Research Article

Social Network Community-Discovery Algorithm Based on a Balance Factor

Lizhao Liu,¹ Xiaomei Shu,² and Biao Cai³

¹School of Mechanical and Electrical Engineering, Chengdu University of Technology, Chengdu 610059, China

²College of Computer Science, Sichuan University, Chengdu 610041, China

³College of Computer Science and Cyber Security, Chengdu University of Technology, Chengdu 610059, China

Correspondence should be addressed to Biao Cai; caibiao@cdut.edu.cn

Received 25 June 2021; Revised 13 November 2021; Accepted 5 January 2022; Published 8 February 2022

Academic Editor: Aura Reggiani

Copyright © 2022 Lizhao Liu 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 discovery plays a crucial role in understanding the structure of networks. In recent years, the application of clustering algorithms in the community-discovery tasks of complex networks has been studied frequently. In this study, we proposed a balance factor of node density and node-degree centrality for the core-node selection problem in community discovery. We also proposed a new community-discovery algorithm based on the balance factor, adaptability, and modularity increment, which is based on the balance factor (BComd). First, the proposed method was able to identify the core nodes in a community. Second, we used node-degree centrality, node density, and adaptability to detect overlaps between communities and then we removed these overlaps from the network to obtain a subnetwork with a clear community structure. Third, we obtained the preliminary community divisions by clustering the subnetworks, and these preliminary communities were usually the core parts of the communities they belonged to. Finally, each preliminary community was compressed into a new node, and then, the new network was clustered using the Louvain algorithm. The experimental results showed that the algorithm identified the core nodes in communities well, effectively discovered overlaps between communities, and had superior performance in large-scale networks.

1. Introduction

Complex-network analysis has been applied in many fields, such as text analysis [1, 2], user identification [3], and recommendation systems [4]. The key attribute of complex networks is their community structure, an object of great interest that has been studied extensively in various works [5, 6]. Research on complex-network community-discovery algorithms holds great significance in analyzing the hierarchical structure of complex networks, the formation of communities, the spreading nature of nodes, and the mining of their internal-network characteristics.

Many community-detection algorithms have been proposed for complex networks [7–9]. The Girvan-Newman (GN) [10] algorithm is a classical community-discovery algorithm based on hierarchical clustering. The basic idea is to detect the edges connecting different communities in a graph and remove them to achieve segmentation. In a graph,

the edges connecting different communities are usually those with large edge-betweenness. Edge-betweenness is the proportion of the number of shortest paths passing through the edge to the total number of shortest paths in the graph. The time complexity of GN algorithms is $O(mn^2)$, where n is the number of nodes and m is the number of edges. Because of its high complexity, the algorithm often is not directly applicable to large-scale networks. The k-means clustering algorithm [11] divides data into multiple clusters based on the minimum error function. This algorithm has the characteristics of fast clustering, easy implementation, and effective classification in large datasets and is widely used for community detection in complex networks. However, the initial clustering-center selection of the traditional k-means clustering algorithm is a random process and cannot guarantee effective clustering [12]. And CNN [35, 36] based community-detection method [33] can make good performance.

In response to the problems of the noted algorithms, we have proposed a balance factor for determining the core nodes in the community. This balance factor can select reasonable nodes as initial clustering centers, thus avoiding the problem of ineffective clustering caused by randomly selecting the initial centers. On the basis of this balance factor, we proposed the BComd algorithm. The time complexity of the algorithm is $O(n\bar{k})$, where n is the number of nodes in the network and $\bar{k}$ is the average of the node degrees in the network. The BComd algorithm had a considerably lower time complexity than the GN algorithm. In summary, the proposed method achieved better experimental results than traditional tests using several commonly used publicly available large network datasets.

This paper is organized as follows: Section 2 introduces several theories. In Section 3, the proposed method is described in detail, and then the experimental results are introduced in Section 4. Finally, the conclusion is summarized in Section 5.

2. Methodology

2.1. Notions. In this section, the concepts and methods used by the algorithm are introduced separately. The node density and degree are used to construct the balance factor, and the modularity measure factor and fitness factor are used for community clustering.

2.1.1. Node Density. We determined the node density [13] by the number of edges and nodes in the subgraph generated by that node's h-hop forward width-first search. The higher the node density, the stronger the community belongingness of the node, which is inconsistent with traditional degree centrality; the leaf nodes have the smallest degree but have the largest community belongingness-1. The node density in the network is defined as follows:

$$\text{density}(v_i^h) = \frac{|E'|}{(|V'|(|V'| - 1)/2)}, \quad (1)$$

where i represents the i th node, h is the number of forward hops from the node, G' represents the subgraph generated by the h-hop width-priority search of the node V' , V' is the node set of the subgraph, $|V'|$ represents the number of nodes V' in the subgraph, E' represents the edge set of the subgraph G' , and $|E'|$ refers to the number of middle edges in E' . Node density is directly related to the location of nodes in the network. Nodes with low node density are usually closely related to other communities. In contrast, nodes with high node density generally are not related to other communities generally.

2.1.2. Modularity and Modularity Increment. Modularity is a measurement concept proposed by Newman and Girvan [10] and is a quality index for evaluating network partitions. The range of modularity is $[-1/2, 1]$ [23], and it is the difference between the number of connected edges of nodes in

the community and the number of connected edges under random conditions. The greater the modularity, the better the effect of community division. Modularity is defined as follows:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(C_i, C_j), \quad (2)$$

where m is the number of edges of the network, and A_{ij} is the element of the i th row and j -th column of collocation matrix A corresponding to the network. If there is a connecting edge between nodes i and j , A_{ij} is 1; otherwise, it is 0. Additionally, k_i is the degree of node i , so $k_i k_j / 2m$ indicates the possibility of connected edges between node i and node j in a random situation. C_i is the community to which node i belongs, and when the communities to which node i and node j belong are the same, $\delta(C_i, C_j) = 1$; otherwise, it is 0. Let e_{ij} be the ratio of the number of edges connecting the vertices in community i to the vertices in community j to the total number of edges. Then it can be expressed with the following equation:

$$e_{ij} = \frac{1}{2m} \sum_{vw} A_{vw} \delta(C_v, i). \quad (3)$$

Let $a_i = \sum_j e_{ij}$, which represents the ratio of the number of edges associated with nodes to the total number of edges in the community i , where

$$a_i = \frac{1}{2m} \sum_v k_v \delta(C_v, i). \quad (4)$$

The modularity can also be represented as follows:

$$Q = \sum_i (e_{ii} - a_i^2). \quad (5)$$

Newman also proposed the equation for ΔQ [14], the change of merging the community i and j , which is as follows:

$$\Delta Q = e_{ij} + e_{ji} - 2a_i a_j = 2(e_{ij} - a_i a_j). \quad (6)$$

2.1.3. Adaptability. Lancichinetti et al. [15] defined an adaptability equation, as follows:

$$f_G = \frac{k_{in}^G}{(k_{in}^G + k_{ex}^G)^\alpha}, \quad (7)$$

where k_{in}^G and k_{ex}^G are the sum of the total internal degree and the total external degree of module G , respectively, and α is the resolution parameter that controls the community's size. A natural choice for α is 1. Formula (7) gives the ratio of the internal degree of the community to the total degree, which corresponds to the so-called weak definition of the community introduced by Radicchi et al. [16]. Given the adaptability function, the adaptability of node A corresponds to module g . Then f_g^A is defined as the change in the adaptability of the module with and without node A , which is as follows:

$$f_g^A = f_{g+\{A\}} - f_{g-\{A\}}, \quad (8)$$

where the symbol $g + \{A\}$ (or $g - \{A\}$) represents the subgraph obtained from module g , where node A is located inside (outside). In this study, α was set to 1.

2.2. Proposed Algorithm. We constructed the balance factor based on the partitioning of the degree and density of the graph. The degrees in a graph showed a long-tailed distribution, and it was found in [13] that the density also generally followed a long-tailed distribution. It has been verified that nodes with small density tend to be the cross nodes between communities [13], so we constructed the balance factor of community belongingness based on the characteristics of density and degree.

This section introduces the algorithm proposed in this paper. The balance factor based on node density and node-degree centrality is given in detail in Section 2.2.1, which identifies the core nodes in the community. In Section 2.2.2, the balanced distribution of nodes in the network is introduced. Experiments found that the fitting curves of the balanced distribution of nodes in the Amazon, DBLP, and YouTube networks all had a similar parabolic shape. According to this finding, we proposed a method to detect overlapping parts in the network. Section 2.2.3 explains this community-detection method (i.e., the BComd algorithm), which is based on the balance factor, adaptability, and modularity increment.

2.2.1. Balance Factor Construction. The center of a network is especially important because it affects the entire network's operation, so choosing the right core node improves the accuracy of community detection. Degree centrality is a direct metric to portray the centrality of nodes, which reflects the importance of nodes in the network. Node density reflects the community affiliation of nodes, and nodes with low node density tend to establish close connections with other communities. In contrast, nodes with high node density do not establish connections with other communities. In this paper, node density was combined with node-degree centrality as the criteria for selecting core nodes. Node-degree centrality is defined as the number of links incident to the node: $C d(i) = k_i$, where k_i denotes the degree of the node v_i . Balance factor B is then defined as follows:

$$B(v_i) = \frac{\alpha C d(i)}{K} + (1 - \alpha) \text{Density}(v_i^1), \quad (i = 1, 2, \dots, n), \quad (9)$$

where the value of K is equal to the maximum degree in the network, n is the number of nodes in the network, and $\alpha = 0.4$. Nodes located at the center of the cluster shared many edges with other partners in the group. Therefore, the core nodes should have a high degree of centrality and aggregation. According to equation (9), the nodes with high $B(v_i)$ values in the network satisfied this requirement. Therefore, we choose the node with the highest $B(v_i)$ value as the core

node. According to equation (9), the node-degree centrality had a greater influence on selecting core nodes than node density. Selecting nodes with large values as the initial nodes of the community can reduce the negative impact caused by the community's uncertainty to which the seed nodes belong. For instance, if a structural hole in the network were randomly selected as the initial node of the community, then during the community expansion phase, the added nodes would have a greater chance of coming from different communities.

Figure 1 shows the balance factor of the nodes in the Karate network. The size of the node represents its $B(v_i)$ value, so the larger the node, the larger the $B(v_i)$ value. The $B(v_i)$ values of nodes 1 and 34 were the largest, and these two nodes also happened to be the core nodes of these two real communities. Choosing these as the initial nodes of the two communities would lead to better clustering results. Conversely, if node 10 was randomly selected as the initial node of the community, it is highly likely to be added into node 3 and node 34 when the community is expanded, and these two nodes would belong to different communities. In summary, our proposed $B(v_i)$ value helped choose a reasonable node as the initial node. Choosing such a node could reduce the negative impact caused by the uncertainty of the community to which the seed node belongs, thus improving the clustering accuracy of the algorithm.

2.2.2. Balanced Distribution of Nodes. If the overlapping nodes between communities could be detected and these nodes and their edges could be removed from the network, the community structure would be more apparent (see the proof in Section 4.3). We clustered the subnetworks obtained by removing the overlaps to obtain the initial community divisions, usually the core parts of the communities they belonged to. We compressed each primary community into one node and then clustered the new network using the Louvain algorithm to improve the clustering accuracy. However, we did not know the overlap between communities for unknown networks, so we used the following method to detect the overlap.

We used $\log(k_i - \min k) / \log(\max k - \min k)$ as the x coordinate and $B(v_i) / \max B$ as the y coordinate, where $\max k$ ($\min k$) was the maximum (minimum) degree of the network and $\max B$ was the maximum $B(v_i)$ value in the network. All nodes in the network were output on the coordinate system. Figure 2 shows the balanced distribution of nodes on the Amazon, DBLP, and YouTube datasets; these datasets were derived from the Stanford Large Network Dataset Collection (SNAP) [17].

Because one horizontal coordinate corresponded to multiple vertical coordinates in the image, we took the average of the vertical coordinates and performed a curve fit using a cubic function. As shown by the red parabola in Figure 3, we found that the fitted curve had a similar parabolic shape.

We used half of the average density of nodes in the network as the boundary. Nodes with a density higher than half of the average were considered to have higher node

FIGURE 1: The balance factor of nodes in the Karate network.

FIGURE 2: The balance distribution of nodes.

FIGURE 3: Nodes with high node density (with red color).

density. These are the nodes denoted by red nodes in Figure 3.

Next, a line was drawn perpendicular to the x axis through the lowest point of the parabola. We assumed the node on the right side of the line had a high degree of centrality, so we changed the node's color to the right of the line to red, as shown in Figure 4. The blue nodes in Figure 4 were considered to have low density and degree centrality, so these nodes were more likely to be strongly connected to multiple communities. The chance was greater that they were in the overlapping parts between communities.

To further find the overlap between communities, we defined the set of nodes v_i and its neighbors as a subgraph G_{v_i} . We set $\alpha = 1$, and the adaptability of the subgraph G_{v_i} was

the ratio of the internal degree of the subgraph G_{v_i} to the total degree. If G_{v_i} had high adaptability, this indicated dense internal connectivity and sparse external connectivity, which meant that the neighbors of the node v_i were close to each other. In contrast, if G_{v_i} had low adaptability, it meant that the neighbors of the node v_i were sparsely connected and might belong to different communities. Therefore, if the adaptability of G_{v_i} was low, we considered that node v_i tended to be closely connected with other communities. For the adaptability of the subgraph G_{v_i} , a threshold value δ was set.

Our algorithm divided the nodes in the network into two groups. Nodes located in the red area and with $f_{G_{v_i}} > \delta$ were in group 1, whereas nodes located in the blue area or with

FIGURE 4: Red and blue partition results.

$f_{G_{v_i}} < \delta$ were in group 2. Nodes in group 2 and the edges linked to them were removed from the network to obtain a subnet with a clearer community structure. We set the threshold δ to three-fourths of the average $f_{G_{v_i}}$ value. This approach effectively detected overlaps between communities and removed the overlaps before clustering, which gave better community-detection results, as demonstrated in Section 4.3.

2.2.3. BComd Algorithm. The algorithm started with the core node s and had two sets: community C and the node set W adjacent to community C (each node in W has at least one neighbor in community C). We selected the neighbor node with the largest ΔQ from W in each step and calculated its adaptability. If the adaptability of the neighbor node was positive, we added it to community C and updated W to include the newly discovered neighbor nodes. If the adaptability of the neighbor was negative, it was deleted from W . This process was continued until W was empty. To facilitate the description of the algorithm, we assigned the following definitions.

Definition 1. $G = (V, E)$ was defined as an undirected unweighted network, where V was the vertex set, and E was the edge set. Note that $|V| = n$ and $|E| = m$.

Definition 2. $N_G(s)$ was defined as the set of neighbor nodes of node s .

Definition 3. $N_g(s)$ was defined as the set of neighbor nodes of node s in group 1.

2.2.4. Community Integration. Shang et al. [18] proposed a community-membership function. Considering that a given network $G = (V, E)$ and a prepartition $SC(G) = \{SC_1, SC_2, \dots, SC_w\}$, then SC_k ($k = 1, 2, \dots, w$) are the subcommunities in the network, where w is the number of subcommunities and $\Gamma(SC_k)$ is the set of nodes adjacent to subcommunity SC_k . This is an aggregation of the external nodes connected to SC_k , as follows:

$$f_{CM}(SC_i, SC_j) = \frac{|\Gamma(SC_i) \cap SC_j|}{|\Gamma(SC_i)|}, \quad (10)$$

where $\Gamma(SC_i) \cap SC_j$ represents the connectivity between subcommunities SC_i and SC_j , and $|\Gamma(SC_i)|$ represents the number of neighbor nodes connected to SC_i . Here $f_{CM}(SC_i, SC_j)$ evaluates the degree of closeness between SC_i and SC_j . Shang et al. [18] also defined a function represented by f_{CI} to calculate the degree of mutual membership between two communities:

$$f_{CI}(SC_i, SC_j) = f_{CM}(SC_i, SC_j) + f_{CM}(SC_j, SC_i), \quad (11)$$

($i, j = 1, 2, \dots, W$).

This function is called the mutual membership function. First, we set the merge threshold δ . Then, for community C , f_{CI} for all its neighboring communities was calculated, and the community with the largest f_{CI} was selected. If $f_{CI} > \delta$, then both communities were merged.

2.3. Complexity Analysis. In this section, we analyze the time complexity of each stage. During the calculation of $B(v_i)$, $B(v_i)$ for each node should be calculated. We defined the average degree of nodes in the network as $\bar{k}$ and the total number of nodes in the network as n . Then the time complexity of this process was $O(n\bar{k})$. The time complexity of the sorting operation was $O(n \log n)$. In the grouping strategy, the time complexity was caused by removing nodes and their edges from the network. We defined the number of nodes to be deleted as $|b_2|$. For each node to be deleted, the number of edges related to it was equal to its degree, so the time complexity was $O(|b_2|\bar{k})$. In the process of community expansion, the BComd algorithm processed only one node at each step, and the time complexity of processing each node was almost linear, so the time complexity was $O(n)$. Because, in most cases, $\bar{k} > \log n$, the complexity of the BComd algorithm was $O(n\bar{k} + n \log n + |b_2|\bar{k} + n) = O(n\bar{k})$.

3. Experiment Setting

This section briefly introduces the network used in the experiment and the two evaluation metrics used in this study. It was experimentally demonstrated that eliminating overlapping parts obtained a clearer community structure and better community-detection results. Last, we compared the proposed algorithm with known algorithms. All experiments were performed on a PC equipped with a 2.70 GHz Intel Core processor and 4 GB RAM.

Input: Network $G(V, E)$

Output: All Communities

Step 1: Calculate the $B(v_i)$ of all nodes in the network. Sort the nodes in V in descending order to obtain sequence l .

Step 2: Use the method in Section 3.2 to divide the nodes in the network into two groups. Delete the nodes in group 2 and the edges linked to them from the network; then delete the nodes in group 2 from sequence l .

Step 3: Select the first node s in l as the core node and remove it from sequence l . At this point, there is only node s in community C . Add $N_g(s)$ to W .

Step 4: For each node in W , calculate its ΔQ for the scenario in which it joins community C .

Step 5: Select neighbor v with the maximum ΔQ from W and calculate its adaptability. a . If the adaptability is positive, add node v to community C and remove it from both W and sequence l . $W = W \cup N_g(v) - (w \cap C)$. For node z , which was previously in W , we do not need to recalculate its ΔQ but simply update it according to the following equations:

$$\begin{cases} \Delta Q = \Delta Q + 1/m - 2Cd(v)/2m \times (z)/2m, & v \text{ is adjacent to } z, \\ \Delta Q = \Delta Q - 2(Cd(v)/2m) \times (Cd(z)/2m), & \text{otherwise.} \end{cases}$$

For those nodes newly added to W , calculate their ΔQ . b . If the adaptability is negative, remove node from W .

Repeat Step 5 until there are no nodes in W . At this point, the detection of community C is finished.

Step 6: Go back to Step 3 and repeat the following steps until there are no nodes in sequence l .

Step 7: Add the nodes and edges deleted in Step 2 back to the network. At this point, the network consists of two parts: one with the previously deleted nodes and the other with the multiple communities that have been detected. We treated each previously deleted node as a community with only one node.

Step 8: Compress each community into a single node. The link weights between the new nodes are derived from the sum of the link weights between the nodes in the corresponding two communities. The links between nodes in the same community lead to a self-loop of that community in the new network. The new network is clustered using the Louvain algorithm.

Step 9: Execute the Community Integration Strategy.

Step 10: Export all communities.

ALGORITHM 1: BComd Algorithm 1

3.1. Dataset. The Karate Club [19] is a social network composed of 34 nodes and 78 edges. Each node represents a member of the club, and an edge indicates that two members are friends. The bottlenose dolphin community (Dolphin) [20] is a social network of frequent interactions between 62 dolphins living in New Zealand. The nodes in the network represent dolphins, and an edge represents the corresponding frequent contacts of dolphins. There are two communities in the network. American college football [21] is a network with 115 nodes and 616 edges. The nodes in the network represent the football teams, and an edge between two nodes represents a game between the two teams. American political books [22, 23] is a social network about American political books. Each node in the network represents a book. An edge between two books indicates that they are often purchased together, and the network has three communities, as shown in Table 1.

Amazon is the network of the products sold on Amazon.com, the nodes in the network refer to products, and an edge between two nodes indicates that they are often purchased together. The network has 334,863 nodes and 925,872 edges. DBLP is a coauthorship network in which the nodes in the network refer to authors, and an edge between two authors indicates that they have published at least one paper together. Last, YouTube is an online social network. The Amazon dataset, DBLP dataset, and YouTube dataset were all provided by Stanford University SNAP (<https://snap.stanford.edu/data/index.html#onlinecoms>).

3.2. Evaluation Metric. We compared BComd with other algorithms using the following two evaluation metrics:

Given a network $G = (V, E)$, let T be the set of real communities and D be the set of communities detected by the community-detection algorithm. Each real community $T_i \in T$ (or each detected community $D_i \in D$) is a set of member nodes. The average F1 parametric is a popular metric for evaluating the similarity between two sets, and when applied to community detection, it can be formed as follows [24]:

$$F1(T, D) = \frac{1}{2} \left(\frac{1}{|T|} \sum_{T_i \in T} F(T_i, D) + \frac{1}{|D|} \sum_{D_j \in D} F(D_j, T) \right), \quad (12)$$

where

$$F(T_i, D) = \max_{D_j \in D} F1(T_i, D_j). \quad (13)$$

$F1(T_i, D_j)$ is the harmonic mean of precision and recall, and formula $F(D_i, T)$ can be expressed in the same way.

Normalized mutual information (NMI) [15] is an important evaluation metric for community detection that can measure how similar the community-segmentation results of the algorithm are to the real results. Assume that vectors X and Y denote two specific segmentation results of the network, where the i th bit of each vector indicates the class to which the i th point belongs. The larger the mutual information of X and Y , the more the information that can be provided to the other, and the more similar they are:

$$I(X, Y) = H(X) - H(X|Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right), \quad (14)$$

TABLE 1: Real networks.

Network	No. of nodes	No. of edges	No. of communities
Karate	34	78	2
Dolphin	62	159	2
Football	115	613	12
Political books	105	441	3
Amazon	334863	925872	75149
DBLP	317080	1049866	13477
YouTube	1134890	2987624	16386

where $p(x, y)$ denotes the joint-distribution probability of X and Y . Adjusting the mutual information to a scale between zero and one is called standardized mutual information, and the expression is as follows:

$$U(X, Y) = \frac{2I(X, Y)}{H(X) + H(Y)}. \quad (15)$$

NMI ranges from zero to one, and the larger the value, the better the algorithm performance.

4. Experimental Results and Analysis

4.1. Experimental Results. In this study, we compared the proposed algorithm with the FG algorithm [25], GN algorithm [26], k-means algorithm [11], sparse linear-coding method (SLC) [27], equation (13), algorithm [28], and MIGA algorithm [29]. The results are given in Table 2. The BComd algorithm was compared with each reference algorithm using the F1 and NMI metrics. The results showed that the proposed method performed well, especially on the first two datasets for which the community structure was clear. For both the Karate Club and Dolphin networks, the values of F1 and NMI obtained by our method were 1.0, which means that the community structure of both networks detected by our method was consistent with the actual structure. The MIGA algorithm achieved the best NMI and F1 values for the Football dataset. For the American political books, the proposed method obtained the maximum value of NMI, whereas the best value of F1 was obtained by equation (20) algorithm. In summary, the BComd algorithm performed well on networks with clear community structures, but it did not work sufficiently well on small networks with complex community structures.

In terms of the effect of the large graph, we compared the results of the proposed algorithm with the LPA algorithm [30], WLPA algorithm [24], Louvain algorithm [31], and WLouvain algorithm [24]. The measurement method was to run each community-detection algorithm on the Amazon dataset, DBLP dataset, and YouTube dataset. Then we compared the running results (all communities) with the top 5000 real communities on each network. We observed that BComd had the best performance. The proposed algorithm obtained the highest F1 and NMI values in all three datasets, especially on the YouTube network. It was significantly better than the other algorithms. Therefore, the BComd algorithm was superior to the other algorithms on large networks, as shown in Table 3.

4.2. Proof of Effectiveness of Eliminating Overlaps. The idea of the BComd algorithm was to detect overlapping nodes between communities and delete these nodes and their edges from the network to obtain a clearer community-structure subnet. The subnetworks were then clustered to obtain primary-community divisions, and these primary communities were usually the core of the community they belonged to. Each primary community was compressed into one node, and then the new network was clustered using the Louvain algorithm to improve the clustering accuracy.

This section demonstrates experimentally that a more explicit community structure and better community-detection results could be obtained by eliminating overlaps. The idea of the proof is as follows:

- (1) Use the BComd algorithm introduced in Section 2.2.2 including Step 2 of the algorithm, which eliminates the overlaps and then performs community detection to get result ①
- (2) Use the BComd algorithm introduced in Section 3.3 while omitting Step 2 to perform the community-detection directly without eliminating the overlaps to obtain result ②
- (3) Compare results ① and ②. If the results of ① are better than ② on multiple networks, eliminating the overlaps can lead to a clearer community structure and better community detection

This section was run on three large networks: the Amazon, DBLP, and YouTube datasets. The community-detection results were compared with ground-truth communities to get their F1 and NMI values. The experimental results obtained are shown in Table 4. Table 4 shows that the effect of ① was better on the three networks. Therefore, the proposed method effectively found the overlaps between communities. It also demonstrated that eliminating overlaps can yield a clearer community structure and better community-detection results.

4.3. Experimental Result Analysis. From the experimental results, we found that the algorithm proposed in this paper performed well in large-scale networks. We believe there were two reasons for this. First, the descending balance factor we designed selected more reasonable nodes as the seed nodes for community expansion. These seed nodes that satisfied both high node density and high centrality were more likely to be the central or edge nodes of the community. Such initial nodes reduced the negative impact

TABLE 2: Experimental results of the community-discovery algorithms on small graphs (F1 and NMI).

Dataset	Criterion	FG	GN	k-means	Equation (20)	SLC	MIGA	BComd
Karate	F1	0.971	0.970	0.879	1.0	0.971	1.0	1.0
	NMI	0.837	0.836	0.666	1.0	0.837	1.0	1.0
Dolphin	F1	0.937	0.980	0.770	0.961	0.980	0.965	1.0
	NMI	0.652	0.890	0.417	0.814	0.890	0.814	1.0
Football	F1	0.528	0.802	0.730	0.859	0.846	0.864	0.842
	NMI	0.697	0.878	0.822	0.865	0.793	0.916	0.847
Political books	F1	0.725	0.808	0.655	0.829	0.798	0.797	0.782
	NMI	0.568	0.568	0.454	0.597	0.584	0.585	0.637

The bold values show best performance.

TABLE 3: Experimental results of the community-discovery algorithms on large graphs (F1 and NMI).

Dataset	Criterion	LPA	WLPA	Louvain	WLouvain	BComd
Amazon	F1	0.45	0.45	0.463	0.459	0.566
	NMI	0.425	0.413	0.43	0.424	0.535
DBLP	F1	0.39	0.39	0.39	0.391	0.493
	NMI	0.249	0.255	0.249	0.268	0.370
YouTube	F1	0.043	0.144	0.043	0.161	0.440
	NMI	0.016	0.032	0.016	0.051	0.234

The bold values show best performance.

TABLE 4: Experimental results demonstrating the effectiveness of eliminating overlaps (F1 and NMI).

Dataset	Criterion	①	②
Amazon	F1	0.588	0.481
	NMI	0.363	0.234
DBLP	F1	0.490	0.329
	NMI	0.272	0.142
YouTube	F1	0.491	0.117
	NMI	0.157	0.018

The bold values show best performance.

caused by the uncertainty of the community to which the seed node belonged, thereby improving the accuracy of the algorithm. As can be seen from Figure 5, nodes with strong ties to more than one community had smaller $B(v_i)$ values, whereas those at the center of the community had larger $B(v_i)$ values. This further demonstrated that the balance factor described in this paper can find core nodes in a network.

Additionally, we used degree centrality, node density, and adaptability to find overlaps between communities, and we removed these overlaps from the network to obtain a subnet with a clear community structure. Our experiments found that the subnet had many groups. In the initial community delineation obtained by clustering the subnet, many communities had an adaptability of one, indicating that the subnet had a clear community structure with strong connections within communities and sparse connections between communities. It also showed that the overlaps we were looking for were correct. After clustering the subnets to obtain the preliminary community structure, we compressed each preliminary community into a node. We then used the Louvain algorithm to cluster the new network, thereby

improving the accuracy of community detection. As demonstrated in Section 4.3, eliminating overlapping parts could yield subnets with clearer community structure and better community-detection results. However, our algorithm was less effective in small-scale networks because the balanced distribution of nodes in a small graph did not satisfy a parabolic distribution, so removing any overlapping nodes from the small graph made it impossible to obtain subnets with a clear community structure. This led to a decrease in clustering accuracy. For networks with clear community structure, such as the Karate Club and Dolphin networks, our algorithm achieved 100% accuracy (see Figure 5). When the community structure was complex, however, it was impossible to obtain subgraphs with clear community structure by removing nodes, such as Football and American political books. Figure 6 shows that the clustering accuracy of the method was low.

4.4. Running-Time Analysis. Table 5 shows the average running time spent by each method on the three large networks with the units in seconds. The LPA, Louvain, and BComd algorithms were all written in *Python*. All experiments were conducted on a PC with a 2.70 GHz Intel Core processor and 4 GB RAM. The running times are shown in Table 5. We identified a difference in running time between the BComd algorithm and Louvain algorithm at the stages of $B(v_i)$ calculation and clustering subnets. The proposed algorithm was slower than Louvain's algorithm on the Amazon and DBLP datasets but was faster on the YouTube dataset.

If the size of the subnet was the same as the original network, the algorithm in this paper ran longer than the Louvain algorithm because the BComd algorithm needed to calculate the $B(v_i)$ value for each node, which took more

FIGURE 5: The balance factor of each node in the Karate Club and Dolphin databases.

FIGURE 6: The balance distribution of nodes in a small graph.

TABLE 5: Running time (unit: s).

Dataset	LPA (s)	Louvain (s)	BComd (s)
Amazon	425.072	153.292	867.161
DBLP	428.598	557.458	792.686
YouTube	523.304	19057.285	5432.796

The bold values show best performance.

time. However, when the algorithm deleted many nodes and edges, the generated subnet was much smaller than the original network. Clustering such subnets reduced the running time. After the initial community division was obtained by subnet clustering, each community was compressed into one node. A community that contained multiple edges and multiple nodes was compressed into one node. The previously deleted nodes and edges were added back. The new network was much smaller than the original network, and the running time was further reduced for the new network clustering. If the deleted nodes were a fixed percentage of the total network nodes, then the number of deleted nodes and edges was greater for a larger network, and the advantage of running the subnet was greater. However, the ratio of deleted nodes to the total nodes of the network was not a fixed value; rather, it was related to the network's structure and the size of the overlap we detected. Nevertheless, we still found that the larger the network, the greater the advantage of running the subnet, and when such

an advantage was large enough to offset the extra time spent on the $B(v_i)$ computation, the BComd algorithm was faster than the Louvain algorithm. Although it ran slower than the Louvain algorithm on the Amazon network and DBLP network, our algorithm also ran faster than the Louvain algorithm on the YouTube dataset.

5. Conclusion

We proposed a balanced factor measure based on node-degree centrality and node density, representing the community belongingness of nodes. Accordingly, we proposed a balanced factor-based algorithm for social network community detection. The algorithm could select more reasonable nodes as seed nodes for community expansion, effectively determining the overlapping nodes, and also deleted overlapping nodes in the initial community division to obtain better community division results. The final experiments also verified the effectiveness of the proposed method. The next step in our research should

focus on the practical optimization of the balance factor-based community-detection algorithm applied to industrial super-large datasets. The computational complexity of the balance factor of the algorithm was low, and the main complexity was reflected in the clustering process. Therefore, the parallel optimization of the clustering process should be the focus of future research.

Data Availability

The data of this paper are available from the following hyperlink: <http://snap.stanford.edu/data/index.html#onlinecoms>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially funded by NSFC (Grant 61701049), the Soft Science Fund of Sichuan Province (Grant 2019JDR0117), and the Digital Media Science Innovation Team of CDUT (Grant 10912-kytd201510).

References

- [1] C. Akimushkin, D. R. Amancio Jr, and O. N. Oliveira, "Text authorship identified using the dynamics of word co-occurrence networks," *PLoS One*, vol. 12, no. 1, Article ID e0170527, 2017.
- [2] D. R. Amancio, "Comparing the topological properties of real and artificially generated scientific manuscripts," *Scientometrics*, vol. 105, no. 3, pp. 1763–1779, 2015.
- [3] Y. Li, Z. Su, J. Yang, and C. Gao, "Exploiting similarities of user friendship networks across social networks for user identification," *Information Sciences*, vol. 506, pp. 78–98, 2020.
- [4] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 2, pp. 357–370, 2019.
- [5] S. Ahajjam, M. El Haddad, and H. Badir, "A new scalable leader-community detection approach for community detection in social networks," *Social Networks*, vol. 54, pp. 41–49, 2018.
- [6] W. Li, S. Jiang, and Q. Jin, "Overlap community detection using spectral algorithm based on node convergence degree," *Future Generation Computer Systems*, vol. 79, pp. 408–416, 2018.
- [7] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.
- [8] C. Liu, Y. Du, and J. Lei, "A SOM-based membrane optimization algorithm for community detection," *Entropy*, vol. 21, no. 5, Article ID 533, 2019.
- [9] S. Rani and M. Mehrotra, "Community detection in social networks: literature review," *Journal of Information and Knowledge Management*, vol. 18, no. 2, Article ID 1950019, 2019.
- [10] M. E. 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.
- [11] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, Berkeley, CA, USA, July 1965.
- [12] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert Systems with Applications*, vol. 40, no. 1, pp. 200–210, 2013.
- [13] C. Biao and L. Zeng, "Community detection method based on node density, degree centrality, and K-means clustering in complex network," *Entropy*, vol. 21, no. 12, Article ID 1145, 2019.
- [14] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 2, Article ID 066133, 2004.
- [15] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, vol. 11, no. 3, Article ID 033015, 2009.
- [16] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proceedings of the National Academy of Sciences*, vol. 101, no. 9, pp. 2658–2663, 2004.
- [17] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford Large Network Collection," 2019, <http://snap.stanford.edu/data/index.html#onlinecoms>.
- [18] R. Shang, H. Liu, L. Jiao, and A. M. GhalamzanEsfahani, "Community mining using three closely joint techniques based on community mutual membership and refinement strategy," *Applied Soft Computing*, vol. 61, 2017.
- [19] 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.
- [20] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.
- [21] J. Q. Jiang and L. J. McQuay, "Modularity functions maximization with nonnegative relaxation facilitates community detection in networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 391, no. 3, pp. 854–865, 2012.
- [22] M. M. Newman, "Newman's network data collection," 2019, <http://www-personal.umich.edu/~mejn/netdata>.
- [23] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.
- [24] Y. Hu and B. Yang, "Characterizing the structure of large real networks to improve community detection," *Neural Computing & Applications*, vol. 28, no. 8, pp. 2321–2333, 2017.
- [25] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 70, pp. 264–277, 2010.
- [26] 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.
- [27] A. Mahmood and M. Small, "Subspace based network community detection using sparse linear coding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 801–812, 2016.
- [28] B. Tian and W. Li, "Community detection method based on mixed-norm sparse subspace clustering," *Neurocomputing*, vol. 275, pp. 2150–2161, 2018.

- [29] R. Shang, J. Bai, L. Jiao, and C. Jin, "Community detection based on modularity and an improved genetic algorithm," *Physica A: Statistical Mechanics and Its Applications*, vol. 392, no. 5, pp. 1215–1231, 2013.
- [30] 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, Article ID 036106, 2007.
- [31] V. D. Blondel, J. J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 10, 2008.
- [32] B. Cai, Y. Wang, L. Zeng, Y. Hu, and H. Li, "Edge classification based on Convolutional Neural Networks for community detection in complex network," *Physica A: Statistical Mechanics and Its Applications*, vol. 556, 2020.
- [33] L. Wang, X. Yuan, M. Zong et al., "Multi-cue based four-stream 3D ResNets for video-based action recognition," *Information Sciences*, vol. 575, pp. 654–665, 2021.
- [34] Y. Liu, X. Yuan, X. Jiang et al., "Dilated Adversarial U-Net Network for automatic gross tumor volume segmentation of nasopharyngeal carcinoma," *Applied Soft Computing*, vol. 111, Article ID 107722, 2021.