entropy

Article

Community Detection Method Based on Node Density, Degree Centrality, and K-Means Clustering in Complex Network

Biao Cai [1,2,*], Lina Zeng [1,*], Yanpeng Wang [1], Hongjun Li [1] and Yanmei Hu [1]

[1] College of Information Science &Technology, Chengdu University of Technology, Chengdu 610059, China
[2] Key Laboratory of Manufacturing Process Testing Technology of Ministry of Education of China, Southwest of University of Science and Technology, Mianyang 621010, China
* Correspondence: caibiao@cdut.edu.cn (B.C.); linazeng23@126.com (L.Z.)

Received: 4 November 2019; Accepted: 21 November 2019; Published: 23 November 2019

Abstract: Community detection in networks plays a key role in understanding their structures, and the application of clustering algorithms in community detection tasks in complex networks has attracted intensive attention in recent years. In this paper, based on the definition of uncertainty of node community belongings, the node density is proposed first. After that, the DD (the combination of node density and node degree centrality) is proposed for initial node selection in community detection. Finally, based on the DD and *k*-means clustering algorithm, we proposed a community detection approach, the density-degree centrality-jaccard-*k*-means method (DDJKM). The DDJKM algorithm can avoid the problem of random selection of initial cluster centers in conventional *k*-means clustering algorithms, so that isolated nodes will not be selected as initial cluster centers. Additionally, DDJKM can reduce the iteration times in the clustering process and the over-short distances between the initial cluster centers can be avoided by calculating the node similarity. The proposed method is compared with state-of-the-art algorithms on synthetic networks and real-world networks. The experimental results show the effectiveness of the proposed method in accurately describing the community. The results also show that the DDJKM is practical a approach for the detection of communities with large network datasets.

Keywords: community detection; CB-uncertainty (Community belongings uncertainty); DD (the combination of node density and node degree centrality); *k*-means

1. Introduction

Recently, complex networks have attracted a great deal of attention in various fields [1,2], including sociology, computer science, mathematics, and biology. For large-scale networks, the presence of communities is an important feature, as it indicates the existence of groups of vertices within which connections are dense, but between which they are sparse [3]. Indeed, community detection has been widely applied in, e.g., community establishment in social media [4], the collection of similar features in parallel processing [5,6], and sharing research interests by intergroup authors in co-authorship networks [7].

To date, a large number of community detection algorithms for complex networks have been proposed [8,9], including hierarchical clustering algorithms [10], label propagation algorithms [11–13], density-based algorithms [14,15], random-walk-based algorithms [16,17], and so on. The *k*-means clustering algorithm divides the data into clusters (the cluster number is predetermined) based on minimum error functions [18]. This algorithm is characterized by rapid clustering, easy implementation, and effective classification in large-scale dataset, and has been widely applied for community detection in complex networks. Additionally, the *k*-means clustering algorithm shows low time complexity compared to clustering methods based on centrality and similarity [19–21]. Nevertheless, conventional k-means clustering algorithms have several limitations [22]. First, the selection of initial cluster centers in traditional k-means clustering algorithms, which has a determining effect on the clustering result, is a random process. Hence, effective clustering cannot be guaranteed [23]. Second, the node similarity has a significant effect on the convergence rate and accuracy of k-means clustering algorithms. Therefore, the iteration times in the k-means clustering algorithm can be effectively reduced, and the accuracy of community classification can be effectively improved by selecting appropriate initial cluster centers, defining appropriate node similarities, and setting appropriate stop conditions.

In this paper, the k-means clustering-based DDJKM algorithm for community detection was proposed, in which the community belongingness of nodes was described by the node uncertainty; density was introduced by information entropy, and the initial cluster centers were selected by the balance of the degree centrality, density, and the similarity of nodes. In this algorithm, the node similarity matrix is constructed as the clustering matrix by the node similarity in the network. This algorithm can effectively select the clustering center, thus preventing the selection of initial cluster centers that are too close to each other, and reducing the iteration times in the clustering process. The experimental results show the feasibility of the algorithm.

The rest of the paper is organized as follows: The theory behind the proposed algorithm, including the calculation equations for node uncertainty, node degree, node density, node balance, and node similarity, is discussed in Section 2. The details of DDJKM algorithm are given in Section 3. The performance of the proposed algorithm is evaluated in real-world networks and artificial networks, and compared with those of existing algorithms in Section 4. Finally, the conclusion is presented in Section 5.

2. Theory

2.1. Uncertainty

In the study of community structures in complex networks, the community belongingness (CB) of a node is certain if this node and its adjacent nodes are in the same community. Otherwise, the CB of a given node exhibits uncertainty. This is consistent with evaluations of information uncertainty by information entropy, where information uncertainty is proportional to information entropy. Therefore, the uncertainty of CB of nodes was established as follows:

The network is represented by an unweighted, undirected graph $G = (V, E)$, where $V(G) = \{v_1, v_2, \ldots, v_n\}$ refers to the node set, and $E(G) = \{e_1, e_2, \ldots, e_k\}$ refers to the edge set. $|V| = n, |E| = m$. $N(v_i)$ refers to the neighbor node set in the subgraph generated by the h hops forward breadth-first search (BFS) of v_i. If all $N(v_i)$ are in community c_j, the CB uncertainty of v_i in c_j is 0; if the majority of $N(v_i)$ are in community c_j, the CB uncertainty of v_i in c_j is considered to be low; if the majority of $N(v_i)$ are not in community c_j, the CB uncertainty of v_i in c_j is considered to be high. The parameter m refers to the number of communities in the network, and the CB uncertainty of the node refers to a quantified parameter if the node does not belong to a specific community. The CB uncertainty of a node in a specific community is defined as a random variable $C\left(c_1, c_2, c_3, \ldots, c_m\right)$, and the probability of the i-th node in the q-th community is defined as $p(c_q)$, where $q = 1,2,\ldots,m$. Then, the CB uncertainty of v_i is defined as:

$$Entorpy\left(v_i^h\right) = -\sum_{q=1}^{m} p(c_q)\, log_2\, p(c_q) \quad (1)$$

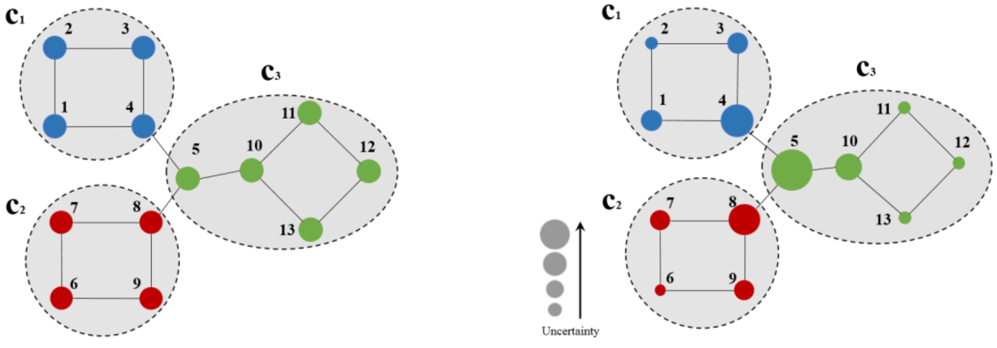
where i refers to the node number, h refers to the forward hops of BFS, and G_i^h refers to the subgraph generated by h-hop BFS of v_i as the initial node respectively. $p(c_q)$ refers to the ratio of the number of nodes in the subgraph $G_i^h(N)$ to the number of nodes in the community $c_q(N')$:

$$p\left(c_q\right) = \frac{N'}{N} \tag{2}$$

Figure 1 describes the CB uncertainty of example nodes. As shown in Figure 1a, three communities $\left(c_1, c_2, c_3\right)$ were presented, node 2 was identified in c_1, as well as nodes 1–4 in its subgraph of node 2 generated by two-hop forward BFS. According to Equation (2), the quantity ratios of nodes in the subgraph generated by a two-hop forward BFS of node 2 that are in c_1, c_2, and c_3 and all nodes in the subgraph were $p(c_1) = 1, p(c_2) = 0,$ and $p(c_3) = 0$, respectively. The uncertainty of node 2 at h = 2 was calculated by Equation (1):

$$Entorpy\left(v_2^2\right) = -(1 \times \log_2 1 + 0 + 0) = 0$$

Figure 1b shows the uncertainty of nodes on the sample network (node uncertainty decreased with its size). According to Figure 1b, nodes with high uncertainty are marginal ones connected to the community (e.g., nodes 4, 5, 8, and 10 in Figure 1b). Herein, node 5 exhibits maximum uncertainty, as it is connected to all three communities. On the other hand, nodes with low uncertainty are marginal ones that are not adjacent to any other community (e.g., nodes 2, 6, and 11–13 in Figure 1b), as the community belongingness of these nodes is highly likely.



(**a**) (**b**)

Figure 1. (**a**) Sample network; (**b**) Node uncertainty on the sample network at h = 2.

2.2. Community Belongingness

To determine the CB uncertainty of a given node, it is essential to obtain the CB of the node in advance. However, the initial CB of nodes for community detections in complex networks is unknown, and the CB uncertainty of nodes cannot be used as criteria for the selection of initial nodes in community detection algorithms; instead, quantified evaluation of the CB certainty of the corresponding node is required. As density is a measurable parameter in nature, we propose that the selection of initial nodes for community detection shall be based on the node density, instead of the entropy in the network. The node density is determined based on quantities of edges and nodes in the subgraph generated by a h-hop forward BFS of this node; it quantifies the CB certainty of this node in a specific community. The node density is defined as:

$$Density\left(v_i^h\right) = |E'|/(|V'|(|V'| - 1)/2) \tag{3}$$

where i refers to the i-th node, h refers to the forward hop count from v_i, V' refers to the set of nodes in the subgraph G' with h hops forward BFS from v_i, $|V'|$ refers to the quantity of nodes in V', E' refers to the set of edges in the subgraph G', and $|E'|$ refers to the quantity of edges in E'.

Figure 2 shows a sample network for the calculation of node density, and Table 1 summarizes the node density of the two-hop subgraph of each node.

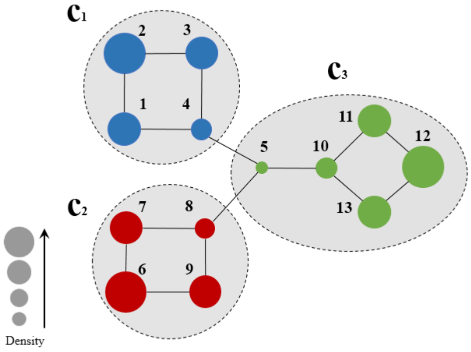


Figure 2. Community Belongingness of each node on the sample network of Figure 1a when *h* = 2.

Figure 2 illustrates a sample network for the calculation of node density. Herein, a two-hop forward was involved due to the small size of the sample network. For example, from the calculation of the density of node 1, the set of nodes two hops forward from node 1 is:

$$n(v_i^h) = n(v_1^2) = \{1, 2, 3, 4, 5\}.$$

Five nodes and five edges were observed in the subgraphs. The density of node 1 can be calculated by Equation (3):

$$Density(v_1^2) = \frac{5}{(5\times4)\ /2} = 0.2.$$

Table 1 summarizes the density of each node in two-hop subgraph of the sample network in Figure 2. As observed, the value of density is proportional to the CB certainty of the node, which is directly related to its location in the network. For instance, nodes 2, 6, node 12, which are marginal nodes in the network, exhibited high node density, while node 5, in the central part of the network, exhibited lowest node density. The real community structure has a similar characteristic: nodes with low node densities tend to occur with close connections to other communities, while nodes with high node densities exhibit no connections to other communities. This is the opposite to the node centrality in conventional community detections, and can be used for the determination of seed nodes for community division.

Table 1. CB uncertainty of each node in the two-hop subgraph on the sample network shown in Figure 2.

Node	Density
2, 6, 12	0.667
1, 3, 7, 9, 11, 13	0.5
4, 8, 10	0.333
5	0.2

2.3. Similarity

In complex networks, the connections among intracommunity nodes are dense, while intercommunity nodes are sparse [24]. Node similarity is an effective parameter for the quantification of node affinity; the degree of similarity between two nodes is proportional to their common adjacent nodes, i.e., nodes with high similarity tend to connect to each other. So, the similarity of two nodes is a key parameter in the evaluation of the affinity of nodes i and j [25]. Node similarity includes common neighbors, Cosine, Jaccard, Sorensen index, PHI, Preferential attachment, Adamic-Adar, Allocation of resources [26–33], and Random walk similarities [34–36]. In this paper, we interpret similarity of v_i and v_j by calculating it based on their Jaccard correlation coefficients:

$$JacSim(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)|}{|N(v_i) \cup N(v_j)|} \tag{4}$$

where $N(v_i)$ and $N(v_j)$ are adjacent node sets of node v_i and v_j , $|N(v_i) \cap N(v_j)|$ refers to the quantity of common adjacent nodes shared by v_i and v_j , and $|N(v_i) \cup N(v_j)|$ refers to the quantity of nodes in the union of common adjacent node sets of v_i and v_j .

2.4. Balance

It is well known that the selection of seed nodes with good centrality can improve the performance of *k*-means clustering. Centrality parameters including betweenness, closeness, k-shell, and uniform H-index have limitations in community detections [37]. The community centrality can precisely describe node centrality [38], and the computing complexity of community centrality is $O\ (nk^5)$. Despite this, the node degree centrality is a key parameter describing the community centrality in networks. Only the selection of seed nodes in *k*-means clustering algorithms based on node degree centrality may lead to overly-short distances between initial cluster centers, thus affecting clustering performance. As it can precisely reflect the CB certainty of nodes, the node density can be combined with the degree centrality as criteria for the selection of initial nodes. Therefore, $DD(v_i)$, the parameter for selection of the i -th initial node, is defined as:

$$DD(v_i) = Density(v_i^h) \times Degree(v_i) \tag{5}$$

where h refers to the hop count of forward BFS, $Density(v_i^h)$ refers to the node density of v_i calculated by Equation (3), and $Degree(v_i)$ refers to the node degree of v_i .

3. Method

In *k*-means clustering algorithms, the number of clusters is a key parameter. In [39], the Monte Carlo-based algorithm proposes an effective method by which to determine the community quantity. Hence, this study focuses on the effective selection of initial seed nodes and community detection in networks using *k*-means clustering algorithms in complex network with known community numbers.

As mentioned, node density is proportional to the CB certainty of a node in a specific community, and can be employed for the selection of seed nodes. However, the seed nodes cannot be selected based on the node density alone, as it may lead to the selection of isolated nodes, thus reducing the accuracy of clustering. Meanwhile, the seed nodes cannot be selected based on the degree centrality alone either, as most of the seed nodes selected in this way may be in same community due to the limited information contained in the degree centrality. Therefore, we propose DD , a parameter balancing node degree centrality and node density, as a criterion for initial node selection.

In summary, the DDJKM algorithm based on node density, degree centrality, and conventional *k*-means clustering algorithms is proposed. In this algorithm, initial cluster centers are selected based on a combination of node degree, density, and similarity, while node centrality is also considered to avoid the selection of isolated nodes, thus avoiding local convergence in clustering and improving the effectiveness of community detection.

3.1. DDJKM Algorithm

Input: undirected connection network $G = \{V, E\}$, the quantity of communities to be divided is $K. V$, and E are sets of nodes and edges.

Output: community division = Com (1), Com (2), …, Com (K).

Step 1: Establish the *n*-dimensional vector $E\ (G)$ of the node degree and the *n*-dimensional vector $D(G)$ of node density based on $Density\ (v_i^h)$:

$$D(G) = \left(Density(v_1^h), Density(v_2^h), \cdots, Density(v_n^h)\right) \tag{6}$$

Step 2: All nodes in the network are arranged in descending order, $DD\left(v_i\right)$, which is the product of node density and node degree according to Equation (5). In cases of nodes with same $DD(v_i)$, these nodes are arranged in ascending order of node number. In this way, $DDSeq(G)$, a sequence of $DD(v_i)$ of nodes in the entire network, is established;

Step 3: Select the first element in $DDSeq(G)$ as the first initial node in the *k*-means clustering algorithm, add it to the clustering center node set $Seed(v)$, and obtain $S(v)$, which consists of nodes in the network that are not clustering center nodes:

$$S(v) = G(v) - Seed(v) \tag{7}$$

where $G(v)$ is the set of all nodes in network G.

Step 4: Calculate the node similarity using Equation (4) and establish the *n*-dimensional $Jaccard\ (G)$ of nodes in network G:

$$Jaccard(G) = \begin{bmatrix} JacSim(v_1, v_1) & \cdots & JacSim\left(v_1, v_n\right) \\ \vdots & \cdots & \vdots \\ JacSim\left(v_n, v_1\right) & \cdots & JacSim\left(v_n, v_n\right) \end{bmatrix} \tag{8}$$

where $JacSim(v_i, v_j)$ refers to the *Jaccard* correlation coefficient between v_i and v_j.

Step 5: Calculate the correlation matrix $DDJ(G)$ of nodes in network G using Equations (6) and (8):

$$DDJ(G) = (D(G)D(G)^T)\, Jaccard(G) \tag{9}$$

where $D(G)D(G)^T$ is matrix product of $D(G)$ and $D(G)^T$, and $DDJ(G)$ is the Hadamard product of $D(G)D(G)^T$ and $Jaccard(G)$.

Step 6: Calculate the average correlation (R_p) of nodes in $S(v)$ and nodes in $Seed(v)$:

$$R_p = \sum_{q=1}^{|Seed(v)|} R_{qp} / |Seed(v)| \tag{10}$$

where R_{qp} refers to the node correlation (correlation value in the correlation matrix $DDJ(G)$) of v_p and s_q , $q = 1, 2, \ldots, |Seed(v)|, p = 1, 2, \ldots, |S(v)|, |Seed(v)|$ refers to the number of nodes in $Seed(v)$, and $|S(v)|$ refers to the quantity of nodes in $S(v)$.

Step 7: Determine the minimum average correlation ($\mathrm{Min}R_p$) and establish $MinMean(v)$ that consists of nodes in $S(v)$ with average correlation = $\mathrm{Min}R_p$.

Step 8: Calculate $DD\left(v_i\right)$, which is the product of node density and node degree of each node in the node set $MinMean(v)$, and add the node with the maximum $DD(v_i)$ to $Seed(v)$.

Step 9: If $|Seed(v)|$ = *K*, terminate iteration; if not, return to Step 6.

Step 10: Execute the *k*-means community detection clustering algorithm.

Step 11: Export *K* communities (Com (1), Com (2), …, Com (K)) as each community corresponds to a clustering result.

3.2. K-Means Community Detection Clustering Algorithm

Input: K clustering centers, node similarity matrix $Jaccard(G)$.

Output: Cluster (1), Cluster (2), …, Cluster (K).

Step 1: The Euclidean distance of node similarity vector is:

$$Jacd(jv_a, jv_b) = \sqrt{\sum_{i=1}^{n}\left(JacSim(v_a, v_i) - JacSim(v_b, v_i)\right)^2} \tag{11}$$

where jv_a and jv_b refer to similarity vectors (in $Jaccard\ (G)$) corresponding to $\mathrm{v_a}$ and $\mathrm{v_b}$. The Euclidean distance of other nodes to K clustering centers are inversely proportional to their similarity. Then, all nodes are categorized into the cluster whose clustering center has a shortest distance from this node. In this way, K clusters (Cluster (1), Cluster (2), …, Cluster (K)) are generated.

Step 2: Recalculate the clustering center of Cluster (*j*) and define it as a new clustering center C_j:

$$C_j = \sum_{n=1}^{|Cluster(j)|} Jaccard(G)_{nj} / |Cluster(j)| \tag{12}$$

where $Jaccard\ (G)\ _{nj}$ refers to the vector in $Jaccard\ (G)$ corresponding to v_n in the *j*-th cluster, $n = 1, 2, \ldots, |Clustr\ (K)|$, and $|Cluster\ (j)|$ refers to the number of nodes in the *j*-th cluster.

Step 3: Calculate the Euclidean distances of all new and previous clustering centers to determine their maximum variation (MaxDist).

Step 4: If MaxDist remains unchanged or the maximum iteration times (Max-Iteration) were reached, iteration is terminated; proceed to the next step, otherwise return to Step 1.

3.3. Complexity Analysis

The complexity of community detection in this study is mainly caused by the density and community detections. In the calculation of density, the density of each node should be calculated. Meanwhile, we define the forward hop count as h, the average node density as $\bar{d}$, the total number of nodes in the network as n, and the time complexity in the process as $O(n\bar{d}^h)$. As the density calculation is a local process, it can be achieved by distributed computation; the time complexity is $O(\bar{d}^h)$ where $h \leq 3$ in most cases. The DDJKM algorithm involves the calculation of a correlation degree matrix $DDJ(G)$, which is a sparse matrix. Meanwhile, $D\ (G)\ \times D\ (G)^T$ is a sparse matrix whose calculated complexity does not exceed $O(m)$. In community detection, the degree and local similarity of each node should be obtained, taking $O(mn)$ operations to traverse all edges and adjacent nodes, where m is the number of edges. The complexity of the *k*-means algorithm is $O(nKt)$, where K refers to the cluster quantity and t to the iteration times. As $K \ll n$ and $t \ll n$ in most cases, the complexity of DDJKM algorithm is $O(\bar{d}^h + mn + nKt + m) = O(mn)$.

4. Experimental

In this section, we used seven real network datasets and the LFR benchmark datasets to validate the performance of the proposed algorithm . The real-world networks include Zachary's karate club network [40], the Dolphin social network [41], Books about US politics network [42,43], the American college football network [44], the Amazon copurchase network [45], and the YouTube network [45]. LFR benchmark networks possess properties found in real-world networks, such as heterogeneous distributions of degree and community size. First, we present some commonly-used evaluation measures. Then, we explain the real network and computer-generated networks we use, and compare our algorithm with some known algorithms.

4.1. Evaluation Measures

Normalized mutual information (NMI) is taken as the performance measure. NMI reflects the similarity between the true community and the detected community structures. Given two parts, A and B, of a network, C is the confusion matrix. In C, C_{ij} is the number of nodes of community i of part A that are also in community j of part B [46]. NMI $I(A,B)$ is defined as follows [47]:

$$I(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}\, log\left(\frac{C_{ij}\cdot N}{C_{i}\cdot C_{\cdot j}}\right)}{\sum_{i=1}^{C_A} C_{i}\cdot log\left(\frac{C_i}{N}\right) + \sum_{j=1}^{C_B} C_{\cdot j}\cdot log\left(\frac{C_{\cdot j}}{N}\right)} \tag{13}$$

where, $C_A(C_B)$ is the number of classes in part $A(B)$, $C_i \cdot (C_{\cdot j})$ is the number of elements of C in row i (column j), and N is the total number of nodes. If $A = B$, $I(A,B) = 1$; if A and B are totally different, $I(A,B) = 0$. As NMI increases, the detected communities become more approximate to the true communities.

Given a network $G = (V,E)$, let $\mathbf{T}$ be the set of ground-truth communities and $\mathbf{D}$ be the set of communities detected by the community detection algorithm. Each ground-truth community $T_i \in \mathbf{T}$ (or each detected community $D_i \in \mathbf{D}$) is a set consisting of the member nodes. Average $F1$ score is a popular metric to evaluate the degree of similarity between two sets. When applied in community detection, it can be formed as [48].

$$F1(\mathbf{T},\mathbf{D})=\frac{1}{2}\left(\frac{1}{|\mathbf{T}|}\sum\nolimits_{T_i\in\mathbf{T}}F(T_i,\mathbf{D})+\frac{1}{|\mathbf{D}|}\sum\nolimits_{D_j\in\mathbf{D}}F\big(D_j,\mathbf{T}\big)\right) \tag{14}$$

where

$$F(T_i,\mathbf{D})=\max_{D_j\in\mathbf{D}}F1\big(T_i,D_j\big) \tag{15}$$

and $F1\big(T_i,D_j\big)$ is the harmonic mean of precision and recall. The formulation of $F\big(D_j,\mathbf{T}\big)$ can be expressed in the same way.

4.2. Testing Networks

4.2.1. Real-World Networks

In the following part, we provide a simple description of the real network used in the experiments. For all these networks, the community structure is recognized which makes them suitable to evaluate the community detection methods. Zachary's karate club [40] is one of most the widely-used networks in community detection. The 34 members of the club constitute the 34 nodes of the network. The relationships between members constitute the 78 edges of the network. The Dolphin social network [41], proposed by Lusseau, is shown in Figure 3. The connection of any two dolphins represents a tighter connection between them. The dolphin social network consists of 62 dolphins as the nodes and 159 connections as the edges. The network can be detected as two communities, as shown in Figure 4. The Books about US politics [42,43] network consists of 105 books about US politics published in 2004 and sold by amazon.com. Based on the descriptions and reviews of the books posted on Amazon, Newman divided the network into three communities. The network is shown in Figure 5. The American college football [44] network was proposed by Girvan and Newman. The nodes represent different football teams, and the edges represent the matches between them. The network consists of 115 nodes and 616 edges. The network consists of 12 communities comprising 12 football teams. The network is shown in Figure 6. The Amazon copurchase and YouTube networks are provided by SNAP [45].

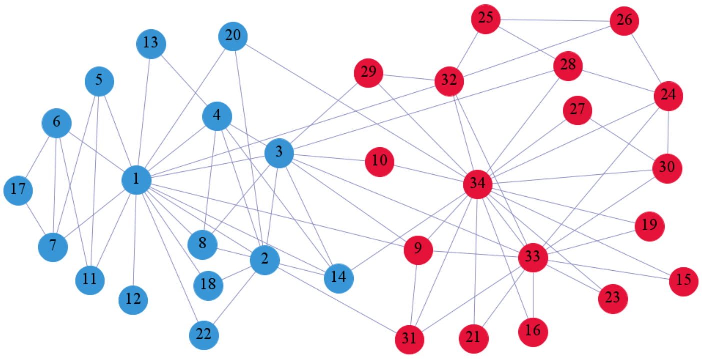


Figure 3. Zachary's karate club.

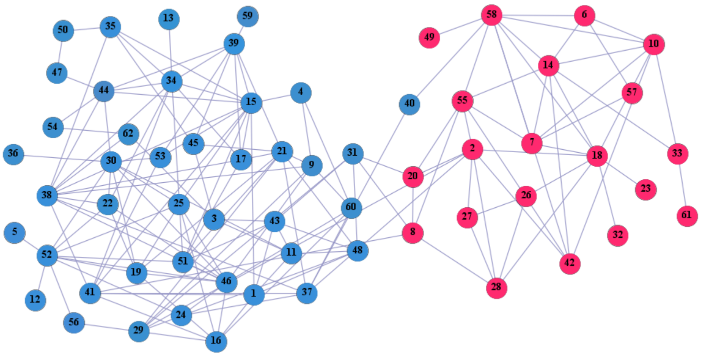


Figure 4. Dolphin social network.

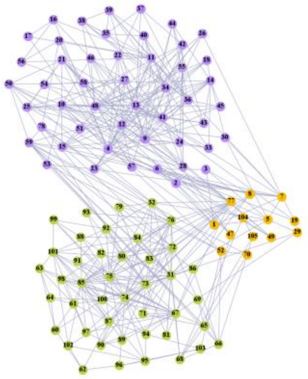


Figure 5. Books about US politics.

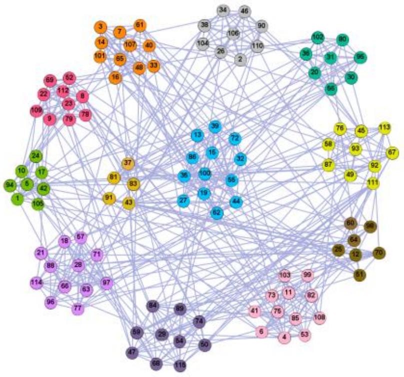


Figure 6. American College football.

4.2.2. Computer-Generated Network

We tested our algorithm on LFR benchmark networks which were proposed by Lancichinetti et al. [49]. The LFR generation program provides a rich set of parameters through which the network topology can be controlled, including network size N, the average degree $\langle k \rangle$, the maximum degree k_{max}, the minimum and maximum community size, C_{min} and C_{max} respectively, and the mixing parameters μ. The node degrees are governed by power laws with exponents of τ_1 and τ_2. In this work, we employ four types of LFR networks with scales of 1000 (LFR1), 2000 (LFR2), and 5000 (LFR3, LFR4) nodes with other corresponding parameters, as shown in Table 2.

Table 2. Parameter settings of LFR benchmark networks.

Network	N	τ_1	τ_2	C_{min}	C_{max}	$\langle k \rangle$	k_{max}	μ
LFR1	1000	2	1	20	50	20	50	0.1-0.9
LFR2	2000	2	1	20	100	20	50	0.1-0.9
LFR3	5000	2	1	20	50	20	50	0.1-0.9
LFR4	5000	2	1	20	100	15	75	0.1-1.0

4.3. Experimental Results and Analysis

In this study, the performance of the proposed algorithm was evaluated using five real-world networks and LFR networks. According to the small world effect, which indicates that the average minimum route between any two nodes in a complex network is 6, h in the forward BFS shall be set as 3 to achieve optimized performance. The criteria for iteration termination in the proposed algorithm are consistent with those in conventional k-means algorithms, i.e., once the Euclidean distances of new and previous clustering center vectors remain unchanged, iteration is terminated, indicating convergence at constant clustering, which is defined as one of the iteration termination conditions. Meanwhile, the Max-Iteration variable was set to 100 since the maximum number of observed in this paper iterations was 20. Therefore, the network parameters in this study were determined based on $h = 3$ and Max-Iteration = 100.

4.3.1. Experiments on Real-World Networks

We used the five real-world networks mentioned above to verify the efficiency of our algorithm. As shown in Figure 3; Figure 7, the final community structure of the Zachary's karate club network detected by DDJKM was consistent with the actual structure. It can be seen from Figures 4 and 8 that the structure in the Dolphin social network detected by our algorithm is also very close to the actual structure. Only node 40 is misidentified by our algorithm, and it can be seen that node 40 is in close proximity to two communities. The results for the Books about US politics network detected by our algorithm are shown in Figure 9. In the American college football network, our algorithm divides it by 12 (Figure 10) and 11 (Figure 11). Compared with the results shown in Figure 6, we can see that our algorithm performs well on the American football network; most nodes are correctly classified into their actual community structures.

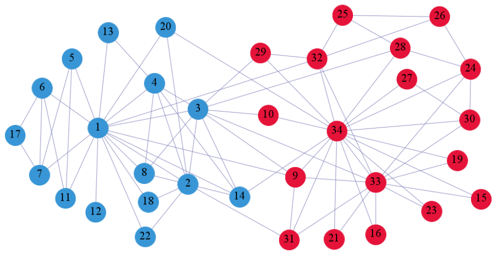


Figure 7. The community structure of the Zachary's karate club network as detected by the proposed method.

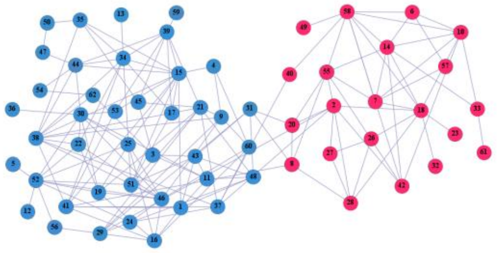


Figure 8. The community structure of the Dolphin social network as detected by the proposed method.

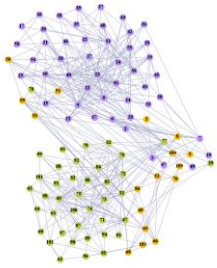


Figure 9. The community structure of the Books about US politics network as detected by the proposed method.

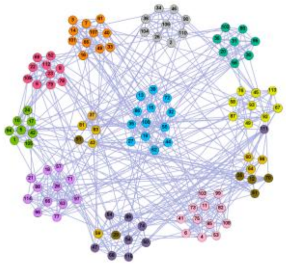


Figure 10. The community structure of the American college football network as detected by the proposed method (12 communities).

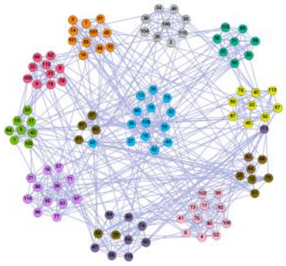


Figure 11. The community structure of the American college football network as detected by the proposed method (11 communities).

We compared the performance of our algorithm with the GN algorithm [24], the Newman fast greedy algorithm (FG) [50], the sparse linear coding method (SLC) [51], the MIGA algorithm [52], the Equation (20) algorithm [53], and the k-means algorithm in Section 3.2 on real-world networks. The results are presented in Table 3. The F1-score (F1) and Normalized mutual information (NMI) were used to compare our algorithm with the reference algorithms. Our algorithm performed well on most of the networks. Furthermore, the algorithm grouped most of the nodes into the correct communities and the normalized mutual information value (NMI) reached 0.933 and 0.923, respectively, when 11 and 12 communities were divided in the American college football network.

We use the top-5000 ground-truth communities of the Amazon copurchase and the YouTube networks provided by SNAP [45]. We compared the experimental results of our proposed algorithm with the weighted version of LPA (WLPA) [48] on these real-world networks. As shown in Table 4, we can see that the DDJKM algorithm performed well. The score of DDJKM on the Amazon network is slightly lower than of WLAP, but its score on the YouTube is higher than that of WLAP, and the mixing (μ) of the YouTube network is higher than the Amazon network, i.e., up to 0.840, which indicates that our algorithm can also achieve good community detection results on a highly-mixed network.

Table 3. Experimental results (F , NMI) of the community detection algorithm. The best results are marked in bold.

Network		**GN**	**FG**	**MIGA**	**SLC**	**Equation (20)**	**k-means**	**DDJKM**	
Karate	$\|\boldsymbol{c}\|$	2	2	2	2	2	2	2	
	$F1$	0.970	0.971	**1**	0.971	**1**	0.879	**1**	
	NMI	0.836	0.837	**1**	0.837	**1**	0.666	**1**	
Dolphins	$\|c\|$	2	2	2	2	2	2	2	
	$F1$	0.980	0.937	0.965	0.980	0.961	0.770	**0.982**	
	NMI	**0.890**	0.652	0.814	**0.890**	0.814	0.417	0.889	
Polbooks	$\|c\|$	3	3	3	3	3	3	2	3
	$F1$	0.808	0.725	0.797	0.798	**0.829**	0.655	0.784	0.726

	NMI	0.568	0.568	0.585	**0.584**	0.597	0.454	0.571	0.530
Football	$\lvert c \rvert$	12	12	12	12	12	12	11	12
	$F1$	0.802	0.528	0.864	0.846	0.859	0.730	**0.920**	0.885
	NMI	0.878	0.697	0.916	0.793	0.865	0.822	**0.933**	0.923

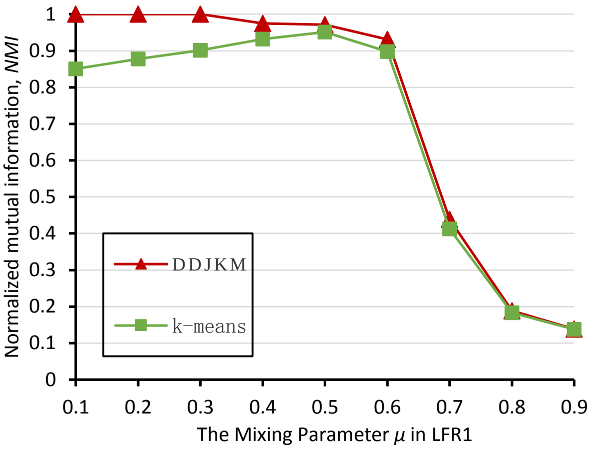
Table 4. Experimental results (F , NMI) of the community detection algorithm. The best results are marked in bold.

Network		WLPA	DDJKM
Amazon	$F1$	**0.582**	0.554
	NMI	**0.761**	0.755
YouTube	$F1$	0.273	**0.482**
	NMI	0.547	**0.625**

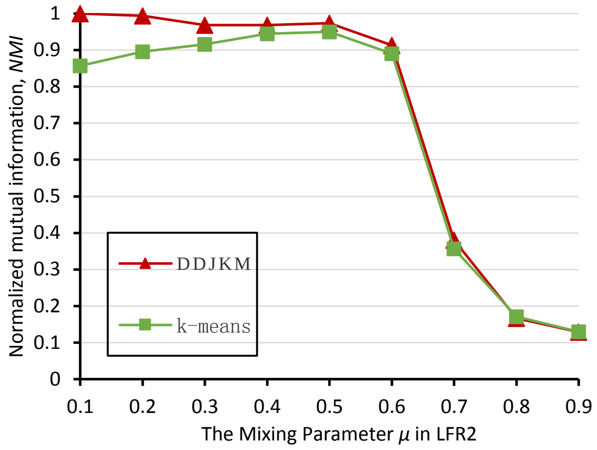
4.3.2. Experiments on LFR Benchmark Networks

Next, we used LFR networks LFR1, LFR2, and LFR3 to test DDJKM and the k-means algorithm described in Section 3.2. Because the results of the k-means algorithm are different each time, we took the average of the results of the above three networks and ran them 20 times using these algorithms.

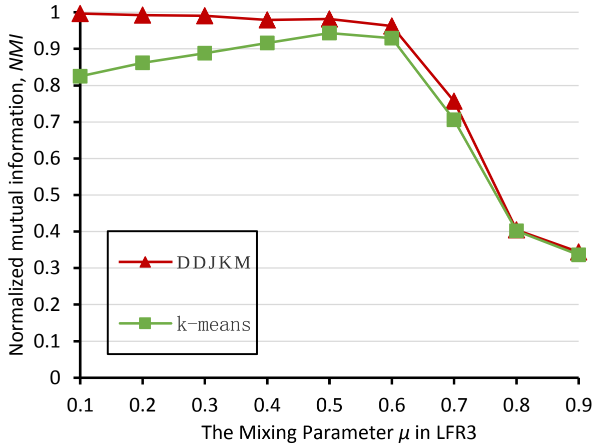
Figure 12 shows the results of our algorithm and the k-means algorithm on the LFR1, LFR2, and LFR3 networks; the DDJKM results showed the best performance. The DDJKM algorithm performs well in the range of $\mu < 0.6$, and with an increase of μ , the DDJKM algorithm was stable on the LFR network of 1000, 2000, and 5000 nodes, and there is no significant difference in the performance of the network with different numbers of nodes and community scales. This means that the DDJKM algorithm is stable in the dense network, and is not affected by the number of nodes or the community scale. However, when $\mu > 0.6$, the NMI value of DDJKM and the k-means algorithms running on the three computer-generated networks suddenly drop everything, because the community structure is less obvious as the mixing parameters increase, causing too many nodes to merge into the same community. Therefore, the accuracy of the algorithms continues to decrease.



(**a**)



(**b**)



(**c**)

Figure 12. Values of NMI over the 20 runs on (**a**) LFR1, (**b**) LFR2, and (**c**) LFR3.

On the LFR (LFR4) network of 5000 nodes, we ran some of the known community detection algorithms, i.e., Newman's fast greedy algorithm (FG), Louvain (Lvn) [10], Label Propagation (LPA) [12], PCN, and PSC [54] and compared their results with the results of our algorithms. We generated 100 LFR networks per μ value, ran the algorithms on all the 100 generated datasets, and averaged the results for each algorithm. The results of the NMI performance are shown in Figure 13. We present the detailed results of the algorithms on the LFR4 networks of 5000 nodes in Table 5. On the networks generated with higher mixing values (i.e., μ > 0.8), our algorithm with PCN and PSC was among the top four best performing algorithms according to the NMI values; our algorithm has slightly lower accuracy than PCN and PSC when the mixing parameters are high; on most networks, PCN, PSC, and our algorithm yield the best results; Newman's algorithm and the Louvain algorithm only have higher NMI values when the mixing value is low, as they tend to merge communities which may lead to a resolution limit [55]. The NMI value of LPA is relatively high when the mixing value is low in a large-scale network. However, with the increase of mixing values, the community structure is less obvious, and its accuracy is significantly reduced. Our algorithm can still successfully identify the community, and its performance is better than Newman's greedy fast algorithm, Louvain, and LPA.

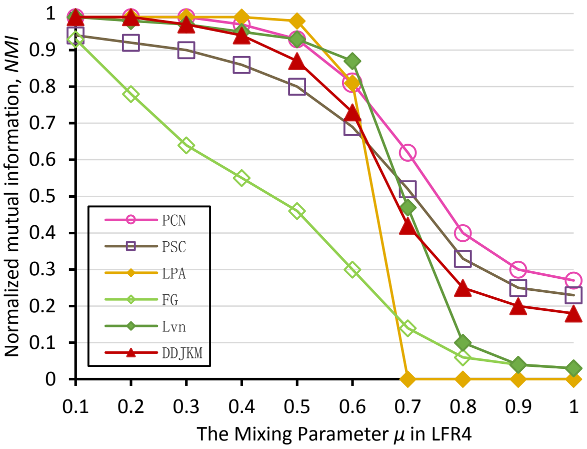


Figure 13. Comparison of our method and known algorithms on LFR4.

Table 5. Generated LFR benchmark networks of 5000 (LFR4) nodes.

$\|V\|$	μ	*NMI*					
		DDJKM	PCN	PSC	LPA	FG	Lvn
5000	0.1	0.99	0.99	0.94	0.99	0.93	0.99
5000	0.2	0.99	0.99	0.92	0.99	0.78	0.98
5000	0.3	0.97	0.99	0.90	0.99	0.64	0.97
5000	0.4	0.94	0.97	0.86	0.99	0.55	0.95
5000	0.5	0.87	0.93	0.80	0.98	0.46	0.93
5000	0.6	0.73	0.81	0.69	0.81	0.30	0.87
5000	0.7	0.42	0.62	0.52	0.00	0.14	0.47
5000	0.8	0.25	0.40	0.33	0.00	0.06	0.10
5000	0.9	0.20	0.30	0.25	0.00	0.04	0.04
5000	1.0	0.18	0.27	0.23	0.00	0.03	0.03

5. Conclusions

In this study, the concepts of CB uncertainty of nodes based on information entropy and of CB certainty of nodes as node density were defined. In addition, based on node density and degree centrality, a k-means clustering-based community detection algorithm, DDJKM, was proposed. This algorithm can select clustering centers well, thus preventing the selection of initial cluster centers which are too close to each other, and reducing the iteration times in the process. The proposed algorithm exhibited good performance in several representative, real-world networks, as well as in artificial networks. In future works, as the node density can reflect its community belongingness, nodes can be divided into two categories, i.e., with CB certainty and with CB uncertainty, so that study of community detection can focus on the detection of nodes with CB uncertainty. In this way, the number of required iterations for the community division of nodes can be effectively reduced.

Author Contributions: Conceptualization, B.C. and L.Z.; Investigation, L.Z., H.L. and Y.H.; Methodology, B.C.; Project administration, B.C.; Resources, H.L. and Y.H.; Supervision, B.C.; Validation, L.Z. and Y.W.; Writing–original draft, L.Z.; Writing–review and editing, B.C. All authors have read and approved the final manuscript.

Funding: This work was partially funded by NSFC (Grant 61802034), Soft science fund of Sichuan Province (Grant 2019JDR0117), the Open Fund of Key Laboratory of the Ministry of Education of China (Grant 13zxzk01) and the Digital Media Science Innovation Team of CDUT (Grant 10912-kytd201510).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Watts, D.J. A twenty-first century science. *Nature* **2007**, *445*, 489.
2. Wang, F.Y.; Zeng, D.; Carley, K.M.; Mao, W. Social computing: From social informatics to social intelligence. *IEEE Intell. Syst*. **2007**, *22*, 79–83.
3. Wang, Y.W.; Wang, H.O.; Xiao, J.W.; Guan, Z.H. Synchronization of complex dynamical networks under recoverable attacks. *Automatica* **2010**, *46*, 197–203.
4. Papadopoulos, S.; Kompatsiaris, Y. Community detection in social media performance and application considerations. *Data Min. Knowl. Disc*. **2012**, *24*, 515–554.
5. Tong, S.C.; Li, Y.M.; Zhang, H.G. Adaptive neural network decentralized backstepping output-feedback control for nonlinear large-scale systems with time delays. *IEEE Trans. Neural Netw*. **2011**, *22*, 1073–1086.
6. Liu, Y.; Moser, J.; Aviyente, S. Network community structure detection for directional neural networks inferred from multichannel multisubject EEG data. *IEEE Trans. Biomed. Eng*. **2014**, *61*, 1919–1930.
7. Perianes-Rodríguez, A.; Olmeda-Gómez, C.; Moya-Anegón, F. Detecting, identifying and visualizing research groups in co-authorship networks. *Scientometrics* **2010**, *82*, 307–319.
8. Fortunato, S. Community detection in graphs. *Phys. Rep*. **2010**, *486*, 75–174.
9. Liu, C.; Du, Y.; Lei, J. A SOM-Based Membrane Optimization Algorithm for Community Detection. *Entropy* **2019**, *21*, 533.
10. Blondel, V.D.; Guillaume, J.-L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech*. **2008**, *2008*, P10008.
11. Barber, M.J.; Clark, J.W. Detecting network communities by propagating labels under constraints. *Phys. Rev. E* **2009**, *80*, 026129.
12. Raghavan, U.N.; Albert, R.; Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **2007**, *76*, 036106.
13. Šubelj, L.; Bajec, M. Ubiquitousness of link-density and link-pattern communities in real-world networks. *Eur. Phys. J. B* **2012**, *85*, 1–11.
14. Jin, H.; Wang, S.; Li, C. Community detection in complex networks by density-based clustering. *Phys. A* **2013**, *392*, 4606–4618.
15. Gong, M.; Liu, J. Novel heuristic density-based method for community detection in networks. *Phys. A* **2014**, *403*, 71–84.
16. Zhou, H. Distance, dissimilarity index, and network community structure. *Phys. Rev. E* **2003**, *67*, 061901.
17. Rosvall, M.; Bergstrom, C.T. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. USA* **2008**, *105*, 1118–1123.
18. MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, USA, 21–27 July 1965, 27 December–7 January 1965; Le Cam, L.M., Neyman, J., Eds.; University of California Press: Berkeley, CA, USA, 1967; Volume 1, pp. 281–297.
19. Jiang, Y.; Jia, C.; Yu, J. An efficient community detection method based on rank centrality. *Phys. A* **2013**, *392*, 2182–2194.
20. Li, Y.; Jia, C. A parameter-free community detection method based on centrality and dispersion of nodes in complex networks. *Phys. A* **2015**, *438*, 321–334.
21. Wang, T.; Wang, H. A novel cosine distance for detecting communities in complex networks. *Phys. A* **2015**, *437*, 21–35.
22. Popat, S.K.; Emmanuel, M. Review and comparative study of clustering techniques. *Int. J. Comput. Sci. Inf. Technol.* **2014**, *5*, 805–812.
23. Celebi, M.E.; Kingravi, H.A.; Vela, P.A. A comparative study of efficient initialization methods for the k-means clustering al-gorithm. *Expert Syst. Appl.* **2013**, *40*, 200–210, doi:10. 1016/j.eswa.2012.07.021.
24. Girvan, M.; Newman, M.E. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826.
25. Bilal, S.; Abdelouahab, M. Node similarity and modularity for finding communities in networks. *Phys. A Stat. Mech. Its Appl.* **2018**, *492*, 1958–1966.
26. Lü, L.; Zhou, T. Link prediction in complex networks: A survey. *Phys. A* **2011**, *390*, 1150–1170.

27. Salton, G.; Mcgill, M.J. *Introduction to Modern Information Retrieval*; McGraw–Hill: New York, NY, USA, 1983.
28. Jaccard, P. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin De La Societe Vaudoise des Science Naturelles* **1901**, *37*, 547.
29. Srensen, T. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Videnski Selsk Biol. Skr*. **1948**, *5*, 1–34.
30. Ravasz, E.; Somera, A.L.; Mongru, D.A.; Oltvai, Z.N.; Barabasi, A.L. Hierarchical organization of modularity in metabolic networks. *Science* **2002**, *297*, 1551–1555.
31. Barabasi, A.L.; Albert, R. Emergence of scaling in random networks. *Science* **1999**, *286*, 509–512.
32. Adamic, L.; Adar, E. Friends and neighbors on the web. *Soc. Netw.* **2003**, *25*, 211–230.
33. Zhou, T.; Lü, L.; Zhang, Y.C. Predicting missing links via local information. *Eur. Phys. J. B*. **2009**, *71*, 623.
34. Pons, P.; Latapy, M. Computing Communities in Large Networks Using Random Walks. *J. Graph Algorithms Appl.* **2006**, *10*, 191–218.
35. De Meo, P.; Ferrara, E.; Fiumara, G.; Provetti, A. Mixing local and global information for community detection in large networks. *J. Comput. Syst. Sci.* **2014**, *80*, 72–87.
36. Okuda, M.; Satoh, S.; Iwasawa, S.; Yoshida, S.; Kidawara, Y.; Sato, Y. Community detection using random-walk similarity and application to image clustering. *ICIP* **2017**, 1292–1296.
37. Bao, Z.K.; Ma, C.; Xiang, B.B.; Zhang, H.F. Identification of influential nodes in complex networks: Method from spreading probability viewpoint. *Phys. A Stat. Mech. Its Appl.* **2017**, *468*, 391–397.
38. Cai, B.; Tuo, X.G.; Yang, K.X.; Liu, M.Z. Community centrality for node's influential ranking in complex network. *Int. J. Mod. Phys. C* **2014**, *25*, 1350096.
39. Newman, M.E.; Reinert, G. Estimating the Number of Communities in a Network. *Phys. Rev.Lett*. **2016**, *117*.
40. Zachary, W.W. An information flow model for conflict and fission in small groups. *J. Anthropol. Res*. **1977**, *33*, 452–473.
41. Lusseau, D.; Schneider, K.; Boisseau, O.J.; Haase, P.; Slooten, E.; Dawson, S.M. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behav. Ecol. Sociobiol.* **2003**, *54*, 396–405.
42. Newman, M. Mark Newman’s Network Data Collection. Available online: http://www-personal.umich.edu/~mejn/netdata (accessed on 25 August 2019).
43. Newman, M. Modularity and community structure in networks. APS March Meeting. *American Physical Society* **2006**, *103*, 8577–8582.
44. Jiang, J.Q.; McQuay, L.J. Modularity functions maximization with nonnegative relaxation facilitates community detection in networks. *Phys. A* **2012**, *391*, 854–865.
45. Leskovec, J.; Krevl, A. SNAP Datasets: Stanford Large Network Dataset Collection. Available online: http://snap.stanford.edu/data (accessed on 25 August 2019).
46. Gong, M.G.; Fu, B.; Jiao, L.C.; Du, H.F. Memetic algorithm for community detection in networks. *Phys. Rev. E* **2011**, 006100.
47. Danon, L.; Díaz-Guilera, A.; Duch, J.; Arenas, A. Comparing community structure identification. *J. Stat. Mech. Theory Exp.* **2005**, P09008.
48. Hu, Y.; Yang, B. Characterizing the structure of large real networks to improve community detection. *Neural Comput. Appl.* **2017**, *28*, 2321–2333.
49. Lancichinetti, A.; Fortunato, S.; Radicchi, F. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **2008**, *78*, 046110.
50. Clauset, A.; Newman, M.E.J.; Moore, C. Finding community structure in very large networks. *Phys. Rev. E* **2010**, *70*, 264–277.
51. Mahmood, A.; Small, M. Subspace based network community detection using sparse linear coding. *IEEE Trans. Knowl. Data Eng*. **2016**, *28*, 801–812.
52. Shang, R.; Bai, J.; Jiao, L.; Jin, C. Community detection based on modularity and an improved genetic;algorithm. *Phys. Stat. Mech. Its Appl.* **2013**, *392*, 1215–1231.
53. Tian, B.; Li, W. Community Detection Method Based on Mixed-norm Sparse Subspace Clustering. *Neurocomputing* **2018**, *275*, 2150–2161.

54. Tasgin, M.; Bingol, H.O. Community detection using preference networks. *Phys. Stat. Mech. Its Appl.* **2017**, *495*, 126–136.

55. Fortunato, S.; Barthélemy, M. Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA* **2007**, *104*, 36–41.