Research Article

Effectively Detecting Communities by Adjusting Initial Structure via Cores

Mei Chen, Zhichong Yang, Xiaofang Wen, Mingwei Leng, Mei Zhang, and Ming Li

1 School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China
2 College of Educational Science and Technology, Northwest Minzu University, Lanzhou 730030, China

Correspondence should be addressed to Mei Chen; mei.chen.lzjtu@hotmail.com

Received 1 May 2019; Revised 29 September 2019; Accepted 4 October 2019; Published 3 November 2019

Copyright © 2019 Mei Chen 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 is helpful to understand useful information in real-world networks by uncovering their natural structures. In this paper, we propose a simple but effective community detection algorithm, called ACC, which needs no heuristic search but has near-linear time complexity. ACC defines a novel similarity which is different from most common similarity definitions by considering not only common neighbors of two adjacent nodes but also their mutual exclusive degree. According to this similarity, ACC groups nodes together to obtain the initial community structure in the first step. In the second step, ACC adjusts the initial community structure according to cores discovered through a new local density which is defined as the influence of a node on its neighbors. The third step expands communities to yield the final community structure. To comprehensively demonstrate the performance of ACC, we compare it with seven representative state-of-the-art community detection algorithms, on small size networks with ground-truth community structures and relatively big-size networks without ground-truth community structures. Experimental results show that ACC outperforms the seven compared algorithms in most cases.

1. Introduction

There are many different kinds of networks in the real world, such as biological networks, ecological networks, social networks, etc. Usually, many real-world networks have intrinsic community structures. A community in a network is always expressed as a group of nodes with dense connections within a community but sparse connections with other communities. Community detection can help us to discover the nontrivial structures and topological features of complex networks.

So far, to resolve the community detection problems in complex networks, various algorithms have been developed based on the widely used concept of similarity measures or certain criterions. The well-known FastQ [1] and spectral clustering [2, 3], and two new algorithms Attractor [4] and ISCD [5] can be regarded as similarity-based methods [6]. FastQ has introduced a fast greedy strategy for modularity maximization. It effectively corresponds to a simple nearest neighbor agglomerative clustering of the network where the adhesion coefficient \( a_{rs} \) is used as a similarity measure. But if the links between two communities connect low-degree nodes, this approach will fail to detect the communities. Spectral clustering models view the community detection as a graph partitioning, which apply spectral analysis to obtain the cut minimization. However, spectral clustering algorithms are not efficient because their running time is cubic in the size of the input dataset, which often limits the usability of these community detection algorithms only to small networks in practical use. Another limitation of spectral clustering algorithms is that they depend on an input parameter \( k \) which is hard to determine. The Attractor converts the edge between two nodes to a distance according to the Jaccard similarity and calculates the gravitation between them. But the accuracy of this method needs to be improved. The ISCD uses a common neighbor similarity between two nodes to propose a new
Complexity also depends on an input parameter evaluation criterion of internal-link compactness, but it also depends on an input parameter \( k \) which is hard to determine.

The criterion-based community detection methods have proposed several criterions, such as modularity [7], betweenness [8], and minimum-cut [9, 10]. But to evaluate the quality of partitioning according to certain criteria often needs to apply optimization methods to detect communities. For example, the most widely used method, modularity maximization, detects communities by searching for one or more divisions with particularly high modularities over possible divisions of a network. Since exhaustively searching over all possible divisions is usually intractable, the time complexity of modularity maximization or minimum-cut in the optimal version of community detection is proved NP-complete [11]. Besides high time cost, the max modularity does not often result in optimum partitions in real networks [12, 13].

Core-based methods [14–16] also play an important role in community detection. Algorithms for finding the cores are efficient and amenable for parallelization [17]. But the lack of ability to distinguish influential nodes is a problem to this method [18].

Different from heuristic methods with certain criterion, we propose a simple, yet effective and fast, similarity-based community detection algorithm named ACC (Adjusting initial Community structure via Cores). ACC can overcome the limitations that many existing community detection methods have. In other words, some of the exiting methods are heuristics with heavy computation in practice but ACC are not.

The most remarkable characteristic of ACC is the ingenious combination of the advantages of similarity-based methods and core-based methods. Based on a naive fundamental assumption that nodes in the same community are more similar to each other than to those in other communities, ACC first groups nodes together according to a novel similarity to obtain the initial community structure. In particular, the similarity proposed in ACC considers not only the number of common neighbors but also the exclusion degree between two adjacent nodes. Then, based on another expression of the assumption that connections of the nodes in the same community are dense while connections of the nodes in different communities are sparse, ACC regards a node with the max local density in an initial community as the core of that community and adjusts the initial communities according to cores.

Our key contributions are as follows. (1) We present a novel similarity which considers not only common neighbors between two adjacent nodes but also their mutual exclusive degree. The threshold of the similarity is easy to be set. (2) We define the influence of a node on its neighbors as its local density, which makes discovering the core in a community much easier. (3) We propose a new community detection algorithm ACC with near-linear time complexity, which can find high-quality communities in different networks.

The remaining sections are organized as follows. We review the related works in Section 2. Then, we give preliminary of ACC in Section 3 and elaborate ACC algorithm in Section 4. After that, in Section 5, we present the performances of ACC on not only networks with ground-truth community structures but also networks without ground-truth community structures, which show how effective our method is, compared to state-of-the-art methods. Finally, Section 6 concludes the work.

2. Related Works

To date, many different methods have been proposed for community identification. We only report some popular methods.

Spectral clustering [3, 19] has become one of the most popular clustering algorithms, and it is currently being used in a wide range of applications. It considers the graph as a similarity matrix and solves a data clustering problem where each cluster is a community. Spectral clustering algorithm gets the top-\( k \) eigenvectors of eigensystem to form an \( n \times k \) matrix. Then, every column of the matrix is the attributes of the corresponding nodes. It groups these \( n \) nodes with \( k \)-Means to get the final community structure. Unfortunately, the running time of spectral clustering algorithms might be cubic in the size of the input dataset, which makes it prohibitive to use this approach on very large datasets.

FastQ [1] algorithm is an agglomerative method that merges nodes into bigger and bigger communities hierarchically, using the modularity criterion. It initializes the network from a state in which each node is a sole member of a community. Then, it repeatedly merges communities pair, which results in the smallest decrease in modularity and ends up with a state at which all nodes in the network are arranged in a community. The result of FastQ can be represented as a dendrogram. Each level of dendrogram indicates a different community structure. FastQ selects the community structure corresponding to the highest modularity value as the final result. But since FastQ is based on the optimization for modularity, the community it has detected is not always corresponding to the ground-truth community in real application.

Newman2006 [20] algorithm first constructs a modularity matrix for networks. Then, it arranges nodes corresponding to the positive element in top eigenvector in a community and other nodes in the opposite community. Newman2006 ends the process when no positive eigenvalue exists.

Louvain method [21] initializes each node as a single community and shifts the community label of each node according to the modularity gain until the labels converge. Then, it considers each community as a node to merge some of them again according to the modularity gain.

Infohiermap algorithm [22] is a flow-based community method. It reveals hierarchical organization by multilevel compression of random walks on networks.

The label propagation (LPA) approach [23, 24] is based on the simple idea that a node should be assigned to the community to which most of its neighbors belong. LPA has been widely concerned. In addition to the advantage of linear time complexity, it does not need to define objective
function and the number of community in advance. But since LPA simply updates the label of a node according to the plurality vote of its neighbors, it suffers from the problem of randomness caused by random update order, which affects the accuracy and stability of the community.

In recent years, considerable efforts have been put in improving the effectiveness and efficiency of community detection [25, 26]. PPC (Personalized PageRank Clustering) [27] employs the inherent cluster exploratory property of random walks to reveal the clusters of a given graph, which combines random walks and modularity to reveal the clusters of a graph. PPC has a linear time and space complexity. Attractor [4] is a community detection algorithm based on distance dynamics. Attractor converts the edges between two nodes to a distance according to the Jaccard similarity and calculates the gravitation between them. The gravitation makes the nodes within one community close to each other and the nodes from different communities far away. MEA-rSN [28] is a multiobjective evolutionary algorithm based on similarity to find communities in signed networks. The two objectives to be optimized are based on the concepts of positive and negative cluster similarity between two neighboring nodes. MHGNMF [29] takes higher-order information among the nodes into consideration to enhance the clustering performance. SUM [30] is a similarity-based method which detects communities by suspecting the maximum degree nodes.

3. Preliminary

In this section, we first prepare the necessary notions about community detection. Then, we define a novel similarity and a new local density. Lastly, we introduce the community detecting strategy used in ACC algorithm.

3.1. Related Notions. Let \( G(V, E) \) be an undirected graph, where \( V \) is the set of nodes, \( E \subseteq V \times V \) is the set of edges. \( e = (u, v) \in E \) indicates a connection between the nodes \( u \) and \( v \). The number of nodes in a graph can be represented as \( n = |V| \), and the number of edges can be represented as \( m = |E| \).

For a node \( u \in V \), the neighbors of node \( u \) are the members of set \( N(u) \) containing its adjacent nodes which share a common incident edge with \( u \): \( N(u) = \{ v \in V \mid (u, v) \in E \} \). The degree of \( u \) is denoted as \( d_u, d_u = |N(u)| \).

In this paper, nodes with links to two or more communities are defined as borders.

3.2. A New Similarity. In general, if two nodes have a number of common neighbors, we believe that the two nodes are similar. For two adjacent nodes \( u, v \in V \), their common neighbors are denoted as \( CN(u, v) \), and \( CN(u, v) = N(u) \cap N(v) \). Therefore, \( |CN(u, v)| \) can be used to measure the similarity degree. If there are too many different neighbors between \( u \) and \( v \), they will be dissimilar. We call this dissimilarity mutual exclusive degree. Suppose that the degree of node \( u \) is smaller than that of \( v \), \( \min(d_u, d_v) - 1 - |CN(u, v)| \) means that the number of neighbors of \( u \) which are not the neighbors of \( v \), where \( \min(d_u, d_v) \) is the smaller degree of nodes \( u \) and \( v \). Therefore, the mutual exclusive degree of two adjacent nodes \( u, v \in V \) can be defined as \( \min(d_u, d_v) - 1 - |CN(u, v)| \).

Given a graph \( G(V, E) \), through the number of the common neighbors and mutual exclusive degree of \( u \) and \( v \), the novel reasonable structural similarity \( \text{sim}(u, v) \) between two adjacent nodes \( u \) and \( v \) is defined as follows:

\[
\text{sim}(u, v) = |CN(u, v)| - \text{Exclusion}(u, v) - |N_1(\text{smallld})|,
\]

where \( N_1(\text{smallld}) \) is the neighbors with degree 1 of the smaller degree node.

3.3. Local Density and Core. We define the influence of a node on its neighbors as its local density and regard the node with the greatest density in a community as a core. For a given network \( G(V, E) \), a node \( u \in V \), \( N(u) = \{ v \in V \mid (u, v) \in E \} \), and \( |N(u)| = d_u = p + q \), where \( p \) is the number of \( u \)'s neighbors which have no common neighbors with \( u \) and \( q \) is the number of \( u \)'s neighbors which have common neighbors with \( u \). We assume that the influence of \( u \) on each of its neighbors is 1. Thus, the total influence of \( u \) on the \( p \) neighbors is \( \sum_{p} 1 = p \). The influence of \( u \) on each of the \( q \) neighbors is weakened by the common neighbors, and the real remaining influence is that 1 subtracts the weakened influence. If we define the weakened influence as the Jaccard similarity, \( \text{Jaccard}(u, v) = |(N(u) \cap N(v))|/(N(u) \cup N(v))| \) [31], then the influence of \( u \) on each of the \( q \) neighbors is \( 1 - \text{Jaccard}(u, v) \). Thus, the total influence of \( u \) on the \( q \) neighbors is \( \sum_{q} (1 - \text{Jaccard}(u, v)) \). The influence of \( u \) on all its neighbors is \( p + \sum_{q} (1 - \text{Jaccard}(u, v)) = p + q - \sum_{q} \text{Jaccard}(u, v) = d_u - \sum_{q} \text{Jaccard}(u, v) = \sum_{v \in N(u)} (1 - \text{Jaccard}(u, v)) \). Therefore, the local density of node \( u \) can be expressed as

\[
\text{den}_i(u) = \sum_{v \in N(u)} (1 - \text{Jaccard}(u, v)).
\]

4. The ACC Algorithm

We introduce the ACC algorithm in this section. First, we describe the process of community detection of ACC in detail. Then, the time complexity of ACC is analyzed.

4.1. The ACC Algorithm. We present our ACC algorithm as follows:

Step 1. Obtain the initial community structure: given a network \( G(V, E) \), for a node \( u \in V \), \( N(u) = \{ v \in V \mid (u, v) \in E \} \), if \( \text{sim}(u, v) > \zeta \), where the \( \zeta \) is the threshold of similarity, we put the two nodes into one community. The number of the initial communities discovered in this step is marked as \( k \).

Step 2. Adjust the initial community structure according to cores: we select the top-\( k \) nodes with the highest local density as cores of communities. ACC
considers that there is at most one core in one community. Thus, if there are more than one core in an initial community, the initial community will be broken up into as many communities as the number of the cores, and each core represents a community. Then, we rebuild new communities by assigning unlabelled neighbors of each core to the core.

**Step 3.** Expand communities: we assign each remaining unlabelled node to the community to which its highest density neighbor belongs. If an unlabelled node does not have a highest density neighbor, it will be regarded as an initial community.

**Step 4.** Merge small communities and reassign borders: for each small community whose size is smaller than 3, we merge it to the community which has most links to the small community. Then, we regard nodes having links to two or more different communities as the borders and reassign each border to the community which has most links to the border. Next, considering that reassigning the borders could produce small communities whose size can be smaller than 3, we merge each small community whose size is smaller than 3 to the community which has most links to it.

4.2. Time Complexity Analysis. To invest the initial communities, the similarity of any two linked nodes in a network is required, and thus the time computation of Step 1 of ACC is \(O(|E|)\). In Step 2, since ACC needs to get the Jaccard distance and local density, the time complexity is \(O(d \cdot |E|)\), where \(d\) is the average degree of nodes in network \(G(V,E)\). During the process of getting the top-\(k\) nodes with the highest local densities, the time complexity is \(O(|V|) + O(k \cdot \log(|V|))\). Moreover, to assign the neighbors of cores to the corresponding core, the time complexity is \(O(k \cdot d)\). Step 3 of ACC takes \(O(q)\) time to assign the unlabelled nodes, where \(q\) is the number of unlabelled nodes, \(q \leq |V|\). To merge the small-size communities into its most links communities, the time complexity is \(O(3 \cdot d)\). The time complexity of reassigning the borders is \(O(|E|)\). In total, the time complexity is \(O(|E| + d \cdot |E| + |V| + k \cdot \log(|V|) + k \cdot d + q + 3 \cdot d)\). Since \(d \ll |E|, q \ll |V|, k \ll |V|\), the time complexity is approximately \(O(|E| + |V| + \log(|V|))\).

5. Experiments and Analysis

In this section, we evaluate our proposed algorithm ACC on real-world networks to demonstrate its benefits.

5.1. Baselines and Benchmarks

5.1.1. Baselines. To evaluate the performance of ACC, we compare it with several representative state-of-the-art community detection algorithms.

Spectral clustering (SC) [2] is one of the most popular community detection algorithms and is currently being used in a wide range of applications. It is based on the graph p-Laplacian.

Newman2006 [20] is a splitting algorithm, which uses the maximum eigenvalue of a matrix and recursively splits a network until the final results are obtained.

Infohiermap [22] is a well-known approach, which requires a strong information-theoretic background and reveals multilevel structures in networks.

FastQ [1] is a bottom-up algorithm based on optimizing, which selects the result from the tree where the Modularity \(Q\) is the maximum.

LPA [23] is a fast community detection algorithm with a linear time complexity, which is based on label propagation.

PPC [27] is an efficient graph clustering algorithm, which employs random walks to detect communities.

Attractor [4] is one of the current most popular community detection algorithms, which is based on distance dynamics.

ACC is implemented in Python 2.7 environment. FastQ and LPA are obtained from igraph which is Python module, and their running environment is also Python 2.7. The source codes of Spectral clustering, Newman2006, Infohiermap, PPC, and Attractor are provided by their authors.

5.1.2. Benchmarks. We evaluate the performances of different community detection algorithms on real-world networks, including small-size networks with class information and relatively large size networks without class information. The basic statistical information of the networks is listed in Table 1. In all the networks used in this paper, \(\zeta\) of ACC algorithm is 0, except Polbooks network in which \(\zeta\) of ACC is −1.

Karate [8, 32] is a famous network, derived by Zachary’s observation on a karate club. It is composed of trainees, the club’s instructor, and the club’s administrator. Because of the dispute between the node 34 and node 1, the network is split into two small communities. The ground-truth community structure is shown in Figure 1(a).

Dolphins [33] is constructed on the basis of observations of 62 bottlenose dolphins over 7 years by Lusseau and Newman. The nodes and the edges of Dolphins network indicate bottlenoses and communications between two bottlenoses respectively. This network could be divided into four small communities. The ground-truth community structure is shown in Figure 2(a).

Risk Map [8] is a network of adventure game Risk 2, which is originated from a turn-based game for two to six players. It is composed of seven groups. The ground-truth community structure is shown in Figure 3(a).

Polbooks [7] is a network, derived from the politic books about US politics published around the time of the 2004 presidential election, which consists of 105 nodes and 441 edges. Nodes represent books sold by the online bookseller Amazon.com. Edges represent frequent copurchasing of books by the same buyers. It is composed of three communities. The ground-truth community structure is shown in Figure 4(a).

Football [8] is derived from the American football games of the schedule of Division I during regular season Fall 2000, where 115 vertices represent teams and 613 edges represent...
Table 1: Basic statistical information about the networks.

| Network    | Nodes | Edges | Communities | Average degree |
|------------|-------|-------|-------------|----------------|
| Karate     | 34    | 78    | 2           | 4.588          |
| Dolphins   | 62    | 159   | 4           | 5.129          |
| Risk map   | 42    | 83    | 7           | 3.952          |
| Polbooks   | 105   | 441   | 3           | 8.400          |
| Football   | 115   | 613   | 12          | 10.661         |
| Santa Fe   | 118   | 197   | 6           | 3.339          |
| Email      | 1,133 | 5,451 | —           | 9.628          |
| CA-GrQc    | 5,242 | 14,496| —           | 5.530          |
| CA-HepTh   | 9,877 | 25,998| —           | 5.264          |
| DBLP       | 317,080 | 1049,866 | —       | 6.622          |
| Amazon     | 334,863 | 925,872 | —       | 5.530          |

Figure 1: Continued.
Figure 1: The results of ACC and its baselines on Karate. (a) Ground truth. (b) ACC. (c) Spectral clustering. (d) Newman2006. (e) Infohiermap. (f) FastQ. (g) PPC. (h) Attractor. (i) LPA.

Figure 2: Continued.
regular-season games between the two teams they connect. The teams are divided into 12 conferences containing around 8–12 teams each, and thereby the real number of communities is also 12. The ground-truth community structure is shown in Figure 5(a).

Santa Fe [8] is a network which consists of 118 nodes and 197 edges. Each node represents a scientist who works at Santa Fe Institute, and each edge indicates the collaborations of scientists. There will be an edge if the two scientists have collaborated with each other at least once. It is composed of six communities. The ground-truth community structure is shown in Figure 6(a).

Email [34] is a network without ground-truth community structure, which consists of 5,451 emails. Nodes in the network are employees and edges are emails.

CA-GrQc [35] is an author collaboration network from the e-print arXiv and covers scientific collaborations between authors whose papers are submitted to General Relativity and Quantum Cosmology category. An edge between two authors represents a common publication. The network has 5,242 nodes and 14,496 edges, spanning from January 1993 to April 2003.

CA-HepTh [35] is an author collaboration network from the e-print arXiv and covers scientific collaborations between authors whose papers are submitted to High Energy Physics-Theory category. An edge between two authors represents a common publication. The network has 9,877 nodes and 25,998 edges, spanning from January 1993 to April 2003.

DBLP [36] is a coauthorship network where two authors are connected if they publish at least one paper together. The DBLP computer science bibliography provides a comprehensive list of research papers in computer science. Publication venue, e.g., journal or conference, defines an individual ground-truth community; authors who published to a certain journal or conference form a community.

Amazon [36] is a network which was collected by crawling Amazon website. It is based on customers who bought an item and also bought a feature of the Amazon
Figure 3: The results of ACC and its baselines on Risk Map. (a) Ground truth. (b) ACC. (c) Spectral clustering. (d) Newman2006. (e) Infohiermap. (f) FastQ. (g) PPC. (h) Attractor. (i) LPA.
Figure 4: The results of ACC and its baselines on Polbooks. (a) Ground truth. (b) ACC. (c) Spectral clustering. (d) Newman2006. (e) Info hier map. (f) FastQ. (g) PPC. (h) Attractor. (i) LPA.

Figure 5: Continued.
Figure 5: The results of ACC and its baselines on Football. (a) Ground truth. (b) ACC. (c) Spectral clustering. (d) Newman2006. (e) Infohiermap. (f) FastQ (g) PPC. (h) Attractor. (i) LPA.

Figure 6: Continued.
website. If a product $i$ is frequently copurchased with product $j$, the graph contains an undirected edge from $i$ to $j$.

5.1.3. Evaluation Matrices. On the networks which already have known community structures, the performances of ACC and the seven baselines are quantitatively measured by three widely used external evaluation measures: Adjusted Rand Index (ARI), Normalized Mutual Information (NMI) and Accuracy, and one popular internal measure Modularity ($Q$).

On the networks without ground-truth community structures, to compare the performances of distinct algorithms in an objective way is a non-trivial task. But since Modularity is the most common internal measure in community detection, we use it to evaluate the quality of communities produced by ACC and the seven baselines.

5.2. Networks with Ground-Truth Community Structures. To demonstrate that ACC can be applied to networks containing different communities, the results of all the algorithms on the networks with ground-truth community structures are exhibited in Figures 1–4, and 6, where colors of nodes indicate different detected communities. The quantitative performances of ACC and its baselines are summarized in Table 2, which are also shown at the bottom of the corresponding figures respectively to give visual explanations. For ground-truth communities of each network, since the ARI, NMI, and Accuracy are always 1, we only show $Q$ on the Bar chart. Note that for LPA, the ARI, NMI, and Accuracy are the average values of thirty times running, and the figures demonstrated are the best picked from figures of thirty times.

5.2.1. Risk Map. Figure 3 shows the results of ACC and its baselines on Risk Map network. From Figure 3(b), we can see that ACC correctly identifies the community structure. Figures 3(e) and 3(g) show that Infohiermap and PPC also get the right partitions. From Table 2, we can see that the performances of FastQ and LPA are slightly behind that of ACC, Infohiermap, and PPC. However, Spectral Clustering, Newman2006, and Attractor identify the community structure incorrectly, which results in a relatively low value of Accuracy, ARI, and NMI.

5.2.2. Dolphins. Figure 2 exhibits the results of ACC and the compared algorithms on Dolphins network. Figure 2(b) shows that ACC successfully finds out the four communities, except for the node sn89. The two compared algorithms PPC and Infohiermap also find the basic community structure. But as demonstrated in Figures 2(g) and 2(e), respectively, PPC wrongly assigns six nodes and Infohiermap wrongly assigns seven nodes. The community structures identified by the other five comparing algorithms are not so clear as the above three. From Table 2, we can also know that ACC achieves the best performance with high values of Accuracy, NMI, and ARI. Hence, on this network, ACC outperforms the seven baselines.

5.2.3. Karate. Figure 1(b) illustrates the clustering results of our algorithm ACC and the other seven algorithms on Zachary Karate network. Our algorithm successfully identifies the two communities and divides this network into two communities clearly. Attractor obtains the second best result with only one wrongly assigned node. Additionally, there is no clear boundary and internal structure between
Table 2: A quantitative comparison of results generated by ACC and its baselines on networks with ground-truth community structures.

| Networks | Measures | GT | ACC | SC | New06 | Info | FastQ | PPC | Attractor | LPA |
|----------|----------|----|-----|----|-------|------|-------|-----|------------|-----|
| Risk Map | Q        | 0.634 | 0.634 | 0.619 | 0.547 | 0.634 | 0.625 | 0.634 | 0.462 | 0.594 |
|          | Accuracy | 1.000 | 1.000 | 0.857 | 0.762 | 1.000 | 0.810 | 1.000 | 0.571 | 0.858 |
|          | NMI      | 1.000 | 1.000 | 0.884 | 0.756 | 1.000 | 0.855 | 1.000 | 0.754 | 0.883 |
|          | ARI      | 1.000 | 1.000 | 0.599 | 0.610 | 1.000 | 0.709 | 1.000 | 0.367 | 0.753 |
| Dolphins | Q        | 0.518 | 0.520 | 0.379 | 0.491 | 0.525 | 0.491 | 0.524 | 0.274 | 0.514 |
|          | Accuracy | 1.000 | 0.984 | 0.677 | 0.710 | 0.887 | 0.839 | 0.887 | 0.565 | 0.823 |
|          | NMI      | 1.000 | 0.949 | 0.592 | 0.635 | 0.898 | 0.733 | 0.902 | 0.529 | 0.792 |
|          | ARI      | 1.000 | 0.945 | 0.420 | 0.496 | 0.828 | 0.680 | 0.828 | 0.269 | 0.691 |
| Karate   | Q        | 0.371 | 0.371 | 0.313 | 0.393 | 0.402 | 0.381 | 0.402 | 0.371 | 0.365 |
|          | Accuracy | 1.000 | 1.000 | 0.912 | 0.618 | 0.824 | 0.735 | 0.823 | 0.971 | 0.820 |
|          | NMI      | 1.000 | 1.000 | 0.646 | 0.677 | 0.699 | 0.692 | 0.711 | 0.927 | 0.690 |
|          | ARI      | 1.000 | 1.000 | 0.669 | 0.512 | 0.702 | 0.680 | 0.702 | 0.939 | 0.661 |
| Polbooks | Q        | 0.415 | 0.495 | 0.448 | 0.467 | 0.526 | 0.502 | 0.419 | 0.501 | 0.500 |
|          | Accuracy | 1.000 | 0.810 | 0.495 | 0.714 | 0.781 | 0.800 | 0.580 | 0.648 | 0.648 |
|          | NMI      | 1.000 | 0.566 | 0.434 | 0.520 | 0.541 | 0.523 | 0.395 | 0.487 | 0.526 |
|          | ARI      | 1.000 | 0.681 | 0.307 | 0.547 | 0.646 | 0.623 | 0.340 | 0.485 | 0.601 |
| Santa Fe | Q        | 0.739 | 0.718 | 0.705 | 0.708 | 0.65/0.704 | 0.749 | 0.740 | 0.698 | 0.674 |
|          | Accuracy | 1.000 | 0.847 | 0.788 | 0.831 | 0.636/0.602 | 0.831 | 0.771 | 0.653 | 0.754 |
|          | NMI      | 1.000 | 0.886 | 0.851 | 0.834 | 0.764/0.805 | 0.867 | 0.880 | 0.808 | 0.753 |
|          | ARI      | 1.000 | 0.827 | 0.493 | 0.735 | 0.582/0.574 | 0.741 | 0.760 | 0.668 | 0.599 |
| Football | Q        | 0.601 | 0.560 | 0.559 | 0.492 | 0.600 | 0.550 | 0.553 | 0.600 | 0.586 |
|          | Accuracy | 1.000 | 0.852 | 0.835 | 0.652 | 0.991 | 0.548 | 0.878 | 0.991 | 0.924 |
|          | NMI      | 1.000 | 0.952 | 0.923 | 0.762 | 0.989 | 0.762 | 0.887 | 0.989 | 0.960 |
|          | ARI      | 1.000 | 0.853 | 0.783 | 0.528 | 0.982 | 0.512 | 0.802 | 0.982 | 0.886 |

GT, ground-truth; SC, spectral clustering; New06, Newman2006; Info: infohiermap.

5.2.4. Polbooks. Figure 4 shows the results of applying our algorithm ACC and its baselines to Polbooks network. From both the Bar charts and community structure in Figure 4(b), we can know that ACC detects the basically correct community structure and gets the best results with the highest Accuracy. Infohiermap achieves the best performances on NMI and ARI. FastQ and LPA also get relatively good partitions. The results of the other algorithms are not as good as that of the four methods.

5.2.5. Santa Fe. Figure 6 shows the results of ACC and its seven baselines on the network Santa Fe. We can see that the result of ACC is much closer to the ground-truth than the results of the baselines, although it does not show very good performance on this network. Table 2 also shows that ACC achieves the highest ARI, NMI, and Accuracy of all the methods.

5.2.6. Football. Figure 5 demonstrates the results of ACC and the compared algorithms on the network Football. From Figure 5(b), we can see that ACC basically identifies the community structure, except for further partitioning of two of the total twelve communities. For algorithms Attractor and Infohiermap, they obtain the correct community except for a wrongly assigned node. LPA and Spectral clustering also yield a comparable result, while Newman2006, PPC, and FastQ produce a relatively bad grouping on this network.

5.2.7. Comprehensive Assessment on Networks with Ground-Truth Community Structures. To comprehensively analyze the performances of ACC and the seven baselines, we use box plot as a descriptive statistics means. Thus, to fairly evaluate each algorithm, the box plots of the four evaluation indexes are shown in Figure 7 respectively. For ARI, NMI, and Accuracy, ACC shows the best performance by a landslide in quartile, median, minimum, and maximum, respectively. Therefore, ACC shows the best statistics performance on networks with ground-truth community structures.

5.3. Networks without Ground-Truth Community Structures. In this section, we evaluate the performances of ACC and its baselines on relatively large scale networks without ground-truth community structures. As there exist no convincing measures for unlabelled network, we use Modularity (Q) to evaluate these algorithms in an informative way. Networks with high Modularity have dense connections between the nodes within communities but sparse connections between nodes in different communities. The value of the Modularity lies in the range [−1/2, 1], Q is equal to 0 only if all nodes are put in a single community [37]. Generally speaking, the smaller the value of Modularity is, the more random the communities obtained by Spectral clustering. The partition found by Newman2006 contains four communities. The results detected by FastQ, Infohiermap, and PPC show that they divide this network into three communities.
distribution of the edges is in the network is and the less clear the community structure is. High value of Modularity indicates a strong community structure. But in practical application, the achievable modularity score depends on the sparseness and size of the network [38]. As shown in the bar graphs of Figures 1–6 and the box plots in Figure 7, the largest Modularity does not necessarily correspond to the best result of community detection, and the community structure corresponding to the largest Modularity often has a certain deviation to the real community structure [12, 13]. Usually, a Modularity value larger than 0.3–0.4 is a clear indication that the subgraphs of the corresponding partition are modules [12]. The maximum Modularity of many networks approximately ranges from 0.4 to 0.7 [12].

The quantitative results of each algorithm on each network are listed in Table 3. Note that Newman2006, Spectral clustering, and Infohiermap do not obtain community structures on DBLP and Amazon due to their high time cost. There are no quantitative results for Attractor on networks CA-GrQc and CA-HepTh, since CA-GrQc and CA-HepTh are two unconnected graphs, whereas Attractor and Spectral clustering can only work on connected graphs. On network email, ACC gets a relatively smaller Modularity than that of PPC and Infohiermap. On network CA-GrQc, ACC gets a Modularity slightly less than that of Newman2006, Infohiermap, FastQ, and PPC. On network CA-HepTh, ACC gets a Modularity slightly less than that of Newman2006, Infohiermap, FastQ, PPC, and LPA. Although ACC does not get the largest Modularity, the Modularity values of ACC are comparable to that of the other methods, and all the Modularity values obtained by ACC are around the reasonable range [0.4, 0.7]. Moreover, if we just focus on Modularity, ACC is not the best one. According to the definition of Modularity, if an algorithm only yields a few communities, it naturally leads to better values of Modularity. While ACC can obtain comparatively reasonable number of communities, its Modularity is not as good as we supposed.

In total, the experiments on all real-world networks demonstrate that ACC not only allows extracting meaningful communities in networks with ground-truth community structures with high external measures Accuracy, NMI, and ARI, but also scales up large-size networks without ground-
truth community structures and yields a good partitioning in terms of the internal measure Modularity. Therefore, ACC is an effective community detection method.

5.4. Threshold of Similarity. We can see in Algorithm 1, ACC needs a similarity threshold, $\zeta$, which can be set to an integer. In this section, we perform experiments on LFR benchmark networks with different community structures and different distributions of node degrees to thoroughly investigate the impact of the similarity threshold on the quality of community detection of ACC and further discuss how to choose an appropriate $\zeta$ for a network.

5.4.1. On LFR Benchmark Networks Generated under Different Mixing Parameters. To quantitatively evaluate the impact of similarity threshold $\zeta$ on networks with different community structures, we have performed experiments on four LFR benchmark networks generated under different mixing parameters by keeping the other parameters unanimous. Mixing parameters $mu$ are set to 0.3, 0.4, 0.5, and 0.6, respectively, for each network. The other parameters for each LFR networks are as follows: the number of nodes is 10,000, the average degree of nodes is 15, and the maximum degree of nodes is 30. Mixing parameter is the fraction of intra-community edges incident to each node. The bigger the mixing parameter of a network is, the more difficult it will be to reveal the community structure. We can see from Figure 8, ACC still can detect the communities correctly from the two LFR benchmark networks generated under mixing parameters 0.5 and 0.6, where the community structures are not very clear.

Besides, as shown in Figure 8, the range of similarity threshold $\zeta$ is relatively big under different mixing parameters; especially when mu is 0.3 and 0.4, the community structures are clear. When mu is 0.3 and 0.4, ACC can get the satisfying results under the threshold 0. As the mixing parameter mu increases, the number of common neighbors of two adjacent nodes on LFR networks decreases gradually when the distribution of node degrees in the networks does not change. Therefore, according to equation (1), the value of the similarity will decrease as the mixing parameter increases. Correspondingly, the range of threshold gradually shifts toward the left side of 0 as shown in Figure 8.

5.4.2. On LFR Benchmark Networks with Different Distributions of Node Degrees. Since our similarity in ACC is determined by the number of common neighbors and the smaller degree of two adjacent nodes, we have performed experiments on LFR benchmark networks with different distributions of node degrees. In this experiment, the size of the networks is still 10,000, the mixing parameter is fixed to 0.4, the average degree of nodes is fixed to 15, and the maximum degrees of nodes are 30, 45, and 60, respectively, on the three networks. We all know that when the average degree of nodes on a network remains unchanged, as the maximum degree of nodes increases, the degrees of the smaller degree nodes will decrease. If the average degree of nodes and the number of nodes are the same, the number of common neighbors of two adjacent nodes on LFR benchmark networks generated under the same mixing parameter will remain little changed. Therefore, the similarity will increase according to equation (1). As is shown in Figure 9, the effective range of $\zeta$ shifts toward the right side as the maximum degree of nodes increases.

On all the networks used in this paper, $\zeta$ is 0 except the network Santa Fe, where $\zeta$ is −1. Thus, in real applications, we suggest setting $\zeta$ to 0. When the appropriate initial communities cannot be obtained, the value of threshold $\zeta$ can be reduced to get satisfying partition.

5.5. Similarity Evaluation. To fairly validate the similarity sim that we propose, sim and three other well-known similarity measures are applied to ACC on the same networks. The three well-known similarity measures are Common Neighbor (CN) [39], Jaccard [31], and Salton and McGill [40], of which Jaccard similarity is a frequently used measure of similarity of two nodes, and it is also a measure of similarity of two links [41].

Given $G(V, E)$, for two adjacent nodes $u$ and $v$, the three similarity measures are as follows:

| Networks     | Measures | ACC | SC | New06 | Info | FastQ | PPC | Attractor | LPA |
|--------------|----------|-----|----|-------|------|-------|-----|-----------|-----|
| Email        | Q        | 0.466 | 0.325 | 0.489 | 0.535 | 0.507 | **0.559** | 0.464 | 0.393 |
|              | Communities | 78   | 48  | 7     | 60   | 11    | 16  | 145       | 45  |
| CA-GrQc      | Q        | 0.741 | —   | 0.782 | 0.842 | 0.812 | **0.846** | —     | 0.737 |
|              | Communities | 523  | —   | 394   | 373  | 426   | 449 | —         | 1053 |
| CA-HepTh     | Q        | 0.604 | —   | 0.613 | **0.742** | 0.714 | **0.742** | —     | 0.620 |
|              | Communities | 961  | —   | 463   | 459  | 547   | 10  | —         | 1775 |
| DBLP         | Q        | 0.672 | —   | —     | —    | 0.735 | **0.790** | 0.592 | 0.673 |
|              | Communities | 13212 | —   | —     | —    | 3090  | 4108 | 36043     | 20874 |
| Amazon       | Q        | 0.737 | —   | —     | —    | 0.869 | **0.901** | 44513 | 0.786 |
|              | Communities | 25933 | —   | —     | —    | 1481  | 3469 | 0.668     | 22278 |

SC: spectral clustering; New06, Newman2006; Info: infohiermap.
For CN, Jaccard and Salton, we determine their relative optimal results from plenty of experiments by tuning their corresponding input thresholds. For our similarity sim, the threshold is always set to 0. The relative optimal results of the four similarities for ACC algorithm and their corresponding thresholds are listed in Table 4. We can see from Table 4, by tuning the corresponding thresholds, each similarity can get an effective result, and results of the four similarities are of little differences. Relatively, sim gets the highest ARI, NMI, and Accuracy on networks Risk Map, Dolphins Karate, Polbooks, and Football. On network Santa Fe, sim achieves the highest ARI and NMI, and Salton gets the highest Accuracy. Therefore, taking all these factors into account, sim is superior to the other three similarity measures.

5.6 Robustness on Incomplete Networks. We are also interested in probing the efficiency of ACC on incomplete networks as in [42, 43]. In this section, we evaluate the robustness of ACC in discovering communities on incomplete networks generated from sampling processes.

To create an incomplete network $G'$, where $G' \subset G$, the edges in $G$ are randomly sampled with a sampling rate 10%
Figure 8: The impact of similarity threshold $\zeta$ on LFR networks generated under different mixing parameters.

Figure 9: The impact of similarity threshold $\zeta$ on LFR networks with different distributions of node degrees.

Table 4: A comparison of the results of applying different similarity measures to ACC algorithm.

| Network | Measures | sim | CN   | Jaccard | Salton |
|---------|----------|-----|------|---------|--------|
| Risk Map | Q        | 0.634 | 0.600 | 0.600   | 0.631  |
|         | Accuracy | 1.000 | 0.881 | 0.881   | 0.833  |
|         | NMI      | 1.000 | 0.857 | 0.857   | 0.871  |
|         | ARI      | 1.000 | 0.735 | 0.735   | 0.734  |
|         | Threshold| 0     | 1     | 0.1     | 0.4    |
| Dolphins | Q       | 0.520 | 0.520 | 0.512   | 0.510  |
|          | Accuracy | 0.984 | 0.984 | 0.887   | 0.903  |
|          | NMI      | 0.949 | 0.949 | 0.881   | 0.839  |
|          | ARI      | 0.945 | 0.945 | 0.808   | 0.796  |
|          | Threshold| 0     | 2     | 0.2     | 0.4    |
| Karate   | Q        | 0.371 | 0.371 | 0.371   | 0.371  |
|          | Accuracy | 1.000 | 1.000 | 1.000   | 1.000  |
|          | NMI      | 1.000 | 1.000 | 1.000   | 1.000  |
|          | ARI      | 1.000 | 1.000 | 1.000   | 1.000  |
|          | Threshold| 0     | 8     | 0.3     | 0.5    |
interval from 10% to 100% as suggested in [43]. Then, we run ACC on these networks to evaluate its robustness. To highlight the variation in the community detection quality, we average the results over 10 datasets for each sampling rate. Figure 10 shows the robustness results of ACC on incomplete networks. The lines represent the mean NMI on each sampling rate. ACC is robust even when several edges are missing.

5.7. The Advantages of ACC. From the above comprehensive studies and experiments, we can conclude that ACC has the following advantages:

(1) ACC can detect communities from different types and scales of networks automatically with high accuracy and stability

(2) ACC can discover the core for a community according to the definition of the local density proposed in this paper

(3) ACC is a community detection algorithm with near-linear in the time complexity

6. Conclusion

In this work, we develop a simple but effective community detection algorithm, named ACC. ACC combines the advantages of similarity-based methods with the advantages of core-based methods. ACC depends on a novel similarity to get initial communities from networks and further adjusts the initial communities according to the cores in them found out through our new definition of local density. Then, ACC assigns the remaining unlabelled nodes to their high local density neighbors and adjusts border nodes to their neighbors with most links with them. ACC can overcome the limitation of heuristic search according to certain criteria, i.e., the calculation of heuristic values is time-consuming. We demonstrate the power of ACC on different

| Network | Measures | sim | CN | Jaccard | Salton |
|---------|----------|-----|----|---------|--------|
| Polbooks | Q | 0.495 | 0.416 | 0.441 | 0.486 |
| | Accuracy | 0.810 | 0.790 | 0.809 | 0.714 |
| | NMI | 0.535 | 0.530 | 0.519 | 0.506 |
| | ARI | 0.641 | 0.564 | 0.543 | 0.568 |
| | Threshold | 0 | 6 | 0.1 | 0.6 |
| Football | Q | 0.560 | 0.579 | 0.555 | 0.587 |
| | Accuracy | 0.852 | 0.809 | 0.852 | 0.817 |
| | NMI | 0.952 | 0.916 | 0.944 | 0.914 |
| | ARI | 0.853 | 0.805 | 0.850 | 0.734 |
| | Threshold | 0 | 2 | 0.3 | 0.3 |
| Santa Fe | Q | 0.718 | 0.724 | 0.709 | 0.711 |
| | Accuracy | 0.847 | 0.856 | 0.864 | 0.873 |
| | NMI | 0.886 | 0.877 | 0.855 | 0.858 |
| | ARI | 0.827 | 0.816 | 0.791 | 0.802 |
| | Threshold | 0 | 1 | 0.8 | 0.9 |
networks by comparing it with some state-of-the-art community detection methods. Experimental results provide compelling evidence that ACC is an effective community detection algorithm.

**Data Availability**

The complex network data used to support the findings of this study can be found in the website http://snap.stanford.edu/data/.

**Conflicts of Interest**

The authors declare that they have no conflicts of interest.

**Acknowledgments**

The research was supported in part by the National Natural Science Foundation of China (grant nos. 61762057 and 61762077) and in part by the Foundation of A Hundred Youth Talents Training Program of Lanzhou Jiaotong University.

**References**

[1] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, Article ID 066133, 2004.

[2] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, Oakland, CA, USA, June 1967.

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

[4] 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, Sydney, Australia, August 2015.

[5] L. Bai, X. Cheng, J. Liang, and Y. Guo, "Fast graph clustering with a new description model for community detection," *Information Sciences*, vol. 388-389, pp. 37–47, 2017.

[6] M. Chen, X. Wen, Z. Yang, M. Li, and M. Zhang, "MulSim: a novel similar-to-multiple-point clustering algorithm," *IEEE Access*, vol. 6, no. 1, pp. 78225–78237, 2018.

[7] 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.

[8] 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.

[9] Y. Wang, H. Huang, C. Feng, and Z. Liu, "Community detection based on minimum-cut graph partitioning," in *Proceedings of the International Conference on Web-Age Information Management*, pp. 57–69, Springer, Nanchang, China, June 2015.

[10] J. Shi and J. Malik, "Normalized cuts and image segmentation, pattern analysis and machine intelligence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[11] U. Brandes, D. Delling, M. Gaertler et al., "Maximizing modularity is hard," http://arxiv.org/abs/physics/0608255.

[12] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 1, pp. 36–41, 2007.

[13] A. Kehagias and L. Pitsoulis, "Bad communities with high modularity," *The European Physical Journal B*, vol. 86, no. 7, pp. 1–11, 2013.

[14] F. D. Malliaros and M. Vazirgiannis, "Clustering and community detection in directed networks: a survey," *Physics Reports*, vol. 553, no. 4, pp. 95–142, 2013.

[15] K. Kloster and D. F. Gleich, "Heat kernel based community detection, knowledge discovery and data mining," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1386–1395, New York, NY, USA, August 2014.

[16] M. Chen, L. Li, B. Wang, J. Cheng, L. Pan, and X. Chen, "Effectively clustering by finding density backbone based-on knn," *Pattern Recognition*, vol. 60, pp. 486–498, 2016.

[17] A. E. Sariy¨ uce, B. Gedik, G. Jacques-Silva, K.-L. Wu, and Ú. V. Çataly¨ urek, "Streaming algorithms for k-core decompositions," *Proceedings of the VLDB Endowment*, vol. 6, no. 6, pp. 433–444, 2013.

[18] L. Wang, T. Lou, J. Tang, and J. E. Hopcroft, "Detecting community kernels in large social networks," in *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, pp. 784–793, Vancouver, Canada, December 2011.

[19] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[20] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, Article ID 036104, 2006.

[21] 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.

[22] M. Rosvall and C. T. Bergstrom, "Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems," *PLoS One*, vol. 6, no. 4, Article ID e18209, 2011.

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

[24] X.-K. Zhang, J. Ren, C. Song, J. Jia, and Q. Zhang, "Label propagation algorithm for community detection based on node importance and label influence," *Physics Letters A*, vol. 381, no. 33, pp. 2691–2698, 2017.

[25] P. Liakos, A. Ntoulas, and A. Delis, "COEUS: community detection via seed-set expansion on graph streams," in *Proceedings of the 2017 IEEE International Conference on Big Data (Big Data)*, pp. 676–685, IEEE, Boston, MA, USA, December 2017.

[26] P. Liakos, A. Ntoulas, and A. Delis, "Scalable link community detection: a local dispersion-aware approach," in *Proceedings of the 2016 IEEE International Conference on Big Data (Big Data)*, pp. 716–725, IEEE, Washington, DC, USA, December 2016.

[27] 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.
[28] C. Liu, J. Liu, and Z. Jiang, “A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 44, no. 12, pp. 2274–2287, 2014.

[29] W. Wu, S. Kwong, Y. Zhou, Y. Jia, and W. Gao, “Nonnegative matrix factorization with mixed hypergraph regularization for community detection,” Information Sciences, vol. 435, pp. 263–281, 2018.

[30] M. Chen, M. Zhang, M. Li, M. Leng, Z. Yang, and X. Wen, “Detecting communities by suspecting the maximum degree nodes,” International Journal of Modern Physics B, vol. 33, no. 13, Article ID 1950133, 2019.

[31] P. Jaccard, “The distribution of the flora in the alpine zone,” New Phytologist, vol. 11, no. 2, pp. 37–50, 1912.

[32] W. W. Zachary, “An information flow model for conflict and fission in small groups I,” Journal of Anthropological Research, vol. 33, no. 4, p. 473, 1977.

[33] D. Lusseau and M. E. J. Newman, “Identifying the role that animals play in their social networks,” Proceedings of the Royal Society of London. Series B: Biological Sciences, vol. 271, no. 6, pp. S477–S481, 2004.

[34] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, “Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters,” Internet Mathematics, vol. 6, no. 1, pp. 29–123, 2009.

[35] J. Leskovec, J. Kleinberg, and C. Faloutsos, “Graph evolution: densification and shrinking diameters,” ACM Transactions on Knowledge Discovery from Data (ACM TKDD), vol. 1, no. 1, pp. 1–40, 2007.

[36] C. Yang, C. Li, Q. Wang, D. Chung, and H. Zhao, “Implications of pleiotropy: challenges and opportunities for mining big data in biomedicine,” Frontiers in Genetics, vol. 6, pp. 1–6, 2015.

[37] U. Brandes, D. Delling, M. Gaertler et al., “On modularity clustering,” IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 2, pp. 172–188, 2008.

[38] B. H. Good, Y. De Montjoye, and A. Clauset, “Performance of modularity maximization in practical contexts,” Physical Review E, vol. 81, no. 4, Article ID 046106, 2010.

[39] F. Lorrain and H. C. White, “Structural equivalence of individuals in social networks,” The Journal of Mathematical Sociology, vol. 1, no. 1, pp. 49–80, 1971.

[40] G. Salton and M. J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, New York, NY, USA, 1983.

[41] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, “Link communities reveal multiscale complexity in networks,” Nature, vol. 466, no. 7307, pp. 761–764, 2010.

[42] D. Shizuka and D. R. Farine, “Measuring the robustness of network community structure using assortativity,” Animal Behaviour, vol. 112, pp. 237–246, 2016.

[43] D. R. Amancio, O. N. Oliveira, and L. D. F. Costa, “Robustness of community structure to node removal,” Journal of Statistical Mechanics: Theory and Experiment, vol. 2015, no. 3, Article ID P03003, 2015.
