively. Step 4 determines whether w has been assigned

Input: $G(V, E)$, the network
Output: $CS\_pre = \{C_1, C_2, \cdots, C_k\}$, the identified preliminary community structure
1 Initialize variables U and CS_pre, which are used to record the unclassified nodes and the preliminary community structure:
$U \longleftarrow V$; $CS\_pre \longleftarrow \phi$;
2 Select the node with the largest degree, denote it as v:
$v \longleftarrow \arg\max_u \{d_u \mid u \in U\}$
3 Get the most similar neighbor of v, denote it as w:
$w \longleftarrow \arg\max_u \{sim(v, u) \mid u \in \Gamma(v)\}$
4 **if** w has not been assigned to any community **then**
5 Create a new community for nodes v and w:
$K \longleftarrow |CS\_pre|$; $C_{K+1} \longleftarrow \{v, w\}$;
6 Insert the created community into the community structure:
$CS\_pre \longleftarrow CS\_pre \cup \{C_{K+1}\}$
7 Remove nodes v and w from U as they are classified:
$U \longleftarrow U - \{v, w\}$
8 **else**
9 Find the community to which w belongs, denote it as C_k:
$k \longleftarrow$ locate(CS_pre, w)
10 Insert node v into C_k:
$C_k \longleftarrow C_k \cup \{v\}$
11 Remove node v from U as it is classified:
$U \longleftarrow U - \{v\}$
12 Repeat steps 2 through 11, until $U = \phi$
13 **return** CS_pre

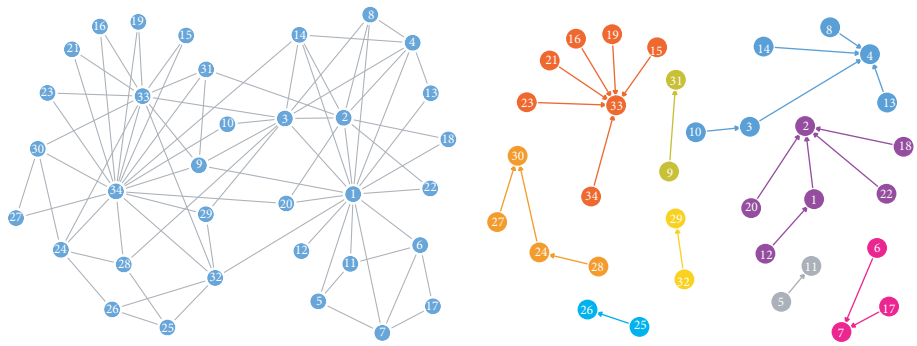
Algorithm 2: **FPC(G)**: forming the preliminary community structure.

to a community or not; if it has not been classified to any community yet, steps 5 and 6 create a new community for nodes v and w and insert the newly created community into CS_pre; then step 7 removes nodes v and w from U as they have been classified into the new community just now. If node w has been already assigned to a community, step 9 finds the community C_k, to which node v's most similar neighbor w belongs, and step 10 inserts node v into community C_k. Since node v has been assigned to community C_k, step 11 removes it from U. Step 12 repeats operations in steps 2 through 11, until $U = \phi$, meaning that all the nodes in the network have been visited. At that time, the preliminary community structure is obtained in CS_pre and is returned as the output of this algorithm in step 13.

To make it clearer, we take Zachary's karate club network [14] as an example to illustrate intuitively the procedure. This is a network with 34 nodes and 78 edges as shown in Figure 1(a), in which the node with the largest degree is node '34', and its most similar neighbor is node '33'. Therefore, node '34' is taken as the exemplar of the first community, and node '33' is also inserted into this community. Then, the node with the largest degree in the remaining nodes is node '1'; its most similar neighbor is node '2'. Since node '2' has not been assigned to a community yet, we create a new community, take node '1' as its exemplar, and insert node '2' into the new community as well. The same thing happens to node pairs ('3', '4'), ('32', '29'), and ('9', '31') sequentially. Then the next largest-degree node is '14'; its most similar neighbor node '4' is already in the third community; therefore, we insert node '14' into the third community. All of the other nodes are processed in the same way, and in the subsequent operations, node pairs ('24', '30'), ('6', '7'), ('5', '11'), and ('25', '26') form new communities; all of the remaining nodes are inserted into communities to which their most similar neighbors belong. At the end of the process, we obtain the preliminary community structure as shown in Figure 1(b), in which each node connects to its most similar neighbor with a directed edge.

3.3. Merge of Small or Sparse Communities. At the end of the first phase of our proposed method, we obtain the preliminary community structure. However, some communities are either too small or too sparse to make sense, just like the preliminary communities {'5', '11'}, {'9', '31'}, {'32', '29'}, {'25', '26'}, {'28', '24', '30', '27'}, and {'6', '7', '17'} in Figure 1(b), because each of them contains only a few nodes, the inside edges of each of them are very sparse; the number of edges inside each of them is much smaller than that of edges connecting to outside, violating the characteristic that connections inside one community are much denser than those across different communities. Keeping them in the final community structure will lead to the low quality. Therefore, we merge some of the preliminary communities to acquire the final result in the second phase, which is carried out by function call **PCM()** in Algorithm 1.

To this end, there are two problems needed to be solved in **PCM()**. The first one is to identify which communities are small or sparse enough that need to be merged into another ones; the second one is to select the communities into which each of the small or sparse communities should be merged.



(a) (b)

Figure 1: The procedure of FPC() on the karate club network.

For the first problem, we propose an index, *community metric*, which takes into account two factors, community size and community sparsity, to find out the preliminary communities needed to be merged. Here, we formalize the relevant concepts and the index as Definition 1 through Definition 3.

Definition 1 (community sparsity). The sparsity of community C_i is defined as follows:

$$\alpha_i = \frac{\left|E_i^{in}\right|}{\left|E_i^{out}\right|}, \tag{4}$$

where E_i^{in} is the set of edges within community C_i and and E_i^{out} is the set of edges connecting nodes in community C_i with other communities.

That is to say, the sparsity of community C_i is defined as the ratio between the number of inner edges of C_i and the number of outer edges of C_i. Obviously, the more edges exist within community C_i, the larger the value of α_i will be, and vice versa.

Definition 2 (community scale). The scale of community C_i is formalized as follows:

$$\beta_i = \frac{|V_i|}{|V|}, \tag{5}$$

where V_i is the set of nodes in community C_i.

Obviously, the scale of community C_i is defined as the ratio of the number of nodes in C_i to the total number of nodes in the network. The more nodes there are in community C_i, the larger value the ratio will be, and vice versa.

Definition 3 (community metric). The community metric is a combination of both the community sparsity and the community scale, which is defined for community C_i as follows:

$$\gamma_i = \alpha_i * \beta_i. \tag{6}$$

On the basis of these definitions, the first problem can be solved by setting a community metric threshold, δ. That is to say, if $\gamma_i < \delta$, community C_i needs to be merged into another community.

For the second problem, we consider a strategy conforming to the construction of preliminary communities. The preliminary communities are formed based mainly on node similarity in the first phase; therefore, we also use the similarity as a criterion here to merge communities, i.e., each of the small or sparse communities is merged into its most similar adjacent community. Here, the similarity between two communities, C_i and C_j, is calculated as follows:

$$Sim\left(C_i, C_j\right) = \frac{\sum_{\substack{u \in C_i \\ v \in C_j}} sim\left(u, v\right)}{\left|C_j\right|}, \tag{7}$$

where $sim(u, v)$ is the similarity between nodes $u \in C_i$ and $v \in C_j$, which is calculated using (3). In function **PCM()** implementing the merge procedure, C_i is a community needed to be merged; C_j is one of its adjacent communities. The numerator of the right term in (7) is the sum of similarities between nodes in communities C_i and C_j. Dividing by the denominator, $|C_j|$, is a constraint on the priority for larger communities to prevent from forming some giant communities.

The logic of entire procedure of the second phase is listed in Algorithm 3; the operations are almost self-explanatory. The variable *CS* is used to record the final community structure; it is initialized as the preliminary community structure, *CS_pre* in step 1. Step 2 calculates the community metric for each of the preliminary communities, steps 3 and 4 select the community with the smallest community metric and its most similar community, step 5 merges them to yield a new community, and step 6 calculates the community metric for that new community. Step 7 replaces the two communities C_t and C_j with that new community in *CS* to reflect the effect of the merge operation. Step 8 repeats operations in steps 3 through 7, until the minimal community metric of the selected community is larger than the given threshold δ, meaning that all the remaining communities are satisfactory; therefore, the merge procedure is terminated and the resulting community structure in *CS* is returned in step 9.

Input: CS_pre, the preliminary community structure; δ, the community-metric threshold
Output: CS, the final community structure
1 Initialize CS, which is used to record the community structure:
$CS \longleftarrow CS\_pre$
2 Calculate the community metric for each of the preliminary communities:
foreach $C_i \in CS$ **do**
$\gamma_i \longleftarrow \alpha_i \times \beta_i$
3 Select the community with the minimal community metric, denote its index as t:
$t \longleftarrow \arg\min_i\{\gamma_i \mid i = 1, 2, \cdots, |CS|\}$
4 Identify the most similar community with C_t, denote its index as j:
$j \longleftarrow \arg\max_i\{Sim(C_t, C_i) \mid i = 1, 2, \cdots, |CS|, i \neq t\}$
5 Merge communities C_t and C_j to form a new community:
$k \longleftarrow |CS|; C_{k+1} \longleftarrow C_t \cup C_j;$
6 Calculate the community metric for the new community:
$\gamma_{k+1} \longleftarrow \alpha_{k+1} \times \beta_{k+1}$
7 Replace the two communities C_t and C_j with the new community to reflect the merging effect:
$CS = CS - \{C_t, C_j\} \cup \{C_{k+1}\}$
8 Repeat steps 3 through 7, until $\gamma_t > \delta$
9 **return** CS

Algorithm 3: **PCM**(CS_pre, δ): merge small or sparse communities.

3.4. Time Complexity. The proposed algorithm is comprised of two phases; the first one is to form the preliminary communities. The main time consumption in this phase is on the selection of the node with the largest degree (step 2 in Algorithm 2) and its most similar neighbor (step 3 in Algorithm 2); the former can be accomplished in $O(\log n)$ in each iteration using a max-heap data structure, the latter can be got down in $O(\log\langle d\rangle)$ with the max-heap, where $\langle d\rangle$ is the average degree of nodes in the network. Since $\langle d\rangle \ll n$, the time consumption of the first phase is $O(n \log n)$.

The second phase is used to improve the quality of the resulting community structure by merging some of the small or sparse communities. The major time is spent on determining the community needed to be merged and its most similar adjacent community in each iteration. Assuming there are K communities in the preliminary community structure, the former operation can be implemented in $O(\log K)$; the latter can also be carried out with $O(\log K)$ time consumption in the worst case. Hence, the second phase can be implemented with $O(K \log K)$ time consumption.

Since $K \ll n$, then $\log K \ll \log n$. Therefore, the proposed method can detect communities from networks with a relatively high efficiency, $O(n \log n)$ time complexity.

4. Experimental Results and Discussion

4.1. Network Datasets and Comparison System. To testify the performance of our proposed method, we have conducted extensive experiments on both some groups of artificial networks and some real-world networks. The artificial networks are synthesized using LFR benchmark network generator [50], which works with some parameters to control the characteristics of generated networks. Here, we consider the influences of both the network scale and community size; therefore, four types of networks are generated, say, small networks with small communities and big communities and larger networks with small communities and big communities, respectively. Each of the small networks and larger networks contains 1000 and 5000 nodes, respectively; the small community contains about 10 nodes at least and 50 nodes at most; the minimum and maximum number of nodes in the big communities are 20 and 100, respectively. The generated networks with small communities and big communities are marked using the suffixes 's' and 'b', individually. The exponents of the power-law distributions that node degree and community size follow are the default values, -2 and -1, respectively. The parameters used to synthesize the four groups of artificial networks are listed in Table 1.

We also performed the experiments on 13 real-world networks; the size of these networks spans from tens to hundreds of thousands of nodes; the information about them is listed in Table 2. These real-world networks can be divided into two categories: the first category includes the first four networks whose ground-truth communities are known a priori; the second one contains the other nine networks, which have no publicly acknowledged ground-truth community structures.

On these networks, we ran our proposed method to detect community structures from them and compared the results to those of 5 popular community detection algorithms, namely, FastQ[24], WalkTrap [38], LPA[28], Attractor[41], IsoFdp[36], which have been already introduced in Section 2. For LPA, since it is a nondeterministic algorithm, we ran it on each network 10 times and take the average of the evaluation metrics as its resulting metric value obtained from that network. For our proposed method, NSA, we empirically set $\delta = 0.13$ for the dolphin social network and $\delta = 0.1$ for other networks in the experiments. The details of how to set the optimal value of δ will be discussed in Section 5.

4.2. Evaluation Metrics. Two indexes, namely, NMI (Normalized Mutual Information) [51] and modularity[7], are

Table 1: The parameters used to generate the LFR networks. In the header row of this table, n is number of nodes contained in the network; $\langle d \rangle$ and d_{max} are the average degree and the max degree, respectively; $\exp_d$ and $\exp_{com}$ are the exponents of the power law distributions that node degree and community size follow; $\min(C_i)$ and $\max(C_i)$ represent the minimal and maximal number of nodes contained in every community, respectively.

Network	n	$\langle d \rangle$	d_{max}	$\exp_d$	$\exp_{com}$	$\min(C_i)$	$\max(C_i)$
LFR1000s	1000	20	50	-2	-1	10	50
LFR1000b	1000	20	50	-2	-1	20	100
LFR5000s	5000	20	50	-2	-1	10	50
LFR5000b	5000	20	50	-2	-1	20	100

Table 2: The information about the real-world networks. n and m are the number of nodes and edges in the network, respectively.

Network	n	m
Karate club[14]	34	78
Dolphin social network[15]	62	159
Risk map[16]	42	83
Scientists collaboration network [6]	118	197
Lesmis[17]	77	254
Polbooks[3]	105	441
ColiNeta[18]	423	519
NetScience[10]	1589	2742
Email[19]	1133	5451
YeastL[20]	2361	7182
PGP[21]	10680	24316
DBLP[22]	317080	1049866
Amazon[22]	334863	925872

adopted as the measure metrics to evaluate the quality of the detected community structure in this paper. The NMI between the ground-truth community structure $P = \{P_1, P_2, \ldots, P_K\}$ and the extracted one $P' = \{P'_1, P'_2, \ldots, P'_{K'}\}$ is calculated as follows:

$$\mathrm{NMI}\left(P, P'\right) = \frac{-2\sum_{i=1}^{|P|}\sum_{j=1}^{|P'|} n_{ij}\log\left(\left(n_{ij}\cdot n\right)/\left(n_i^P \cdot n_j^{P'}\right)\right)}{\sum_{i=1}^{|P|} n_i^P \log\left(n_i^P/n\right) + \sum_{j=1}^{|P'|} n_j^{P'}\log\left(n_j^{P'}/n\right)}, \quad (8)$$

where $n_i^P = |P_i|$, $n_j^{P'} = |P'_j|$, and $n_{ij} = |P_i \cap P'_j|$, respectively.

The NMI is an information-theory based metric, which measures how much the detected community structure agrees with the ground truth. Therefore, it can only be used to evaluate the quality of the detected community structure on networks whose ground-truth community structure is already known. Its value is in the range of $[0, 1]$, larger is better.

Another metric widely used to evaluate the performance of community detection method is modularity[7], which is defined as follows:

$$Q = \sum_i \left(e_{ii} - a_i^2\right), \quad (9)$$

where e_{ii} is the diagonal element of a $K \times K$ matrix $\boldsymbol{e}$, whose element e_{ij} is the fraction of edges between nodes in communities C_i and C_j to the total edges in the network, K is the number of communities in the community structure; a_i is the fraction of edges associated with nodes in community C_i.

The first term $\sum_i e_{ii}$ in the right of (9) is the fraction of edges within communities; the second term $\sum_i a_i^2$ is the expected value of the same fraction in a random graph, in which nodes and degree distribution are the same as in the original network, but edges are connected between nodes randomly. The smaller difference is between the two terms; the more the network approaches a random graph, then the weaker the community structure is. On the contrary, the larger the difference between them is, the network departs further from the random graph, then the stronger the community structure is. That is to say, the modularity measures quality of the community structure from the perspective of how far the detected result deviates from a random network; its effective value falls in $[0, 1]$, higher is better.

4.3. Synthetic Networks. We carried out experiments on four groups of artificial networks to testify the performance of the proposed method. As mentioned above, all the four types of artificial networks are synthesized using the LFR benchmark generator software [50]. Besides the parameters listed in Table 1, another critical parameter for this software is the mixing parameter, μ, which regulates for each node the ratio of edges connected to nodes in other communities. The smaller the value of μ is, the clearer the community structure will be. Obviously, $\mu = 0.5$ is a transitive point, above which communities in networks tend to be obscure.

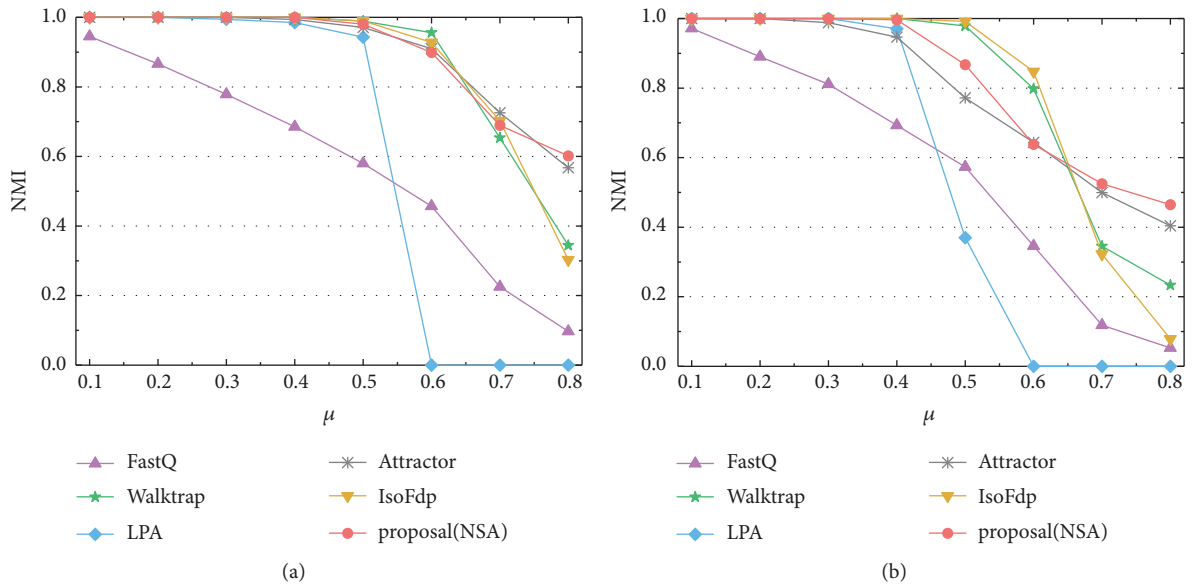


Figure 2: *Comparison of different community-detection algorithms on LFR benchmark networks containing 1000 nodes.* (a) The results detected from small network with small-sized communities. (b) The results identified from small networks with big-sized communities.

In our experiments, we varied the value of μ from 0.1 to 0.8 with an increment of 0.1 for each group of LFR networks. To eliminate the occasionality, we generated 10 networks for each value of μ while keeping the same setting for other parameters. Since the community structures have been already embedded in these synthetic networks, we use NMI as the metric to evaluate the performance of our proposed method and the comparison algorithms. We took these networks as the input one by one to run our proposed method and the comparison algorithms to detect communities and use the average of NMI as the resulting metric. The results detected by our proposal and the comparison algorithms from the small networks with small-sized communities or big-sized communities are illustrated in Figures 2(a) and 2(b), respectively; the results revealed from the larger networks with small-sized communities and big-sized communities are presented in Figures 3(a) and 3(b), separately.

In Figures 2(a) and 2(b), FastQ tends to introduce mistakes in the results no matter communities in networks are well separated or obscure. As mentioned previously, FastQ is a typical modularity-optimization based algorithm; it aims only at acquiring results with larger modularity, rather than high accuracy. In our experiments, all of the results uncovered by it are not satisfactory. Even in the networks with $\mu = 0.1$, it still failed to identify the exact communities, and furthermore, its performance is the worst in comparison algorithms for $\mu \leqslant 0.5$. For $\mu > 0.5$, the quality of its results is only better than that of LPA. LPA performed as well as other comparison algorithms in those networks for $\mu < 0.5$, but its performance dropped dramatically for $\mu \geqslant 0.5$; it even could not detect the effective communities from networks for $\mu > 0.6$. This might be due to its own label-update mechanism; when the community boundaries become obscure, nodes tend to accept incorrect labels to update their own ones, always leading to the trivial results; even all nodes are labeled as members of one giant community. The proposed method, NSA, acquired NMI = 1 on all networks for $\mu < 0.5$, meaning that the detected partitions are perfectly matched with the ground-truth community structures in these networks. For $\mu = 0.5$, NSA also obtained the results as better as those of WalkTrap, Attractor, and IsoFdp. For $\mu > 0.5$, there has been a slip in the quality of the detected community structures for all those three algorithms and the proposed method. For $0.5 < \mu \leqslant 0.6$, the quality of our proposal is better than that of Attractor in networks with larger communities; and for $\mu \geqslant 0.7$, the performance of our proposed method is the best.

In Figures 3(a) and 3(b), we obtained the similar results as those in Figure 2 overall. But they still differ from each other in some way. In Figure 3(a), our proposed method performed the best on almost all networks. For $0.5 < \mu < 0.7$ in Figure 2, NMI of the results extracted by our proposed method is lower than those of WalkTrap and IsoFdp; however, in Figure 3, the proposed method performed better than IsoFdp for $\mu > 0.5$. These results suggest that the performances of the comparison algorithms are not stable on different networks, but our proposed method can steadily extract high-quality community structures from networks with different characteristics. This is also can be manifested from the fact that all the curves of the proposed method in these figures decline more slowly than others. Moreover, we can draw a conclusion by comparing the curves of the proposal's own in these figures that our proposed method inclines to perform better on larger networks with small communities; therefore, it overcomes the problem of resolution limit to some extent.

4.4. Real-World Networks. We also carried out experiments on 13 real-world networks to further test the effectiveness and efficiency of our proposed method. As mentioned in Section 4.1, these networks fall in two categories, ones with

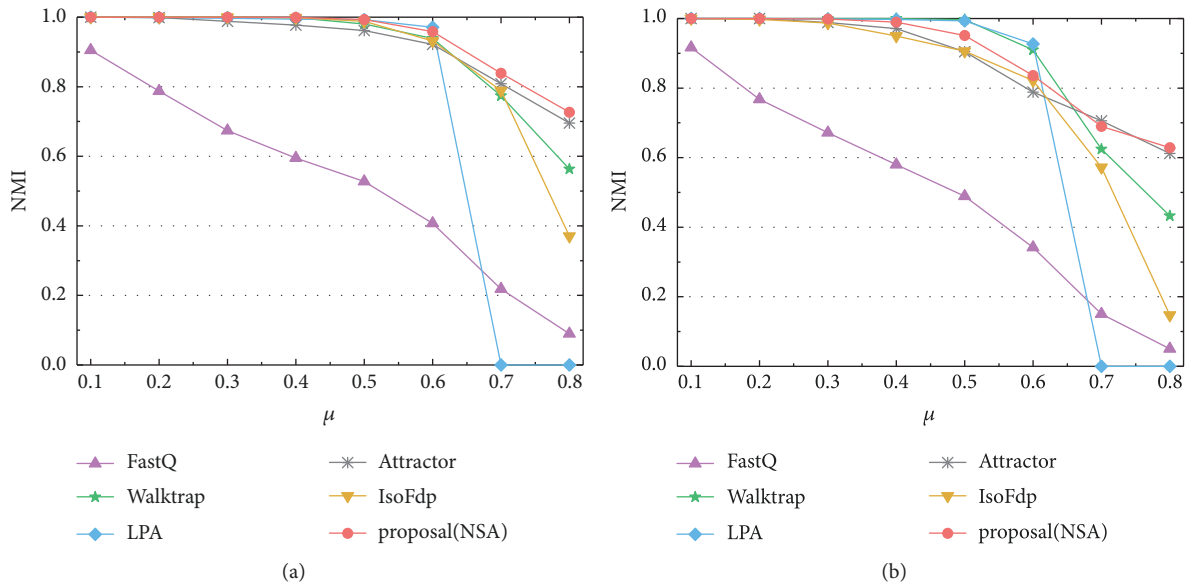


FIGURE 3: *Comparison of different community detection algorithms on LFR benchmark networks containing 5000 nodes.* (a) The results extracted from the larger networks with small-sized communities. (b) The results revealed from the larger networks with big-sized communities.

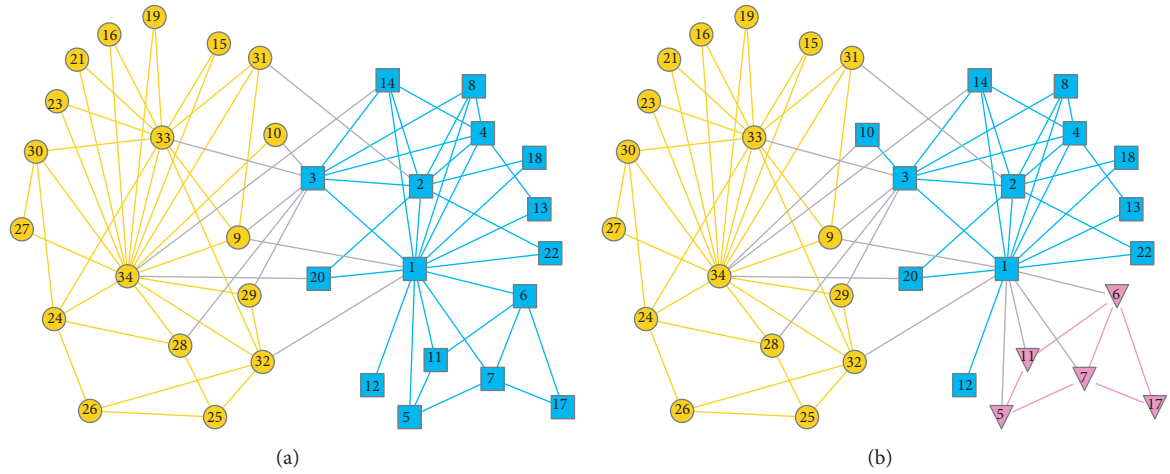


FIGURE 4: *The karate club network.* (a) The ground-truth community structure. (b) The community structure detected by our proposed method, NSA. (The nodes in different communities are plotted in different colors and shapes; this illustration style is also applied in the subsequent figures.)

the ground-truth community structure known a priori and the other ones without publicly acknowledged ground truth.

Networks with Ground-Truth Community Structure. This category includes the first 4 networks listed in Table 2; since their ground-truth community structure is already known, we measure the quality of the community structures identified by the proposed method and comparison algorithms in terms of both NMI and modularity. The values of the two metrics obtained by the proposed method and comparison algorithms have been recorded in Table 3. The scales of these networks are relatively small, facilitating to us visualizing the detected results. Below, we analyze the results extracted by the proposed method from these networks individually.

The Karate Club Network. This is a network depicting the friendships among members of a karate club; it contains 34 nodes and 78 edges. This network was compiled by Wayne W. Zachary, who observed the karate club for 3 years. During the period of study of Zachary, the club split into two factions because of a dispute arisen between the administrator and the instructor. Corresponding to the two parts, the network is always taking the partition of two communities as the ground truth, which is shown in Figure 4(a). The result detected by our proposed method is presented in Figure 4(b).

From Figure 4, we can see that our proposed method detected 3 rather than 2 communities from the network. It seems that the detected result deviates from the ground truth in some ways, but this result coincides with the conclusion

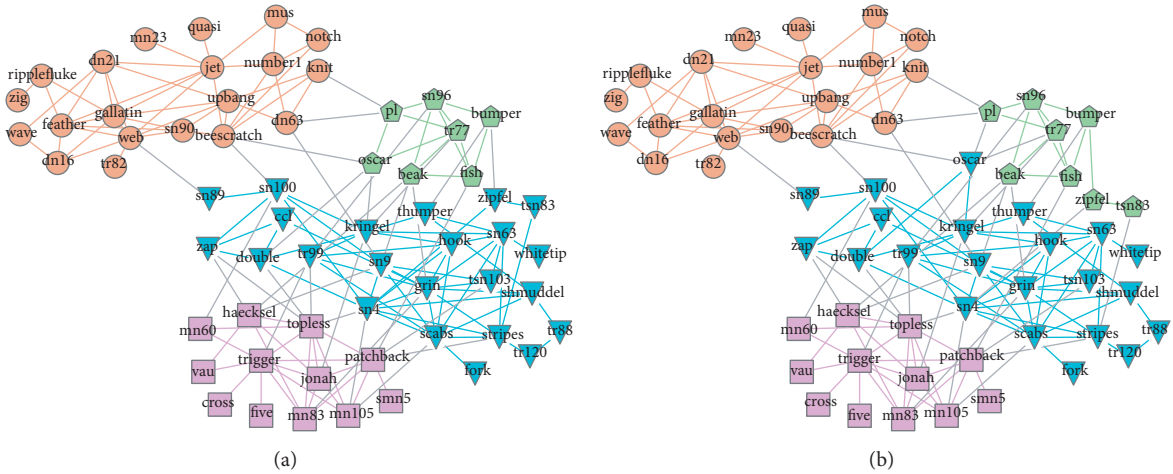


FIGURE 5: *The dolphin social network*. (a) The ground-truth community structure. (b) The community structure identified by our proposed method, NSA.

TABLE 3: The experimental results on networks with ground-truth community structures. The largest values of the two measure metrics are typed in bold.

Network	Metric	FastQ	WalkTrap	LPA	Attractor	IsoFdp	NSA
Karate	Q	0.381	0.353	0.355	0.371	0.371	**0.402**
	NMI	0.693	0.504	0.62	0.924	**1.00**	0.699
Dolphin	Q	0.492	0.489	0.464	0.45	0.505	**0.513**
	NMI	0.719	0.632	0.719	0.69	0.744	**0.887**
Risk map	Q	**0.625**	0.624	0.59	0.598	0.519	0.624
	NMI	**0.894**	0.848	0.821	0.839	0.714	0.848
Scientists	Q	**0.749**	0.733	0.64	0.694	0.668	0.744
	NMI	0.867	0.818	0.743	0.835	0.823	**0.878**

found in the experiments on synthetic networks that our proposed method tends to find small communities from networks to overcome the problem of resolution limit. Moreover, considering from the perspective of measure metrics, the modularity corresponding to the detected result is the largest among those of comparison algorithms. Although our proposed method is not based on the strategy of optimizing modularity, it inclines to acquire the community structure with as larger modularity as possible. If it is not the largest, it is the second largest with a small offset to the largest. These findings can also be manifested in next networks.

Lusseau's Dolphin Social Network. This network describes the interactions of a group of dolphins living in Doubtful Sound, New Zealand. It consists of 62 nodes and 159 edges, which represent dolphin individuals and the cooccurrences of pairs of dolphins being observed, respectively. This network is generally partitioned into 4 groups as the ground-truth community structure, which is as exhibited in Figure 5(a). Figure 5(b) is the community structure uncovered by our proposed method.

In Figure 5, our proposed method detected communities from this network with a high degree of success, it identified 4 communities as well, the absolute majority of nodes are classified into the correct communities, and the result almost approaches the ground-truth community structure. Considering quantitatively, both the values of NMI and modularity corresponding to the result detected by the proposed method from this network are the largest among those of comparison algorithms, which means that the community structure identified by the proposed method is obviously better than those of comparison algorithms.

Risk Map Network. This network is a world political map loaded in the popular game, Risk (https://en.wikipedia.org/wiki/Risk_(game)), in which 42 countries or territories of 6 continents are involved. Therefore, 42 nodes and 83 edges connecting adjacent countries or territories are organized in 6 communities as the ground truth, which is illustrated in Figure 6(a). Feeding this network into the proposed method, we obtained the community structure as shown in Figure 6(b).

Comparing the detected result to the ground truth community structure, the community containing nodes '18' and '23' in the ground truth is split into two small communities in Figure 6(b), owning to the tendency of the proposed method. Besides this, nodes '26', '33', and '34' are misclassified into the wrong communities in the detected result. But nodes '12', '16', '26', '33', and '34' are special ones in this network; the outer edges associated with them are no less

Table 4: The experimental results of modularity on networks. The largest values of the two measure metrics are typed in bold.

Network	FastQ	WalkTrap	LPA	Attractor	IsoFdp	NSA
Lesmis	0.499	0.519	0.515	0.498	0.491	**0.54**
Polbooks	0.502	0.507	0.508	0.501	0.518	**0.524**
ColiNeta	**0.779**	0.746	0.693	0.718	-	0.761
Email	0.499	0.531	0.379	0.464	0.531	**0.544**
NetScience	0.955	0.956	0.896	0.937	-	**0.957**
YeastL	0.573	0.529	0.372	0.511	-	**0.574**
PGP	0.85	0.789	0.765	0.768	0.726	**0.867**
DBLP	0.735	-	0.652	0.637	-	**0.782**
Amazon	0.869	-	0.743	0.741	-	**0.898**

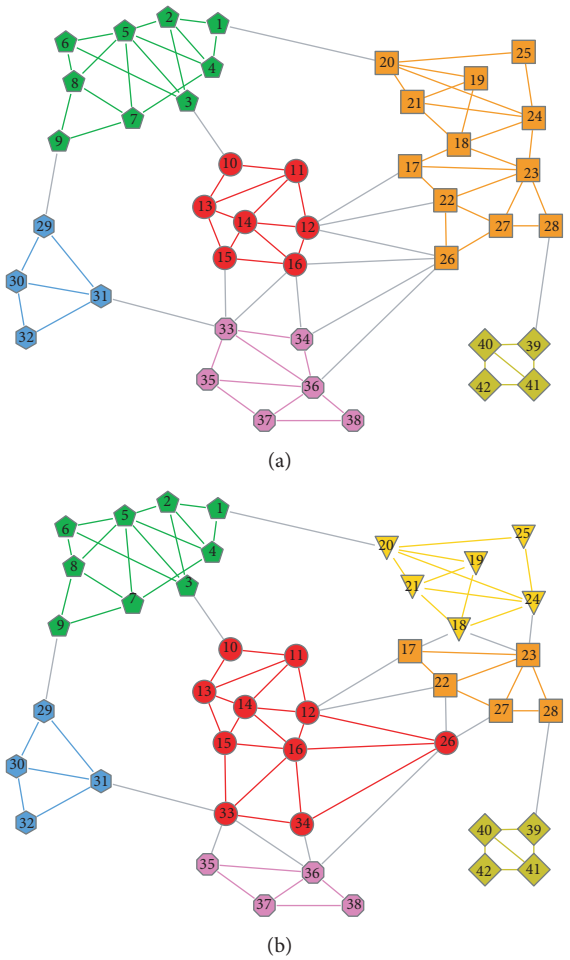


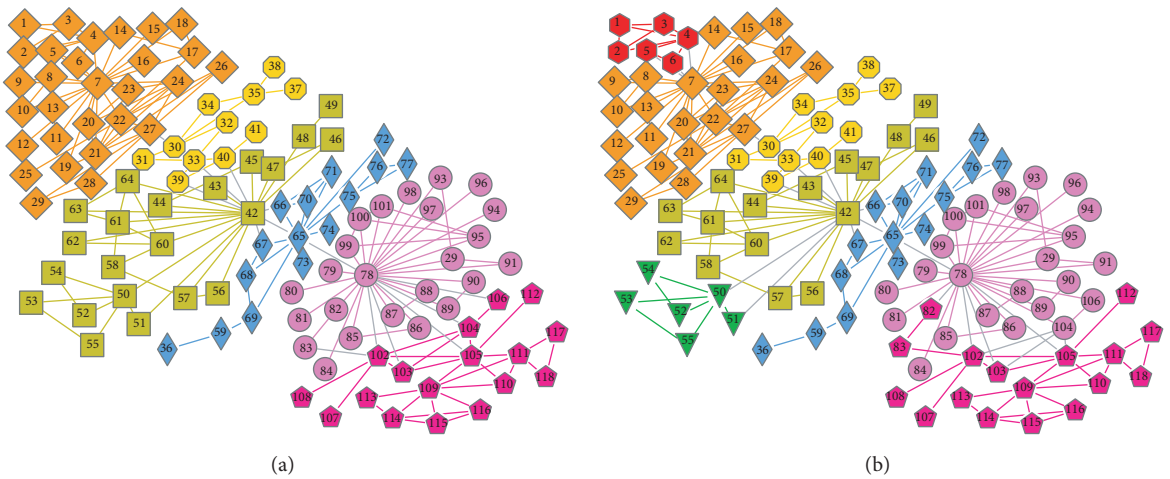
Figure 6: *Risk map network*. (a) The ground-truth community structure. (b) The community structure uncovered by our proposed method, NSA.

even more than those within the communities to which these nodes belong. Therefore, if we ignore the meaning of the actual representation of these nodes and consider qualitatively based on the topology only, the community structure extracted by our proposed method is more rational than the ground truth; more edges associated with these three nodes are located within the community than in the ground truth; thus, more tightly these three nodes are connected to nodes within the same community in Figure 6(b). When considering quantitatively, both values of the two measure metrics of our proposed method are second only to those of FastQ and are the same with those of WalkTrap. These results also confirm that our proposed method provides us with an acceptable solution to the problem of community detection.

Scientists Collaboration Network. This is the largest connected component of a network delineating the coauthor relationship among scientists working at the Santa Fe Institute, New Mexico. Nodes in this network represent scientists; edges stand for the two scientists who have collaborated at least on one paper. There are 118 nodes and 197 edges in total in this network. The nodes can be divided into 6 groups as the ground-truth communities according to the specialties of the scientists, which is as presented in Figure 7(a). Taking this network as the input to the proposed method, we obtained the community structure as illustrated in Figure 7(b).

The proposed method revealed 8 communities from this network; two additional communities are detected in Figure 7(b). These two communities are relatively independent components, especially for the community containing nodes '1'; there are much more inner edges than outer edges. That is to say, nodes in these two communities are connected more tightly to one another than with the remainder of the network. Therefore, isolating them from the network and taking them as independent communities are also reasonable. Considering from the perspective of measure metrics, the value of NMI obtained by the proposed method is the largest, which suggests that the result detected by our proposal is the one most approaches the ground-truth community structure; the modularity value of the proposed method is not the largest though; it is also second only to that of FastQ. These results also testify that our proposed method can extract high-quality community structure from networks.

Networks without Ground-Truth Community Structure. This category contains the last 9 real-world networks listed in Table 2. For the experiments carried out on this category of networks, we evaluate the quality of the extracted community structures using the modularity only due to the absence of the ground-truth community structures. For the proposed method and comparison algorithms, the obtained values of modularity have been recorded in Table 4. To illustrate them



(a) (b)

Figure 7: *The collaboration network of scientists working at the Santa Fe Institute*. (a) The ground-truth community structure. (b) The community structure detected by our proposed NSA algorithm.

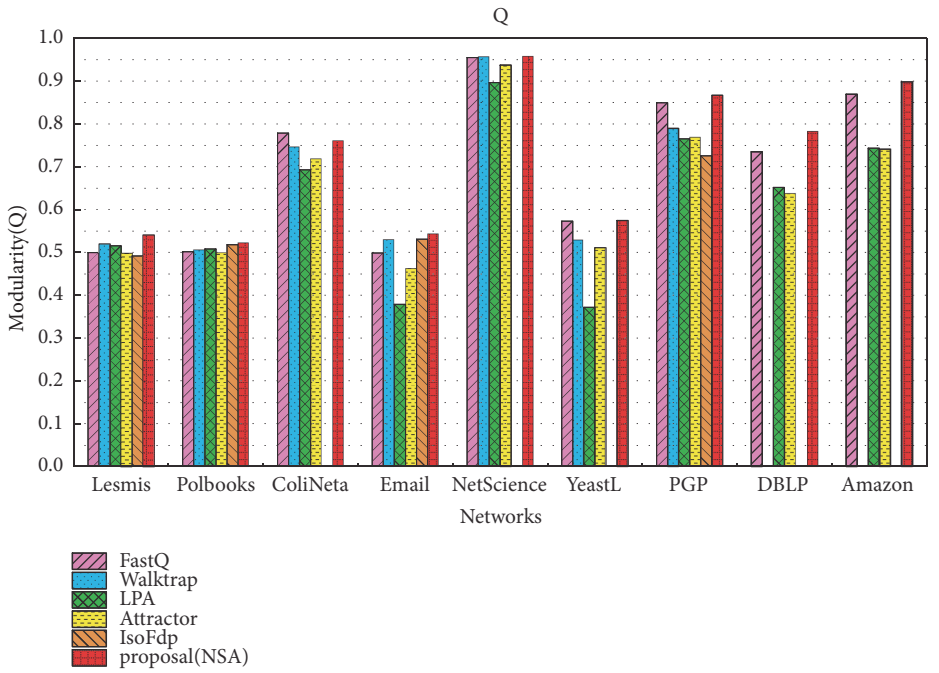
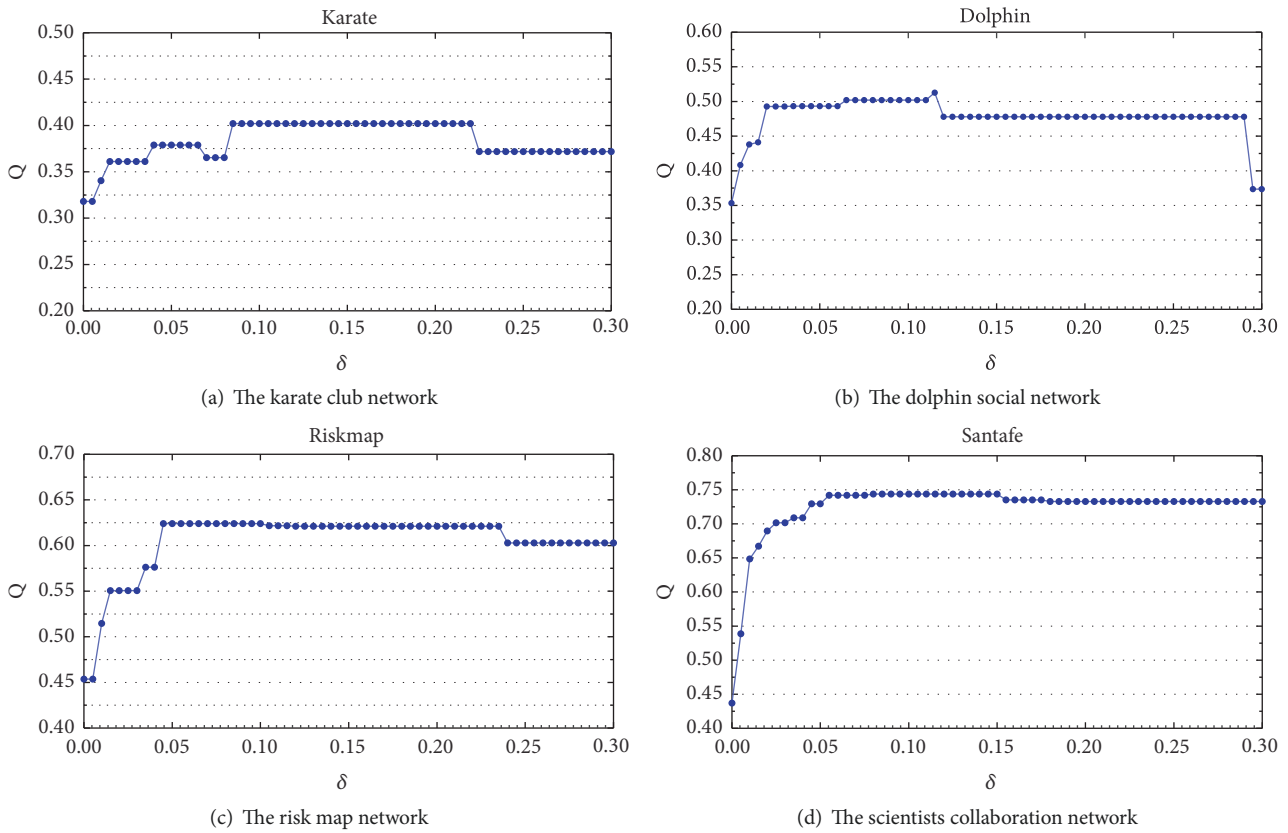


Figure 8: The bar chart of the modularity obtained by comparison algorithms and the proposed method, NSA.

intuitively, we also plotted them in a bar chart, which is presented in Figure 8.

On these networks, our proposed method achieved the largest modularity from 8 of them. On the only other one network, ColiNeta, it still obtained the second largest value of modularity. For FastQ, it is based on the modularity optimization strategy though; it acquired the largest value of modularity on network ColiNeta only. For WalkTrap, it is an approach based on random walk; then its time complexity is relatively high. It cannot manage to get effective results from networks Amazon and DBLP, due to the large scale of these two networks. For LPA and Attractor, they can extract community structures from all those networks, but the quality of the detected results is not satisfactory. For IsoFdp, it can only be applied to connected networks and cannot run on networks ColiNeta, NetScience, and YeastL, as these three networks are disconnected. It cannot detect the community structure from networks Amazon and DBLP effectively either because of their large scale. These comparison results manifest that our proposed method can steadily, effectively, and efficiently provide us with promising solutions for the problem of community detection in networks of wide-range applications and outperform comparison algorithms significantly.



(a) The karate club network

(b) The dolphin social network

(c) The risk map network

(d) The scientists collaboration network

Figure 9: The setting of parameter δ.

5. Parameter Setting

In the second phase of the proposed method, we introduce a threshold δ for the community metric to identify the preliminary communities needed to be merged. As aforementioned, we calculate the community metric $\gamma_i = \alpha_i \times \beta_i$ for every preliminary community C_i in the merge procedure; if the value of γ_i is below the threshold δ, the corresponding community C_i is identified as the one needed to be merged.

Therefore, δ works as a parameter in our proposed method, whose setting can influence the quality of the resulting community structure. Considering qualitativity, the larger or the sparser the network is, the threshold δ should be smaller in accordance with the definitions of community sparsity (α_i), community scale (β_i), and community metric (γ_i). To determine the optimal value of δ, we conduct a group of experiments to explore the relationship between the value of δ and the quality of the resulting community structure on the first four networks listed in Table 2, namely, the karate club network, the dolphin social network, the map of game Risk, and the scientists collaboration network, respectively. The quality of the resulting community structure is measured in term of modularity Q. We vary the value of δ from 0 to 1.0 by increasing 0.005 each time; for each value of δ, we run our proposed method on these networks and observe the change of modularity along with the varies of δ.

The observed results are as illustrated in Figure 9, in which we plotted only the proportion of $\delta \in [0, 0.3]$ because the largest modularities are obtained during $\delta \leqslant 0.3$ on all of those four networks. Our proposed method gets the largest modularity when $\delta = 0.13$ on the dolphin social network and $\delta = 0.1$ on the other three networks. Therefore, we adopt the corresponding value for those four networks and empirically set $\delta = 0.1$ for other networks to perform the experiments. In Figure 9, the largest modularity is obtained around the value of $\delta = 0.1$, and the interval of $[0.05, 0.2]$ covers the optimal value of δ. Therefore, we empirically suggest that δ be adjusted adaptively around 0.1 in the range of $[0.05, 0.2]$ according to the size and the sparsity of networks involved in real-world applications.

6. Conclusion

In this paper, we presented a novel method to detect communities from networks. It is a local method based on node similarity and overcomes the deficiency of high time consumption of global methods. First, we construct the preliminary community structure by repeatedly selecting the node with the largest degree and either taking it as the exemplar of a new community or inserting it into the community to which its most similar neighbor belongs; on the basis of its most similar neighbor's community assignment, i.e., if its most similar neighbor has not been assigned to any community yet, we create a new community for it and its most similar neighbor; if its most similar neighbor has been assigned to a certain community, we insert it into

that community as well. At the end of this process, we obtain a series of preliminary communities. However, some of them might be too small or too sparse, leading to a low-quality result. Therefore, we merge some of the preliminary communities to acquire the final community structure. To do so, we also proposed some indexes which take both the size and sparsity of communities into account to determine which communities should be merged.

To test the performance of the proposed method, we have performed extensive experiments on four groups of synthetic networks and 13 real-world networks and compared the detected community structures with the results extracted by comparison algorithms in terms of NMI and modularity; the comparison results demonstrate that our proposed method can extract high-quality community structures from networks abstracted from various applications, and nodes in the extracted communities are connected more tightly. The proposed method overcomes the problem of resolution limit to some extent and outperforms the competitors successfully.

Data Availability

We have conducted experiments on some artificial networks and some real-world datasets. The artificial networks are synthesized using LFR benchmark network generator, which can be freely available at https://sites.google.com/site/santofortunato/. The parameters used to synthesize the artificial networks are listed in Table 1. The real-world data supporting this study are from previously reported studies, which have been cited in Table 2. Most of the real-world datasets can also be downloaded from http://www-personal.umich.edu/~mejn/netdata/ and https://snap.stanford.edu/data/index.html. The ColiNeta dataset was provided by Jeong et al. [18]. We construct the Risk Map network manually according to the literature [16].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Grant ID: 61602225).

References

[1] J. Kleinberg and S. Lawrence, "Network analysis: The structure of the web," *Science*, vol. 294, no. 5548, pp. 1849-1850, 2001.

[2] P. Chen and S. Redner, "Community structure of the physical review citation network," *Journal of Informetrics*, vol. 4, no. 3, pp. 278–290, 2010.

[3] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577–8582, 2006.

[4] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A. L. Barabási, "Hierarchical organization of modularity in metabolic networks," *Science*, vol. 297, no. 5586, pp. 1551–1555, 2002.

[5] R. Guimerà and L. A. N. Amaral, "Functional cartography of complex metabolic networks," *Nature*, vol. 433, no. 7028, pp. 895–900, 2005.

[6] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.

[7] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 2, Article ID 026113, 2004.

[8] P. M. Gleiser and L. Danon, "Community structure in jazz," *Advances in Complex Systems (ACS)*, vol. 6, no. 4, pp. 565–573, 2003.

[9] Y. van Gennip, B. Hunter, R. Ahn et al., "Community detection using spectral clustering on sparse geosocial data," *SIAM Journal on Applied Mathematics*, vol. 73, no. 1, pp. 67–83, 2013.

[10] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 74, no. 3, Article ID 036104, 19 pages, 2006.

[11] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, 2010.

[12] S. Fortunato and D. Hric, "Community detection in networks: a user guide," *Physics Reports*, vol. 659, pp. 1–44, 2016.

[13] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell Labs Technical Journal*, vol. 49, no. 1, pp. 291–307, 1970.

[14] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.

[15] D. Lusseau, "The emergent properties of a dolphin social network," in *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 270, supplement 2, pp. S186–S188, 2003.

[16] K. Steinhaeuser and N. V. Chawla, "Identifying and evaluating community structure in complex networks," *Pattern Recognition Letters*, vol. 31, no. 5, pp. 413–421, 2010.

[17] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[18] H. Jeong, B. Tombor, R. Albert, Z. N. Oltval, and A.-L. Barabásl, "The large-scale organization of metabolic networks," *Nature*, vol. 407, no. 6804, pp. 651–654, 2000.

[19] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 68, no. 6, Article ID 065103, 2003.

[20] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[21] M. Boguñá, R. Pastor-Satorras, A. Díaz-Guilera, and A. Arenas, "Models of social networks based on social distance attachment," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 70, no. 5, Article ID 056122, 2004.

[22] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowledge and Information Systems*, vol. 42, no. 1, pp. 181–213, 2015.

[23] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 6, Article ID 066133, 2004.

[24] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E:*

Statistical, Nonlinear, and Soft Matter Physics, vol. 70, no. 6, Article ID 066111, 2004.

[25] F. Dabaghi Zarandi and M. Kuchaki Rafsanjani, "Community detection in complex networks using structural similarity," *Physica A: Statistical Mechanics and its Applications*, vol. 503, pp. 882–891, 2018.

[26] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, Article ID P10008, 2008.

[27] L. Waltman and N. J. Van Eck, "A smart local moving algorithm for large-scale modularity-based community detection," *The European Physical Journal B*, vol. 86, no. 11, article 471, pp. 1–14, 2013.

[28] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 76, no. 3, Article ID 036106, 2007.

[29] M. J. Barber and J. W. Clark, "Detecting network communities by propagating labels under constraints," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 80, no. 2, Article ID 026129, 2009.

[30] J. Hou Chin and K. Ratnavelu, "A semi-synchronous label propagation algorithm with constraints for community detection in complex networks," *Scientific Reports*, vol. 7, Article ID 45836, 2017.

[31] J. Ding, X. He, J. Yuan, Y. Chen, and B. Jiang, "Community detection by propagating the label of center," *Physica A: Statistical Mechanics and its Applications*, vol. 503, pp. 675–686, 2018.

[32] A. Laio and A. Rodriguez, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[33] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger, "SCAN: A structural clustering algorithm for networks," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*, pp. 824–833, ACM, New York, NY, USA, August 2007.

[34] M. Este, H. P. Kriegel, S. Jörg, and x. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*, pp. 226–231, AAAI Press, 1996.

[35] H. Shiokawa, Y. Fujiwara, and M. Onizuka, "Scan++: Efficient algorithm for finding clusters, hubs and outliers on large-scale graphs," in *Proceedings of the 3rd Workshop on Spatio-Temporal Database Management, STDBM 2006, Co-located with the 32nd International Conference on Very Large Data Bases, VLDB 2006*, pp. 1178–1189, Republic of Korea, September 2006.

[36] T. You, H.-M. Cheng, Y.-Z. Ning, B.-C. Shia, and Z.-Y. Zhang, "Community detection in complex networks using density-based clustering algorithm and manifold learning," *Physica A: Statistical Mechanics and its Applications*, vol. 464, pp. 221–230, 2016.

[37] X. Wang, G. Liu, J. Li, and J. P. Nees, "Locating structural centers: A density-based clustering method for community detection," *PLoS ONE*, vol. 12, no. 1, Article ID e0169355, 2017.

[38] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *International symposium on computer and information sciences*, pp. 284–293, 2005.

[39] S. A. Tabrizi, A. Shakery, M. Asadpour, M. Abbasi, and M. A. Tavallaie, "Personalized PageRank clustering: a graph clustering algorithm based on random walks," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 22, pp. 5772–5785, 2013.

[40] Y. Su, B. Wang, and X. Zhang, "A seed-expanding method based on random walks for community detection in networks with ambiguous community structures," *Scientific Reports*, vol. 7, Article ID 41830, 2017.

[41] J. Shao, Z. Han, Q. Yang, and T. Zhou, "Community detection based on distance dynamics," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1075–1084, ACM, Australia, August 2015.

[42] H.-L. Sun, E. Ch'ng, X. Yong, J. M. Garibaldi, S. See, and D.-B. Chen, "A fast community detection method in bipartite networks by distance dynamics," *Physica A: Statistical Mechanics and its Applications*, vol. 496, pp. 108–120, 2018.

[43] A. A. Amini, A. Chen, P. J. Bickel, and E. Levina, "Pseudo-likelihood methods for community detection in large sparse networks," *The Annals of Statistics*, vol. 41, no. 4, pp. 2097–2122, 2013.

[44] S. C. de Lange, M. A. de Reus, and M. P. van den Heuvel, "The laplacian spectrum of neural networks," *Frontiers in Computational Neuroscience*, vol. 7, no. 189, 2014.

[45] F. Krzakala, C. Moore, E. Mossel et al., "Spectral redemption in clustering sparse networks," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 110, no. 52, pp. 20935–20940, 2013.

[46] P. Shi, K. He, D. Bindel, and J. E. Hopcroft, "Local Lanczos Spectral Approximation for Community Detection," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, vol. 10534 of *Lecture Notes in Computer Science*, pp. 651–667, Springer International Publishing, 2017.

[47] R. Tackx, F. Tarissan, and J. Guillaume, "ComSim: a bipartite community detection algorithm using cycle and node's similarity," in *International Workshop on Complex Networks and their Applications*, vol. 689 of *Studies in Computational Intelligence*, pp. 278–289, Springer International Publishing, 2017.

[48] T. Wang, L. Yin, and X. Wang, "A community detection method based on local similarity and degree clustering information," *Physica A: Statistical Mechanics and its Applications*, vol. 490, pp. 1344–1354, 2018.

[49] K. R. Žalik, "Maximal neighbor similarity reveals real communities in networks," *Scientific Reports*, vol. 5, Article ID 18374, 2015.

[50] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 78, no. 4, Article ID 046110, 2008.

[51] L. Ana and A. Jain, "Robust data clustering," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. II-128–II-133, Madison, WI, USA, 2003.