Research Article

Community Detection by Node Betweenness and Similarity in Complex Network

Chong Feng ², **Jianxu Ye**,² **Jianlu Hu** ² and **Hui Lin Yuan**¹

¹College of Management, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China

²College of Information Science and Engineering, Northeastern University, Shenyang 110819, China

Correspondence should be addressed to Chong Feng; 15526891069@163.com

Received 22 March 2021; Revised 22 May 2021; Accepted 21 June 2021; Published 23 July 2021

Academic Editor: Honglei Xu

Copyright © 2021 Chong Feng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Community detection of complex networks has always been a hot issue. With the mixed parameters μ increase in network complexity, community detection algorithms need to be improved. Based on previous work, the paper designs a novel algorithm from the perspective of node betweenness properties and gives the detailed steps of the algorithm and simulation results. We compare the proposed algorithm with a series of typical algorithms through experiments on synthetic and actual networks. Experimental results on artificial and real networks demonstrate the effectiveness and superiority of our algorithm.

1. Introduction

Complex networks have made significant affection in understanding complex systems [1]. At present, the application fields of complex networks include biology [2, 3], epidemiology [4], sociology [5, 6], transportation [7], etc. Through the study of these real networks, we can better understand the characteristics of the network and discover the properties of the network [8]. The network is composed of nodes and edges, which can represent various individuals and relationships in the real world. For example, the degree of nodes in the network can measure their importance [9, 10], and another important aspect is the community structure in the network [11]. In most cases, nodes in the same community have similar attributes or closer links. The community structure is strongly significance for disease prevention, information dissemination, and product recommendation. However, the existing community discovery methods cannot fully meet the needs of current applications [12].

In recent years, many methods have been discovered about the community detection [13], including the label propagation algorithm (LPA), edge deletion methods, and spectral clustering. However, these algorithms are limited to a certain extent. For example, although the label propagation algorithm (LPA) runs faster, its results are unstable;

although the accuracy of the GN algorithm is relatively high, the running time of the GN algorithm is much long. If the scale of the network is huge, then the GN algorithm will take a long time; although the spectral clustering algorithm has better ability to find communities in a simple network, the spectral clustering algorithm has high requirements on the similarity matrix, and different similarity matrices may get the different effect. Although the running time of the EM algorithm is much short, it is found that the accuracy of the community is decreasing as the complexity of the network increases. Therefore, it is significant to detect the community structure of networks accurately. Based on the above information, we propose a method based on the combination of node betweenness and network structure similarity to discover complex networks. In the paper, the proposed method is divided into three parts: identifying the central node, the expansion of the community, and the integration of the communities. Firstly, the node's betweenness is used to determine influence of each node in the network and the initial number of communities is determined according to the average betweenness of nodes in the network. Secondly, the remaining nodes use the structural similarity between nodes to expand into the community corresponding to the central node. Finally, according to the number of nodes contained in the community, we divide the community into

large communities and small communities. Then, we integrate small communities into large communities according to the increase in modularity in the initially formed communities. The specific process is shown in Figure 1.

1.1. Basic Principle. A community is a structure that exists in a complex network. In a community, the links of nodes are denser, and the links between communities are relatively sparse. Nodes influence each other in a complex network, but there is usually one central node in the community, which is conducive to the management and operation of the community [12]. Based on the above ideas, we use the betweenness of nodes [14] to find the central node in the community. The remaining nodes need to be filled into the community where the central node is located. We use structural similarity to fill other nodes, and the initial community structure has been discovered. Finally, it is the process of integrating the community. In this paper, the initial community is classified into the large communities and small communities according to the number of nodes in the community [15]. If the number of nodes in the community is greater than a certain threshold, the community is called the large community; otherwise, it is called the small community. This article uses modularity to integrate small communities.

1.2. Contributions. Many classic community detection algorithms have been proposed. They are suitable for different types of networks and need to set parameters, etc. Compared with existing community detection algorithms, the algorithm in this paper has the following contributions:

- (1) We use node betweenness to determine the number of communities and the initial nodes in the community.
- (2) The expansion of the community is achieved by comparing the structural similarity between the remaining nodes and all the nodes in the community.
- (3) We divide the initial community into large communities and small communities according to the number of nodes. The modularity is used to integrate small communities into large communities.

The structure of the paper is as follows: in Section 3, the detailed introduction is given to the algorithm proposed in this article, including some basic concepts and detailed steps of the algorithm. In Section 4, we evaluate communities based on standardized mutual information and modularity, and we used artificial networks and some realistic networks to test the detected communities. In Section 5, the algorithm proposed in this paper is further summarized and discussed.

2. Related Work

In the past few decades, scholars have proposed many community detection methods in complex networks, such as label propagation algorithm [16], the method based on

module optimization [17], spectral clustering method [18], the random walking method [19], and splitting hierarchical clustering method [20].

The label propagation algorithm is an algorithm proposed by Raghavan et al. [16] in 2007. The label propagation algorithm uses the label information of known nodes to predict the label information of unknown nodes. In the process of the label propagation algorithm, each node will randomly generate a label to represent its community at the beginning. In each process of predicting a node, the label of each node is predicted based on the label of its neighbor node. Labels with the same node will belong to the same community, and labels of different nodes will belong to different communities. In terms of calculation time, the running time of the label propagation algorithm is linear, but result of the label propagation algorithm is unstable.

Newman [17] proposed modularity in 2004, which is a function of the difference between the actual number of linked edges in the community and its random expectations. The higher the modularity, the better the community finds the algorithm, and many scholars use modularity as a target function to detect the community structure. Modularity optimization is a difficult problem for NP, and many scholars try to reduce its difficulty by various methods [21, 22]. Based on the Newman [17], Newman [23] (2006) redefined the modularity and adopted a spectral method to solve the problem of optimizing the target function. However, for the existing modularity optimization method, the community discovery effect is not ideal as the complexity of the network becomes greater.

Mcsherry [18] proposes a method based on graph segmentation. In the view of the low expected value of the adjacency matrix, the spectral clustering is proposed to solve the problems of dichotomy, graph coloring, and small groups. Coja-Oghlan and Lanka [24] use spectral clustering methods, respectively, and solve the problem of community detection in sparse networks where the number of nodes increases but the node degree is bounded and networks where the node degree obeys a power-law distribution. Bshouty and Gentile [25] propose a spectral clustering algorithm to solve the community detection problem of the popular planting partition model. Rinaldo and Lei [26] proposed a method which is simple and easy to implement and has theoretical foundation support, so it is one of the most commonly used community detection methods. However, classic spectral clustering is not ideal when dealing with sparse networks. The spectral clustering relies on the similarity matrix, and final effect of different similarity matrices will be different.

The concept of random walking first appeared in Pearson [19], he described random walking, and Rayleigh further extended it. Random walking has been widely used in many research fields and successfully introduced to community detection. The basic principle of random walk is that due to the dense degree of connectivity in the community, random walkers usually walk in the community for a long time. Therefore, a random roamer which is starting from a node will arrive at a node in the same community after a few steps.

FIGURE 1: The flowchart of this proposed method.

The other methods based on random walks are easily extended to community detection in weighted networks, which is one of their advantages. Rosvall and Bergstrom [27] proposed the Infomap method based on information theory. The algorithm uses random walkers as the agent of information dissemination in the network. The performance of Infomap is extremely good, and it has been recognized and used by many scholars. However, the random walk algorithm has randomness, and it is very sensitive to the selection of initial nodes. It also has a great impact on community detection.

For hierarchical clustering algorithms, Girvan and Newman [28] calculate the betweenness of each edge in the network graph. However, the above algorithms still have problems such as long detection time and unstable detection results.

Therefore, Brandes [14] proposes a method of using node betweenness to detect the central node of the network and expand the remaining nodes to the community where the central node is located through structural similarity.

3. Community Detection Based on Node Betweenness

3.1. Preliminaries. We engage in unweighted, undirected networks in the paper. Let the $G = (V, E)$ to be a graph, where $V = \{v_1, v_2, \dots, v_n\}$ represents the nodes, $E = \{e_1, e_2, \dots, e_m\}$ represents the edges, n and m , respectively, represent the number of nodes and edges in the network.

Definition 1. Node betweenness: it reflects the influence of a node in the global network. The higher the node betweenness of a node, the more node pairs passing through the node and the more likely it is to be in the center of the community. Assuming a node v is a node in the network, the node betweenness of this node is calculated as follows [14]:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}, \quad (1)$$

where $\sigma_{st}(v)$ represents the number of shortest paths $s \rightarrow t$ through node v and σ_{st} represents the number of shortest paths $s \rightarrow t$. For a graph in a network, if the betweenness of the node is in an unweighted graph, the weight of the undirected graph is assigned a value of 1 when calculating the betweenness of the node.

Definition 2. Structural similarity: the structural similarity of the nodes reflects the similarity of the two nodes in the graph network structure [12]:

$$\text{Sim}(x, y) = \frac{2 * |N(x) \cap N(y)|}{|N(x)| + |N(y)|}, \quad (2)$$

where x and y are nodes in the graph G , and $N(x)$ and $N(y)$, respectively, represent the set of nodes of the direct neighbors of node x and node y .

Definition 3. Average node betweenness: it reflects the average level of each node betweenness, it is used to determine the number of central initial community, and the average betweenness is calculated as follows:

$$\text{Average_betweenness} = \frac{1}{n} \sum_{i=1}^n C_B(v_i), \quad (3)$$

where n represents the number of all nodes in the entire network graph and $C_B(v_i)$ represents the betweenness of node v_i . The average betweenness is used to find large communities and small communities among the initial community results of the network.

Definition 4. Modularity: when the number of network communities is unknown, modularity is an indicator used to reflect the result of the detection of the community, and its calculation is as follows [12]:

$$Q = \frac{1}{2m} \sum_{i \neq j} \left(a_{ij} - \frac{d(v_i)d(v_j)}{2m} \right) \delta(C_i, C_j), \quad (4)$$

where $d(v_i)$ represents the degree of node v_i and $A_{ij} = \{a_{ij}\}$ represents the adjacency matrix of the graph; if $a_{ij} = 1$, node v_i and node v_j are connected, and if $a_{ij} = 0$, it means that node v_i and node v_j are not connected. $\delta(C_i, C_j)$ is an indicator function whose value is 0 or 1. If $\delta(C_i, C_j) = 0$, then $C_i \neq C_j$, and if $\delta(C_i, C_j) = 1$, then $C_i = C_j$. And m represents the number of edges in the network. Modularity can be used to detect the quality of community detection in a certain detection process.

3.2. Proposed Algorithm. The algorithm consists of three steps: identifying the central node, expanding the community, and integrating the community.

In Step 1, we use node betweenness to search for the central node in the network graph. For the nodes $V = \{v_1, v_2, \dots, v_n\}$, we calculate the betweenness of nodes by formula (1), and the nodes are $V' = \{v_{j1}, v_{j2}, \dots, v_{jn}\}$, where $C_B(v_{j1}) > C_B(v_{j2}), \dots > C_B(v_{jn})$. Average_Betweenness is the average betweenness, and C_{int} is the set where the center node is located. For the node set $\{v_{j1}, v_{j2}, \dots, v_{jn}\}$ sorted by node betweenness, if $C_B(v_{ji}) > \text{Average_Betweenness}$ ($i \in 1, \dots, n$), $C_{\text{int}} = \{\} \cup \{v_{ji}\}$.

The pseudo code is shown in Algorithm 1.

In Step 2, we believed the similarity among nodes in the same community is larger than the others. Similarity could ensure that the core community consisted of nodes with high similarity and dense connection [12]. The expansion of the community adopts the method of maximizing structural similarity. The choice of structural similarity is discussed in detail in Section 5.2. The expansion of the community is shown by the following formula [12]:

$$\max_{v_i \in C_{\text{int}} \text{ and } v_j \in \text{Remain_nodes}} \text{sim}(v_i, v_j), \quad (5)$$

where v_i and v_j represent nodes in the network and C_{int} is the set of nodes in all communities.

In Step 3, for the initially formed community $C = \{C_1, C_2, \dots, C_k\}$, the following cases may occur. First, when there is only one node in the initial community, the node must be the central node of a certain initial community. The reason is that the similarity between central

node and the nodes in Remain_codes is less than that of other central nodes and the Remain_codes. Therefore, none of the remaining nodes have joined the community where this node is located. Second, after integrating the communities where a single node is located, there will still be cases where there are fewer nodes in some communities. In the case of a large network, communities will still exist in large numbers. It seriously affects the quality of community discovery. It is needed to further integrate small communities.

Hu et al. [2] adopted an evaluation method using the number of nodes in the community. He took the average number of nodes in the community as the metric for small communities, as follows:

$$S = \frac{1}{k} \sum_{i=1}^k |C_i|, \quad (6)$$

where $|C_i|$ represents the number of nodes in each initial community, k represents the number of initial communities, and S is the average number of nodes in each initial community.

First, we should search for communities with nodes less than S in the community and integrate the communities. Modularity is introduced as an evaluation index to measure the quality of network community structure and to merge communities. The higher the value of modularity Q , the better the community quality. Therefore, it is necessary to find the community distribution of each node in the network so that Q is maximized. In the third stage, when the increase in the modularity of the two communities satisfies (7), merge the two communities C_i and C_j into one community $C_i \cup C_j$, where C_i and $C_j \in C$:

$$\Delta Q(C_i \cup C_j) > \text{Max}\{\Delta Q(C_i \cup C_j), 0\}. \quad (7)$$

Any two communities are merged to ensure that the modularity Q is growing to the largest. Repeat this process until no small communities.

3.3. The Proposed Algorithm and Its Complexity. The community detection algorithm consists of three parts: identifying the central node, expanding the community, and integrating the communities. In the first stage, it generally takes $O(n^2)$ time in sorting node's betweenness and $O(mn + n \log n)$ time in calculating the node betweenness in undirected graph where m is the number of edges and n is the number of nodes. In the second stage, expanding the community, the process of computing the similarity between the neighbor, the time cost is $O(k^2)$, where k is the average degree of the network. And in the last stage, integrating the community begins with the k big communities, and merging the k big communities into t communities, ($t \leq k \leq n$). It takes $O(k^2)$ when the modularity comparison on the combination of k big communities is computed.

In summary, the algorithm's complexity costs $O(n^2 + mn + n \log n + O(k^2)) \approx O(n^2)$. Compared with some classic algorithms, the time complexity of the LPA

algorithm is $O(m)$, and the time complexity of the algorithm in the paper is larger than that of LPA. When the number of nodes is small, the time complexity of the two methods is almost the same. However, the GN algorithm is $O((m^2) * n)$ and the EM algorithm is $O(MN^3)$ where M is the number of iterations and N is the number of parameters, and the time complexity in the paper is relatively smaller than other methods.

3.4. Example. The process of the algorithm is further explained for the dolphin network. The network of football is shown in Figure 2.

First of all, the central node needs to be selected in the network graph. The node betweenness was used to select the central node. The following is central node selected by node betweenness: the central node is as follows: [36, 1, 40, 37, 7, 17, 20, 54, 51, 57, 39, 28, 29, 43, 14, 33, 13, 38, 23, 45].

The second step is to arrange the remaining nodes in descending order of the betweenness of the nodes based on the detected community center nodes as follows: [50, 15, 30, 32, 34, 6, 27, 42, 41, 47, 8, 9, 59, 52, 0, 10, 18, 61, 19, 21, 44, 2, 24, 5, 26, 16, 46, 3, 25, 53, 49, 55, 56, 4, 11, 12, 22, 31, 35, 60, 58, 48]. We calculate the similarity between the node and each central node and then assign nodes to the community where the central node with the greatest similarity is located. The final detection of the initial communities is shown in Table 1.

Regarding the results of the initial communities, it can be observed that some communities are composed of one node and some communities are composed of multiple nodes. The initial community is divided into large communities and small communities according to the average number of nodes in the community. If the number of nodes in the initial community is greater than the average number of nodes, it is defined the large community; it is defined the small community on the contrary. For small communities, they consist of one or multiple nodes, the nodes in the small communities are attributed to the large communities by modularity. Finally, the number of communities was determined and the small communities were integrated. The following shows the division of small communities and large communities, as shown in Tables 2 and 3.

In the process of integrating small communities into large communities, we calculate the modularity of the small communities and the large communities separately. If the modularity increases, the small community will be attributed to the larger communities; if multiple modularity increments are greater than 0, the community with the largest modularity is selected to join. For example, the community C_1 is integrated with each community in the large communities to calculate the increase in modularity, as shown in Table 4.

For the above table, when C_1 is integrated with C_{12} , C_{14} , C_{17} , and C_{20} , respectively, it is found that there are two

FIGURE 2: Figure of the network of the football dataset.

integration cases where the modularity value is greater than 0. In this case, the node with the largest modularity increment is selected (C_1 and C_{20}). The final division of the community is shown in Table 5, and the visualization result is shown in Figure 3.

4. Experimental Results

Through the experiment of various algorithms, we compare the performance of various algorithms in the synthetic network and real network. The algorithm in the paper is implemented by PyCharm and NetworkX of Python.

4.1. Evaluation Metrics. To compare the quality of different community detection algorithms, we used NMI and ARI to evaluate the results.

4.1.1. Normalized Mutual Information (NMI). NMI is a measure of similarity between two communities. The calculation method is as follows:

$$NMI = \frac{-2 \sum_{i=1}^{C_X} \sum_{j=1}^{C_Y} N_{ij} \log(N_{ij} / N_i N_j)}{\sum_{i=1}^{C_X} N_i \log(N_i / N) + \sum_{j=1}^{C_Y} N_j \log(N_j / N)}, \quad (8)$$

where X represents the real partition, Y is the partition found by the algorithm, C_X is the number of actual communities, C_Y is the number of communities found, N is the number of nodes in the network, N_{ij} is the number of nodes shared by the real community i in partition X and the found community j in partition Y , N_i represents the sum of the i -th row of matrix N_{ij} , and N_j represents the sum of the j -th column of matrix N_{ij} .

4.1.2. Adjusted Rand Index (ARI). ARI is another indicator that compares two regions and is defined as follows:

Input: Graph G , Node set V , Link set E .

Big_community: the set of big scall community, Small_community: the set of small scall community.

C_{int} : the node that their betweenness is over Average_betweenness.

Output: the detected communities C .

Step 1: identifying influent node

- (1) Ranking the node by their node betweenness decreasing. $V' = \{v_{j1}, v_{j2}, \dots, v_{jn}\}$, where $C_B(v_{j1}) > C_B(v_{j2}), \dots > C_B(v_{jn})$.
- (2) Find the node that their betweenness is over Average_Betweenness. $C_{int} = \{v_{j1}, v_{j2}, \dots, v_{jk}\} (k < n)$. And remaining nodes are $Remain_nodes = \{v_{jk+1}, v_{jk+2}, \dots, v_{jn}\}$.

Step 2: expanding the community

- (1) Calculate the degree of similarity between the Remain_nodes and C_{int} , and attribute the remaining nodes to the community where the highest similarity node is located.
- (2) The initial community $C = \{C_1, C_2, \dots, C_k\}$ is formed.

Step 3: integrating the community

- (a) Suppose the S is the average number of nodes in the community.
 - For i in k :
 - If $|C_i| > S$, then the community $|C_i|$ is a big scall community;
 - Big_community = $\{C_1, C_2, \dots, C_m\} (m \leq k)$
 - If $|C_i| < S$; then the community $|C_i|$ is a small scall community;
 - Small_community = $\{C_1, C_2, \dots, C_n\} (n \leq k)$
 - (b) for C_i in Big_community:
 - for C_j in Small_community:
 - calculate the $\Delta Q(C_i \cup C_j)$;
 - if $\Delta Q(C_i \cup C_j)$ is the max;
 - then $C_i = C_i \cup C_j$
- until Small_community is None
- Return C

ALGORITHM 1: The proposed algorithm.

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left\{ \left[\sum_i \binom{a_i}{2} \sum_i \binom{b_i}{2} \right] / \binom{n}{2} \right\}}{(1/2) \left[\sum_i \binom{a_i}{2} + \sum_i \binom{b_i}{2} \right] - \left\{ \left[\sum_i \binom{a_i}{2} \sum_i \binom{b_i}{2} \right] / \binom{n}{2} \right\}}, \quad (9)$$

where n_{ij} , a_i , and b_j are the values in the association table. ARI evolved from the Rand Index (RI), and its value range is $[-1, 1]$. The various values of ARI are described in detail in the literature.

4.2. Results and Discussion of Synthetic Networks. In the experiment, we generated several synthetic networks with different characteristics to compare the performance of community detection algorithms. To keep the synthetic network consistent with the actual network, we use the LFR reference network, where the network characteristics are easy to control. Due to the controlled node degree distribution and community size distribution, they exhibit similar properties to real networks. In the LFR network, different networks can be generated by changing the parameters. One parameter that needs to be emphasized is μ , which is a mixed parameter. Each node shares a small part of $1-\mu$ link with other nodes in its community and shares a small part of μ with other nodes in the network.

Generally speaking, the LFR network has five structural parameters: the number of network nodes n , the average

degree k , the maximum and minimum scales of the community maxc and minc, and the mixed parameter μ . Among all the parameters, the mixed parameter μ is the proportion of the links connecting each node in the community to the nodes in other communities in the total number of nodes.

To verify the proposed algorithm, the hybrid parameter μ and the number of nodes of the network are adjusted to verify the proposed algorithm. Firstly, the influence of mixing parameters on various algorithms is verified. The number of fixed nodes is 1000, the size of the community is between 10 and 20, the average degree $k = 5$, and the mixing parameters are adjusted to between 0.1 and 0.8.

The results of the algorithm are shown in Figure 4. Figure 4(a) shows the result of NMI, and Figure 4(b) shows the result of ARI. As shown in Figure 4(a), as the mixing parameter μ increases, the NMI value of each algorithm decreases. As the mixing parameter μ increases, the structure of the artificial generation network (LFR) becomes more complicated and the community structure becomes fuzzy.

In Figure 4, the value of the normalized mutual information (NMI) of various algorithms changes with the mixing parameter μ . It can be known from Figure 4 that as

TABLE 1: The initial communities of Dolphin network.

Community	Node's index of community
C_1	36, 8
C_2	1, 25, 22, 31
C_3	40
C_4	37, 3
C_5	7, 19
C_6	17, 32, 26
C_7	20
C_8	54, 30, 27
C_9	51, 15
C_{10}	57
C_{11}	39, 48
C_{12}	28, 42, 47, 10, 2
C_{13}	29
C_{14}	43, 34, 52, 0, 61, 16, 46, 53, 12, 35, 58
C_{15}	14

TABLE 2: Small communities of Dolphin network.

Small community	Node's index of community
C_1	36, 8
C_2	1, 25, 22, 31
C_3	40
C_4	37, 3
C_5	7, 19
C_6	17, 32, 26
C_7	20
C_8	54, 30, 27
C_9	51, 15
C_{10}	57
C_{11}	39, 48
C_{13}	29
C_{15}	14
C_{16}	33, 44, 99
C_{18}	38, 50
C_{19}	23, 59
C_{21}	63, 97

TABLE 3: Large communities of Dolphin network.

Large community	Node's index of community
C_1	36, 8
C_2	1, 25, 22, 31
C_6	17, 32, 26
C_{21}	63, 97

TABLE 4: The modularity increasing of integration communities.

Integration communities	Node's index of community
C_1 and C_{12}	-0.0003
C_1 and C_{14}	-0.0093
C_1 and C_{17}	0.0012
C_1 and C_{20}	0.0060

the value of μ increases, the ability of various algorithms to discover the structure of the network community decreases. As the mixing parameters become larger, the structure of the artificial network community will become more complex.

FIGURE 3: The final result of community.

In Figure 4(a), when $0.1 < \mu < 0.3$, the NMI value of the algorithm in this paper and the Louvain is larger than other algorithms. The NMI of GN algorithm, LPA algorithm, and spectral clustering is lower, and EM's NMI is the worst. When $0.3 < \mu < 0.5$, the Walktrap's NMI is the biggest; the LPA and our method are almost the same. However, NMI of GN and spectral clustering is better EM's algorithm. When $\mu > 0.5$, the value of NMI about our algorithm is better than other six algorithms. As a result, it shows that the proposed method has a good result in the process of increasing μ .

Figure 4(b) shows the Adjusted Rand Index (ARI) of each algorithm with the increasing of the mixing parameter μ . It can be seen from Figure 4(b) that when $0.1 < \mu < 0.3$, the Louvain has best showing than other methods. The ARI value of the algorithm in the paper is almost the same as the ARI value of the GN algorithm, and the ARI values of other methods are lower than the algorithm in the paper and GN algorithm. When $0.3 < \mu < 0.5$, the Walktrap has better result than other methods. The ARI of the method in paper, GN, are fluctuating. But the ARI of the method in paper occupies the main position. After the mixing parameter $\mu > 0.5$, the ARI value of the method in the paper is larger than other algorithms.

When the mixing parameters increase, the value of NMI and ARI of the algorithm in the paper is the largest when $\mu > 0.5$. It is better for the method in the paper than other methods.

In Figure 5, the mixed parameter of the fixed network is $\mu = 0.5$, and the size of the community is $\min = 10$ and $\max = 20$. As the number of nodes increases, the tendency in the NMI and ARI of various algorithms is shown in Figures 5(a) and 5(b), respectively.

In Figure 5(a), when the number of nodes is increasing, the NMI of the method in the paper and the LPA keeps rising, when the node's number is increased from 1000 to 9000. When $1000 < n < 3000$, the value of NMI of the

FIGURE 4: Results of different algorithms on synthetic networks with varying μ . (a) Results of NMI. (b) Results of ARI.

FIGURE 5: Results of different algorithms on synthetic networks with node number. (a) Results of NMI. (b) Results of ARI. 1k = 1000.

TABLE 5: The final communities of Dolphin network.

Final result	Nodes of final community
C_1	28, 42, 47, 10, 2
C_2	43, 34, 52, 0, 61, 16, 46, 53, 12, 35, 58, 40, 7, 19, 14, 33, 44, 49, 38, 50
C_3	13, 6, 41, 9, 5, 56, 60, 1, 25, 22, 31, 17, 32, 26, 54, 30, 27, 57, 39, 48
C_4	45, 18, 21, 24, 55, 4, 11, 36, 8, 37, 3, 20, 51, 15, 29, 23, 59

proposed algorithm is the best. However, the LPA's NMI is almost the same as our algorithm. Louvain and Walktrap have the same tendency. The EM's NMI is the worst. When $3000 < n < 6000$, the LPA's NMI is the best, and the proposed algorithm and GN algorithm are the almost same. The Walktrap's NMI has decreasing tendency, and the Louvain keeps the previous tendency. When $n > 6000$, our method's NMI is the best. The performance of other methods keeps previous tendency.

In Figure 5(b), it can be seen that the proposed algorithm and the GN have great advantages. As the number of nodes increases, when $1000 < n < 5000$, the value of ARI of Louvain has the best performance, and the proposed algorithm's NMI is bigger than the GN algorithm. When $5000 < n < 7000$, the ARI of the proposed method is also the best, and the difference of other algorithms' ARI behaves similarly. When $n > 7000$, all six methods have almost the same performance.

When the number of nodes increases, the NMI of the proposed method is better than other methods.

4.3. Results and Discussion of Real-World Network. We also evaluated the effect of the algorithm on some typical real networks. The network data are logged in "<http://www.personal.umich.edu/~mejn/netdata/>" and "<https://github.com/zzz24512653/CommunityDetection/tree/master/network>." The details of these networks are shown in Table 6. $|V|$ represents the number of nodes of the network, $|E|$ represents the number of edges of the network, k represents the average degree of network, β_{th} represents epidemic threshold, AC represents average clustering coefficient, Assortative represents assortative coefficient, and DH represents degree heterogeneity. The community results of the algorithm in the paper are shown in Tables 7–10. The modularity of real networks is shown in Table 11 and Figure 6.

The Karate network is a well-known real social network, and it reflects the differences between club management and coaches through the friendship between community members. The method strictly divides the club into two communities, and most nodes can be divided correctly. The modularity of the proposed algorithm and the EM is higher, the modularity of the other algorithms is relatively low, and community detection results are shown in Figure 7.

The nodes of the Dolphin network represent dolphins, and the edges represent the relationship between dolphins. From the perspective of modularity, the proposed algorithm and the spectral clustering (SC) have higher modularity. However, modularity of the LPA, EM, and GN is much small. The community structure obtained by the proposed algorithm is shown in Figure 8.

TABLE 6: The details of four networks.

Datasets	$ V $	$ E $	k	β_{th}	AC	Assortativity	DH
Karate	34	78	4.588	0.129	0.571	-0.475	0.385
Dolphin	62	159	5.129	0.147	0.259	-0.043	0.325
Football	115	613	10.611	0.093	0.403	0.162	0.040
Pollbooks	105	441	8.4	0.084	0.488	-0.1279	0.328

TABLE 7: The final result of Karate by our method.

Community	Karate
C_1	8, 27, 25, 29, 28, 9, 33, 2, 31, 23, 24, 26
C_2	13, 19, 5, 6, 30, 3, 4, 10, 7, 11, 12, 17, 21, 16, 14, 15, 18, 20, 22, 0, 32, 1

TABLE 8: The final result of Dolphin by our method.

Community	Dolphin
C_1	28, 42, 47, 10, 2
C_2	43, 34, 52, 0, 61, 16, 46, 53, 12, 35, 58, 40, 7, 19, 14, 33, 44, 49, 38, 50
C_3	13, 6, 41, 9, 5, 56, 60, 1, 25, 22, 31, 17, 32, 26, 54, 30, 27, 57, 39, 48
C_4	45, 18, 21, 24, 55, 4, 11, 36, 8, 37, 3, 20, 51, 15, 29, 23, 59

TABLE 9: The final result of Football by our method.

Community	Football
C_1	20, 27, 56, 113, 17, 70, 96, 95, 87, 62, 65, 76
C_2	21, 111, 22, 78, 8, 108, 51, 7, 77, 68
C_3	61, 71, 99, 18, 31, 34, 54, 38, 36, 14, 15, 106, 100, 64, 39, 6, 13, 26, 12, 42, 60, 47, 32, 2, 43, 85
C_4	91, 75, 66, 57, 58, 59, 69, 92, 24, 63, 97, 28, 11, 90, 48, 86, 112, 44, 50
C_5	102, 81, 5, 74, 84, 82, 3, 10, 72, 98, 107, 52, 40
C_6	30, 29, 79, 94, 80, 101, 55, 35, 19
C_7	110, 67, 46, 73, 88, 114, 53, 49, 83
C_8	93, 9, 4, 23, 0, 16, 41, 104
C_9	103, 45, 37, 109, 105, 25, 1, 33, 89

The Football network comes from the American football match in the fall of 2000 [28]. The network uses nodes to represent teams and edges represent matches between teams. In this network, the modularity of the proposed method and the spectral clustering is much high, while the modularity of the remaining methods is highly low. The community structure is shown in Figure 9.

The Pollbooks network is a network of American political books. The nodes represent American political books, and the edges represent two books purchased by the same

TABLE 10: The final result of Pollbooks by our method.

Community	Pollbooks
C_1	49, 57, 43, 48, 44, 32
C_2	12, 35, 39, 37, 36, 38, 33, 55, 42, 34, 16, 46, 17, 19, 15, 10, 47
C_3	14, 22, 24, 50, 41, 27, 23, 25, 45, 54, 21, 9, 3, 8, 53, 40, 11, 20, 13, 26, 56
C_4	84, 93, 100, 95, 101, 81, 96, 97, 89, 66, 73, 102, 86, 94
C_5	6, 5, 29, 105, 2, 1, 18, 7, 4
C_6	77, 83, 79, 82, 98, 78, 92, 30
C_7	99, 60, 62, 61, 59, 63
C_8	71, 75, 74, 91, 87, 88, 72, 76, 70, 80, 31
C_9	64, 85, 65, 28, 104, 90, 103, 58, 67, 51, 52, 69, 68

TABLE 11: The modularity by seven methods in four networks.

Algorithm	Karate	Dolphin	Football	Pollbooks
Proposed algorithm	0.220	0.434	0.585	0.396
EM	0.278	0.069	0.187	0.351
LPA	0.218	0.086	0.135	0.257
GN	0.122	0.105	0.197	0.308
Spectral clustering	0.190	0.403	0.600	0.451
Walktrap	0.302	0.401	0.580	0.500
Louvain	0.395	0.518	0.604	0.526

FIGURE 6: Modularity results for our algorithm in the networks with ground truth.

FIGURE 7: Communities of Karate by the proposed algorithm.

FIGURE 8: Communities of Dolphin by the proposed algorithm.

buyer. In the network, the modularity of the proposed method is less than the modularity of the spectral clustering. Other methods have lower modularity. The community structure is shown in Figure 10.

FIGURE 9: Communities of Football by the proposed algorithm.

FIGURE 10: Communities of Pollbooks by the proposed algorithm.

TABLE 12: Six common similarity measures.

Measures	Definition	Measures	Definition
Jaccard	$ \Gamma(v_i) \cap \Gamma(v_j) / \Gamma(v_i) \cup \Gamma(v_j) $	PA	$k(v_i) \times k(v_j)$
Sørensen	$2 \Gamma(v_i) \cap \Gamma(v_j) / (k(v_i) + k(v_j))$	RA	$\sum_{z \in \Gamma(v_i) \cap \Gamma(v_j)} 1/k(z)$
LHN	$ \Gamma(v_i) \cap \Gamma(v_j) / (k(v_i) \times k(v_j))$	CN	$ \Gamma(v_i) \cap \Gamma(v_j) $

FIGURE 11: Modularity results for the six similarity algorithms.

TABLE 13: The modularity of different similarities in four networks.

Data	Sørensen	CN	Jaccard	LHN	PA	RA
Karate	0.304	0.306	0.201	0.201	0.033	0.306
Dolphin	0.434	0.439	0.434	0.433	0.080	0.359
Football	0.585	0.547	0.585	0.537	0.076	0.505
Pollbooks	0.396	0.317	0.396	0.321	0.057	0.307

5. Discussion and Conclusions

The proposed algorithm is composed of three steps: identifying the central node, expanding the community, and integrating the initial community. The proposed algorithm is mainly compared with six algorithms: GN, label propagation (LPA), EM, spectral clustering, Louvain, and Walktrap. Among them, the GN, label propagation (LPA), spectral clustering, Louvain, and Walktrap are commonly used methods for community detection. The EM is not a commonly used algorithm for community detection, but it is usually used for clustering in machine learning. In this part, the selection and conclusion of similarity are in Section 5.1.

5.1. Similarity Comparison. In the second step of this algorithm, the choice of similarity has much impact on community. Therefore, different similarity measures are selected to test the different results of proposed algorithm. It is listed common similarity measures in Table 12. In Figure 10, PA and RA have the worst performance, because it is usually used to quantify the functional importance of edges. It is more suitable for dynamic networks. Compared with other similarities, Sørensen similarity gives better results. Based on the above results, our algorithm uses Sørensen

similarity to detect similar nodes (see Figure 11 and Table 13).

5.2. Conclusion. Community detection is a much significant research topic in social network's analysis. The goal of community detection is to divide the network into several communities, and each node belongs to only one community. However, when μ increases, the NMI value drops rapidly in existing algorithms. In the paper, we divide the process of community detection into three stages, and we use the method that expands the central node to detect communities. We test our methods on several synthetic and real datasets and compare the results with the previous methods. The experimental results show that our method is more efficient. And as μ increases, the NMI value of our method is higher than other methods.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the Northeastern University Industry-University-Research Strategic Cooperation Project (code: 71971050).

References

- [1] H. J. Li, "Network science and dynamical networks," *ICSES Interdisciplinary Transactions on Cloud Computing IoT Big Data (IITCIB)*, vol. 2, no. 3, pp. 1–3, 2018.
- [2] K. Hu, J.-B. Hu, L. Tang et al., "Predicting disease-related genes by path structure and community structure in protein-protein networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2018, no. 10, p. 100001, 2018.
- [3] H.-J. Li, Z. Bu, A. Li, Z. Liu, and Y. Shi, "Fast and accurate mining the community structure: integrating center locating and membership optimization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2349–2362, 2016.
- [4] A. V. Goltsev, S. N. Dorogovtsev, J. G. Oliveira, and J. F. Mendes, "Localization and spreading of diseases in complex networks," *Physical Review Letters*, vol. 109, no. 12, pp. 2733–2737, 2012.
- [5] Q. Xuan, Z.-Y. Zhang, C. Fu, H.-X. Hu, and V. Filkov, "Social synchrony on complex networks," *IEEE Transactions on Cybernetics*, vol. 48, no. 5, pp. 1420–1431, 2018.

- [6] H. J. Li and J. J. Daniels, "Social significance of community structure: statistical view," *Physical Review E*, vol. 91, p. 012801, 2015.
- [7] P. Chong and B. Shuai, "Measure of hazardous materials transportation network invulnerability based on complex network," *Journal of Central South University*, vol. 45, no. 5, pp. 1715–1723, 2014.
- [8] H.-J. Li and L. Wang, "Multi-scale asynchronous belief percolation model on multiplex networks," *New Journal of Physics*, vol. 21, no. 1, p. 015005, 2019.
- [9] H.-J. Li, Z. Bu, Z. Wang, J. Cao, and Y. Shi, "Enhance the performance of network computation by a tunable weighting strategy," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 3, pp. 214–223, 2018.
- [10] Z. Sun, B. Wang, J. Sheng, Y. Hu, Y. Wang, and J. Shao, "Identifying influential nodes in complex networks based on weighted formal concept analysis," *IEEE Access*, vol. 5, pp. 3777–3789, 2017.
- [11] Z. Wei, "Overview of complex networks community structure and its applications in electric power network analysis," *Proceedings of the Chinese Society of Electrical Engineering*, vol. 35, no. 7, pp. 1567–1577, 2015.
- [12] X. You, Y. Ma, and Z. Liu, "A three-stage algorithm on community detection in social networks," *Knowledge-Based Systems*, vol. 187, 2019.
- [13] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.
- [14] U. Brandes, "A faster algorithm for betweenness centrality," *The Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163–177, 2001.
- [15] A. Bouyer and H. Roghani, "A fast and robust local community detection starting from low degree nodes in social networks," *Future Generation Computer Systems*, vol. 113, 2020.
- [16] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physics Review E*, vol. 76, Article ID 036106, 2007.
- [17] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6 Pt 2, Article ID 066133, 2004.
- [18] F. Mcsherry, "Spectral partitioning of Random graphs," in *Proceedings of the 2001 42nd IEEE Symposium on Foundations of Computer Science*, October 2001.
- [19] K. Pearson, "The problem of the random walk," *Nature*, vol. 72, no. 1867, p. 342, 1905.
- [20] X. F. Wang and Y. B. Liu, "Overview of algorithms for detecting community structure in complex networks," *Journal of University of Electronic Science and Technology of China*, vol. 384, no. 2, pp. 667–674, 2009.
- [21] J. Jia, X. Xiao, B. Liu, and L. Jiao, "Bagging-based spectral clustering ensemble selection," *Pattern Recognition Letters*, vol. 32, no. 10, pp. 1456–1467, 2011.
- [22] R. Guimerà and L. A. Nunes Amaral, "Functional cartography of complex metabolic networks," *Nature*, vol. 433, pp. 895–900, 2005.
- [23] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577–8582, 2006.
- [24] A. Coja-Oghlan and A. Lanka, "Finding planted partitions in random graphs with general degree distributions," *Siam Journal on Discrete Mathematics*, vol. 23, no. 4, pp. 1682–1714, 2010.
- [25] N. H. Bshouty and C. Gentile, "Learning theory," *Multi-View Regression Via Canonical Correlation Analysis*, Springer, New York, NY, USA, pp. 82–96, 2007.
- [26] A. Rinaldo and J. Lei, "Consistency of spectral clustering in stochastic block models," *Annals of Statistics an Official Journal of the Institute of Mathematical Statistics*, vol. 43, no. 1, pp. 215–237, 2015.
- [27] M. Rosvall and C. Bergstrom, "Maps of information flow reveal community structure in complex networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 4, pp. 1118–1123, 2007.
- [28] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.