A Method for Community Detection of Complex Networks Based on Hierarchical Clustering

Chuantao Yin,1,2 Shuaibing Zhu,1 Hui Chen,1 Bingxue Zhang,3 and Bertrand David3

1Sino-French Engineer School, Beihang University, Beijing 100191, China
2Research Institute of Beihang University in Shenzhen, Shenzhen 518057, China
3LIRIS, École Centrale de Lyon, 69130 Écully, France

Correspondence should be addressed to Chuantao Yin; chuantao.yin@buaa.edu.cn

Received 9 January 2015; Accepted 10 April 2015

Academic Editor: Shuai Li

Copyright © 2015 Chuantao Yin 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.

Due to the development and popularization of Internet, there is more and more research focusing on complex networks. Research shows that there exists community structure in complex networks. Connections within a community are dense, while connections between different communities are rare. Generally, a community consists of nodes having similar properties, so finding communities helps to mine some useful information of complex networks, for example, finding a group of people having common interests. How to detect communities in complex networks is a hotspot in recent years.

Newman and Girvan [6] put forward the conception of modularity to measure the quality of algorithms for community detection. The modularity $Q$ is a number less than 1. The bigger $Q$ is, the better quality is. Modularity has become the most widely used assessment criterion for community detection due to two reasons when we do not know the real communities in networks. Firstly, modularity is very convenient to use. Secondly, it seems that the result is better when the modularity is bigger in most cases. Many algorithms [7–9] detect communities by maximizing modularity. As it is impossible to calculate the modularity for every possible case, such kind of algorithms usually adopts greedy strategy to find the case almost maximizing modularity.

As modularity becomes more and more popular, people begin to doubt its effectiveness. Fortunato and Barthélemy [10] said algorithms based on modularity are not able to detect communities whose size is less than an inherent size depending on the number of networks’ edges. It is called by Fortunato the resolution limit of modularity. We find out by experiment that the quality is not necessarily better when the modularity is bigger. Maximizing modularity tends to
detect more communities than the real situation, because the upper boundary of modularity is bigger when we detect more communities. In order to overcome this problem, this paper proposes the conception of extensive modularity on the basis of modularity. The upper boundary of extensive modularity is independent of the number of communities that we detect. We find out that the result maximizing extensive modularity is better than that maximizing modularity.

For now, there are many algorithms having advantages and disadvantages for community detection. Algorithms based on graph theory appear early, and their main idea is to divide the network into \( k \) such parts of almost the same size satisfying that connections in the same part are dense and connections between different parts are rare. Kernighan-Lin Algorithm [11] and bisection of spectrum based on Laplace eigenvalues [12] are representative algorithms based on graph theory. Generally speaking, it is hard to know the number of communities ahead of time, but algorithms based on graph theory need the number, so the accuracy is very low. Algorithms based on hierarchical clustering are common to see for now. This kind of algorithms decomposes datasets hierarchically according to a certain method until it satisfies some certain conditions. According to the principle of classification, algorithms based on hierarchical clustering can be divided into two parts, which are aggregation algorithm and splitting algorithm. GN Algorithm [6] is a representative splitting algorithm, while CNM Algorithm [13] and Newman Fast Algorithm [14] are representative aggregation algorithms. GN and CNM can usually obtain large value of modularity, while the accuracy is not necessarily high. LCV Algorithm [15] detects communities by finding local central vertices. EACH Algorithm [16] detects communities by calculating the edge antitriangle centrality with the isolated vertex handling strategy. LCV and EACH are both algorithms based on hierarchical clustering. Besides, there are also other kinds of algorithms.

The rest of this paper is organized as follows: Section 2 introduces some definitions including central nodes, nodes’ similarity, link strength between a node and a community, modularity, and extensive modularity; Section 3 describes detailed steps of the algorithm proposed by this paper; Section 4 states experiment results; the last section is our conclusion.

2. Definitions for Community Detection

Notation of Main Symbols gives the meaning of the main symbols that we will use in the whole paper.

2.1. Central Node. Central nodes include global central nodes and local central nodes. In a network, \( k \) nodes with maximal degree are called its global central nodes, in which the value of \( k \) can be determined according to our needs. If a node’s degree is bigger than the degree of all its adjacent nodes, we call it local central nodes.

Global central nodes play an important role in the network and are probably important nodes of communities. So global central nodes help to find communities.

2.2. Nodes’ Similarity. We define the similarity of node \( a \) and node \( b \) as

\[
\text{Sim}(a, b) = \frac{|L(a) \cap L(b)|}{|L(a)| + |L(b)|} = \frac{2|L(a) \cap L(b)|}{|L(a)| + |L(b)|}
\]

in which \( L(a) = \text{Adj}(a) \cup \{a\} \). The bigger Sim(a, b) is, the bigger similarity of node \( a \) and node \( b \) is. If two nodes have a large similarity, they are probably in the same community. Paper [16] defines the similarity of node \( a \) and node \( b \) as

\[
\text{Sim}(a, b) = \frac{\lVert \text{Adj}(a) \cap \text{Adj}(b) \rVert}{\lVert \text{Adj}(a) \cup \text{Adj}(b) \rVert}.
\]

This definition has some disadvantages. Firstly, calculating the union set consumes more time than calculating the sum of two numbers. Secondly, we can know intuitively that the similarity \( S1 \) of node 2 and node 4 in Figure 1 should be bigger than the similarity \( S2 \) of node 2 and node 4 in Figure 2. However, according to this definition, we have \( S1 = 1/5 < S2 = 1/3 \), which does not conform to our intuition. According to our definition, we have \( S1 = 2/4 > S2 = 1/3 \), which conforms to our intuition.

2.3. Link Strength between a Node and a Community. We define the link strength between node \( i \) and community \( C \) (\( i \notin C \)) as

\[
\text{Link}[i][C] = 2 \cdot d[i][C] + \sum_{j \in \text{Adj}(i), j \notin C} d[j][C].
\]

Figure 1: Network 1.

Figure 2: Network 2.
We multiply \(d[i][C]\) by 2, because a direct link between node \(i\) and community \(C\) is stronger than an indirect link. If the link strength of node \(i\) and community \(C\) is large, node \(i\) should be probably in community \(C\).

### 2.4. Modularity \(Q\)

The modularity \(Q\) is defined as

\[
Q = \sum_{C_i} (e_{ii} - a_i^2) = \sum_{C_i} e_{ii} - \sum_{C_i} a_i^2
\]

\[
= \frac{1}{m} \sum_{C_i} d_{in}(C_i) - \frac{1}{m} \sum_{C_i} \left[ \frac{1}{2} d_{ext}(C_i) + d_{in}(C_i) \right]^2.
\]

\(Q\) is a number less than 1. The bigger \(Q\) is, the better algorithm's quality is. The quality is generally good when \(Q\) is between 0.3 and 0.7. \(Q\) is difficult to exceed 0.7 in the real situation.

### 2.5. Extensive Modularity \(eQ\)

We define extensive modularity \(eQ\) as

\[
eQ = \begin{cases} 
0, & c = 1 \\
\frac{Q}{1 - 1/c} = \frac{cQ}{c - 1}, & c \neq 1.
\end{cases}
\]

Let us analyze the upper boundary of modularity:

\[
Q = \sum_{C_i} (e_{ii} - a_i^2) = \sum_{C_i} e_{ii} - \sum_{C_i} a_i^2 \leq 1 - \frac{1}{c}.
\]

Through (6), we can know that \(1 - 1/c\) is a more accurate estimation of the upper boundary of \(Q\), which depends on the number of communities that we detect. If we detect only 2 communities, the upper boundary of \(Q\) is 0.5, so \(Q\) is difficult to exceed 0.4. If we detect 5 communities, the upper boundary of \(Q\) is 0.8, so \(Q\) is relatively easy to reach 0.4 or even 0.5. Even if \(Q\) is larger in the latter case than that in the former case, the quality is not always better. Therefore, we define the extensive modularity to eliminate this problem. According to our definition, the upper boundary of extensive modularity is 1, which does not depend on the number of communities that we detect. The bigger \(eQ\) is, the better quality is. We will explain the advantages of extensive modularity through experiments in the following paragraphs.

### 3. Algorithm Description

One city has one or several centers and is expanded layer by layer around the centers. The closer to the center the layer is, the denser the connections to the layer are. The outer layer is connected mainly by itself and inner layers. Inspired by the hierarchical structure of cities, we consider that a community should have one or several global central nodes and should be expanded layer by layer around the global central nodes. The nodes in layer \(p\) are mainly connected by nodes in layer \(p - 1\). According to such community structure, we propose a method for community detection based on hierarchical clustering (CDHC). The process of CDHC Algorithm includes initializing communities, expanding communities, merging small communities and choosing the best result.

We suppose the network is connected, which means we can reach any node from any node through some edges in the network.

#### 3.1. Initialize Communities

The process of initializing communities can be divided into three steps, which are sorting nodes, choosing global central nodes, and merging global central nodes into communities.

**Step 1.** Sort all the nodes by degree in descending order.

**Step 2.** Choose \(k\) nodes with maximal degree as global central nodes of the network. The value of \(k\) is determined by the following equations:

\[
k = \max \{k_1, k_2\},
\]

\[
k_1 = \frac{n}{10} + 2, \quad k_2 = \left| \left\{ v \mid d(v) > \frac{\max \{d(v)\}}{2} \right\} \right|.
\]

The idea is that we want to choose a small number of global central nodes to reduce the quantity of calculation and the global central nodes include the important nodes of each community. If there are only a small number of nodes with large degree and the degree of most nodes is small, we generally have \(k_1 > k_2\), so \(k = k_1\). However, it is probable that the degree of all nodes is very close, and we have to choose most nodes as global central nodes. In this case, we should take \(k = k_2\).

**Step 3.** Initialize the first community, assign the node with maximal degree to the first community, and then mark it as the first community’s central node. For each node \(v\) of the remaining \(k - 1\) global central nodes, we calculate its similarity with each initialized community’s central node. If there exists a similarity bigger than some threshold \(\eta\), assign the node \(v\) to the community maximizing the similarity; otherwise, initialize a new community, assign the node \(v\) to the new community, and mark the node \(v\) as the new community’s central node.

Algorithm 1 shows the process of initializing communities.

#### 3.2. Expand Communities

After finishing the process of initializing communities, we need expand these communities. The process of expanding communities includes marking nodes’ level and calculating link strength.

**Step 1.** Mark all the global central nodes of the network as the first level. If the node \(v\) is connected to a node of the first level and its level is not marked, mark its level as two. If the
node \( v \) is connected to a node of level \( p \) and its level is not marked, mark its level as \( p + 1 \). As we suppose the network is connected, all the nodes’ level can be marked if we repeat the previous operation.

Step 2. Nodes of the first level have already been assigned to communities. For each node \( v \) of level two, we calculate the link strength between the node \( v \) and each community \( C \). Choose the community maximizing the link strength and then assign the node \( v \) to it. By parity of reasoning, for each node \( v \) of level, we repeat the previous operation, until every level’s nodes have been assigned to communities.

Algorithm 2 shows the process of expanding communities.

### 3.3. Merge Small Communities

After the previous two processes, every node is assigned to a community. Some communities are probably of very small size, and we need to merge these communities into the other communities. This process includes determining small communities and calculating the number of common nodes.

**Step 1.** If a community is of size smaller than \( t \), we call it small community; otherwise, we call it large community. The value of \( t \) is determined by the following equation:

\[
t = \max \left\{ 5, \frac{\max |C_i|}{10} \right\}.
\]

We determine the value of \( t \) on the basis of two reasons. Firstly, we deem that communities of size smaller than 5 make no sense. Secondly, when the size of the biggest community is very large, communities of size slightly bigger than 5 have no meaning to exist, just like it is meaningless to compare the size of a city with that of a village.

For the threshold \( \eta \), take at regular interval ten values from 0.1 to 1, and repeat the previous three processes for each value. Then we obtain ten partitions of the network. Generally, we are not able to know the real communities, so we have to choose the best result through an assessment criterion.

**Algorithm 1:** Initialize communities.

```plaintext
input: network \( G \); sort nodes by degree in descending order; choose \( k \) global central nodes \( \{v[1], \ldots, v[k]\} \); initialize community \( C[1] \); nc = 1; //nc: number of communities; assign \( v[1] \) to \( C[1] \); mark \( v[1] \) as central node of \( C[1] \); for \( i = 2 \) to \( k 
```

```plaintext
for \( i = 1 \) to \( n \) \{ \( \text{lev}[i] = 0; \} //initialize nodes’ level
for \( i = 1 \) to \( k \) \{ \( \text{lev}[i] = 1; \}
add 1, 2, \ldots, \( k \) to stack;
foreach \( j \) in \( \text{adj}(i) \); //adj(i): adjacent nodes of \( i \)
if \( \text{lev}[j] = 0 \)
\( \text{lev}[j] = \text{lev}[i] + 1; \)
add \( j \) to stack;
foreach \( v \) in level \( \text{lev} \)
for \( i = 1 \) to \( \text{nc} \)
calculate \( \text{link}[v][C[i]] \);
mark the pos of the largest number;
assign \( v \) to \( C[\text{max_pos}] \);
```

**Algorithm 3:** Merge little communities.

```plaintext
output: communities \( C = \{ C[1], C[2], \ldots, C[c] \} \); determine little communities and large communities;
foreach node \( v \) in each little communities
foreach large community \( C \)
calculate the number of common nodes;
between \( \text{adj}(v) \) and \( C \);
mark the pos of the largest number;
reassign \( v \) to \( C[\text{max_pos}] \);
```

**Algorithm 2:** Expand communities.
now, modularity is the most widely used criterion. However, we find that the result maximizing the extensive modularity is usually better than the result maximizing the modularity and conforms to the real situation very well. So we choose the partition with maximal value of extensive modularity as the best result.

Now we analyze the time complexity of CDHC. The first process costs $O(n \ln n) + O(kc)$. The second process costs $O(m)$, and the third process costs no more than $O(m)$. So the time complexity of CDHC is $O(n \ln n) + O(m) + O(kc)$. In the worst case, $k$ and $c$ are close to $n$, and the time complexity is $O(n^2)$.

4. Experiments

Firstly, we choose three most widely used datasets, which are Zachary Karate Club [17], Dolphin Network [18], and American College Football [19]. Secondly, we introduce some assessment criteria used in the case that the real communities are known. Thirdly, we verify the advantage of extensive modularity on the three datasets. Lastly, we choose some classical or new algorithms and compare them with CDHC on the three datasets.

4.1. Datasets

4.1.1. Zachary Karate Club. It is a social network of friendships between 34 members of a karate club at a US university in the 1970s. A factional division leads to a formal separation of the club into two organizations of nearly equal size.

4.1.2. Dolphin Network. It is an undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound in New Zealand. It contains totally 62 dolphins in two groups.

4.1.3. American College Football. It is a network of American football games between Division IA colleges during regular season fall 2000. It contains 115 teams, which are divided into 12 conferences, each containing 8 to 12 teams.

Table 1 is a summary of the three datasets’ basic information, including NOC (number of communities), number of nodes, and number of edges.

| Dataset    | NOC | Nodes | Edges |
|------------|-----|-------|-------|
| Zachary    | 2   | 34    | 78    |
| Dolphins   | 2   | 62    | 139   |
| Football   | 12  | 115   | 613   |

Table 1: Dataset information.

if we consider every node as a community, the precision will be 1. However, it is obviously not a good result. Generally, if we detect more communities, the precision tends to be larger.

4.2. Assessment Criteria. Given a network $G, T = \{T_1, T_2, T_3, \ldots, T_q\}$ is its real communities, while $R = \{R_1, R_2, R_3, \ldots, R_p\}$ is its detected communities. Here we introduce four assessment criteria to evaluate the overlap ratio of the real communities and the detected communities.

4.2.1. Precision [20]. Precision is defined as

$$\text{precision}(R, T) = \frac{\sum_{j=1}^{p} \max_{\gamma}( |R_{\gamma} \cap T_{j}| / |R_{\gamma}| )}{p}.$$  \hspace{1cm} (9)

Precision is a number between 0 and 1. It means the ration between the retrieved correct items and the retrieved items. It is not a good criterion for community detection. For example, if we consider every node as a community, the precision will be 1. However, it is obviously not a good result. Generally, if we detect more communities, the precision tends to be larger.

4.2.2. Recall [20]. Recall is defined as

$$\text{recall}(R, T) = \frac{\sum_{j=1}^{q} \max_{\gamma}( |R_{\gamma} \cap T_{j}| / |T_{j}| )}{q}.$$  \hspace{1cm} (10)

Recall is a number between 0 and 1. It means the ratio between the retrieved correct items and the correct items. It is also not a good criterion for community detection. For example, if we only detect one community, which is the whole network, the precision will be 1. However, it is obviously not a good result. Generally, if we detect fewer communities, the recall tends to be larger.

4.2.3. F-Measure [21]. F-measure is defined as

$$F\text{-measure}(R, T) = \frac{2}{1/\text{precision}(R, T) + 1/\text{recall}(R, T)},$$  \hspace{1cm} (11)

which is

$$F\text{-measure}(R, T) = \frac{2 \cdot \text{precision}(R, T) \cdot \text{recall}(R, T)}{\text{precision}(R, T) + \text{recall}(R, T)}.$$  \hspace{1cm} (12)

F-measure is a number between 0 and 1. We can see that F-measure is the harmonic average of precision and recall. If F-measure is large, precision and recall need to have a relatively large value. According to what we discuss about precision and recall, if we want a large value of F-measure, the number of communities should not be too large or too small, and the detected communities must be very close to the real communities.

4.2.4. NMI [22] (Normalized Mutual Information). NMI is defined as

$$\text{NMI}(R, T) = - \frac{\sum_{\gamma=1}^{p} \sum_{j=1}^{q} |R_{\gamma} \cap T_{j}| \ln \left( \frac{\left( n \cdot |R_{\gamma} \cap T_{j}| \right) / \left( |R_{\gamma}| \cdot |T_{j}| \right)}{\sum_{\gamma=1}^{p} |R_{\gamma}| \ln (n) \cdot \sum_{j=1}^{q} |T_{j}| \ln (|T_{j}| / n)} \right)}{\sqrt{\sum_{\gamma=1}^{p} |R_{\gamma}| \ln (|R_{\gamma}| / n) \cdot \sum_{j=1}^{q} |T_{j}| \ln (|T_{j}| / n)}}.$$  \hspace{1cm} (13)

NMI is a number between 0 and 1. The larger NMI is, the better result is. NMI is a frequently used criterion, but the calculation is relatively complex.

According to our discussion about the four criteria, we will take F-measure and NMI to evaluate the result of community detection.
Table 2: Experimental results under different threshold $\eta$ on the dataset Zachary.

| $\eta$ | NOC | $Q$ | $eQ$ | $F$-measure | NMI  |
|-------|-----|-----|------|-------------|------|
| 0.1   | 2   | 0.00| 0.00 | 0.692       | 0.000|
| 0.2   | 2   | 0.372| 0.744| 1.000       | 1.000|
| 0.3   | 2.032| 0.704| 0.919| 0.650       |
| 0.4   | 2.026| 0.512| 0.805| 0.373       |
| 0.5   | 2.026| 0.512| 0.805| 0.373       |
| 0.6   | 3    | 0.373| 0.560| 0.884       | 0.831|
| 0.7   | 4    | 0.308| 0.410| 0.945       | 0.724|
| 0.8   | 5    | 0.272| 0.340| 0.811       | 0.623|
| 0.9   | 5    | 0.272| 0.340| 0.812       | 0.621|
| 1.0   | 5    | 0.272| 0.340| 0.812       | 0.621|

Table 3: Experimental results under different threshold $\eta$ on the dataset Dolphin.

| $\eta$ | NOC | $Q$ | $eQ$ | $F$-measure | NMI  |
|-------|-----|-----|------|-------------|------|
| 0.1   | 2   | 0.372| 0.744| 1.000       | 1.000|
| 0.2   | 2.032| 0.704| 0.919| 0.650       |
| 0.3   | 2.026| 0.512| 0.805| 0.373       |
| 0.4   | 2.026| 0.512| 0.805| 0.373       |
| 0.5   | 3    | 0.373| 0.560| 0.884       | 0.831|
| 0.6   | 4    | 0.308| 0.410| 0.945       | 0.724|
| 0.7   | 5    | 0.272| 0.340| 0.811       | 0.623|
| 0.8   | 5    | 0.272| 0.340| 0.812       | 0.621|
| 1.0   | 5    | 0.272| 0.340| 0.812       | 0.621|

4.3. Experimental Results on Extensive Modularity. For the threshold $\eta$, we take at regular interval ten values from 0.1 to 1, and we obtain ten different partitions, respectively, on the datasets Zachary, Dolphin, and Football. For each partition, we use NOC (number of communities), modularity $Q$, extensive modularity $eQ$, $F$-measure, and NMI to evaluate the result.

Table 2 is the ten experimental results under different threshold $\eta$ on the dataset Zachary. Seen from the modularity, the best result should be the case when the threshold is 0.6. We can see that the maximal value of modularity is 0.373. In this case, we find that the detected communities differ much from the real communities. The detected NOC is 3, while real NOC should be 2. Besides, the value of $F$-measure and NMI is not very high. Seen from extensive modularity, the best result is the case when the threshold is 0.2. The maximal value of extensive modularity is 0.770. In this case, the values of $F$-measure and NMI are very close to 1, which signifies the detected communities are very close to the real communities.

Table 3 is the ten experimental results under different threshold $\eta$ on the dataset Dolphin. Seen from the modularity, the best result is the case when the threshold is 0.3. We can see that the maximal value of modularity is 0.477. In this case, we find that the detected communities also differ much from the real communities. The detected NOC is 4, while real NOC should be 2. Besides, the value of $F$-measure and NMI is very low. Seen from extensive modularity, the best result is the case when the threshold is 0.2. The maximal value of extensive modularity is 0.770. In this case, the values of $F$-measure and NMI are very close to 1, which signifies the detected communities are very close to the real communities.

4.4. Experimental Results on Comparison of Algorithms. As we have discussed, the extensive modularity is a better criterion than modularity, so we choose the case maximizing extensive modularity as the best result and the final result. We have chosen four algorithms for comparison, which are GN Algorithm [6], CNM Algorithm [13], EACH Algorithm [16], and CDHC Algorithm [15].

Table 4 is the ten experimental results under different threshold $\eta$ on the dataset Football. Seen from modularity or extensive modularity, the best result is the case when the threshold is 0.5. In this case, the detected NOC is 11, which is very close to the real NOC 12. We see that maximizing modularity or extensive modularity does not find the best result, but it has only a slight gap with the best result.

From the discussion on the experimental results on the three datasets, the result maximizing extensive modularity usually is the best result or very close to the best result, while the result maximizing modularity is usually not very good. We can thus consider that extensive modularity is a better criterion compared to modularity.
Table 5: Experimental results of algorithms on the dataset Zachary.

| Algorithm | NOC | Q  | eQ | F-measure | NMI |
|-----------|-----|----|----|-----------|-----|
| GN        | 5   | 0.401 | 0.501 | ×         | 0.580 |
| CNM       | 3   | 0.381 | 0.572 | ×         | 0.693 |
| EACH      | 2   | 0.372 | 0.744 | ×         | 1    |
| LCV       | 2   | 0.372 | 0.744 | ×         | 1    |
| CDHC      | 2   | 0.372 | 0.744 | 1         | 1    |

Table 6: Experimental results of algorithms on the dataset Dolphin.

| Algorithm | NOC | Q  | eQ | F-measure | NMI |
|-----------|-----|----|----|-----------|-----|
| GN        | 5   | 0.519 | 0.649 | ×         | 0.554 |
| CNM       | 4   | 0.515 | 0.687 | ×         | 0.575 |
| EACH      | 4   | 0.485 | 0.647 | ×         | 0.443 |
| LCV       | 2   | 0.385 | 0.770 | 0.959     | ×    |
| CDHC      | 2   | 0.385 | 0.770 | 0.965     | 0.814 |

Table 7: Experimental results of algorithms on the dataset Football.

| Algorithm | NOC | Q  | eQ | F-measure | NMI |
|-----------|-----|----|----|-----------|-----|
| GN        | 8   | 0.595 | 0.680 | ×         | 0.831 |
| CNM       | 6   | 0.558 | 0.670 | ×         | 0.770 |
| EACH      | 11  | 0.591 | 0.650 | ×         | 0.911 |
| LCV       | 11  | 0.575 | 0.633 | 0.847     | ×    |
| CDHC      | 11  | 0.602 | 0.662 | 0.897     | 0.903 |

detected communities are very close to the real communities. LCV and CDHC are far better than GN, CNM, and EACH on Dolphin. Furthermore, CDHC is slightly better than LCV.

Table 7 is the experimental results of the five algorithms on the dataset Football. The modularity of the five algorithms differs a little. The NOC detected by GN and CNM is far less than the real NOC 12. GN gets the largest value of extensive modularity, but its NMI is relatively low. CDHC gets the second largest value of extensive modularity, and its NMI is very close to the best result. On the whole, EACH and CDHC are better on Football.

From the experimental results on the three datasets, we find that GN and CNM usually get large modularity. However, the detected communities usually differ much from the real communities. Large NOC makes it easy to get a large modularity. CDHC usually gets large extensive modularity, and the detected communities are very close to the real communities. On the whole, CDHC is far better than GN and CNM and is slightly better than EACH and LCV.

Figures 3, 4, and 5 show the communities that CDHC has detected on Zachary, Dolphin, and Football. We can see that the community structure is evident.

5. Conclusion

This paper proposes the conception of extensive modularity on the basis of modularity. The upper boundary of extensive modularity is independent on the number of communities that we detect, so it overcomes the weakness of modularity that maximizing modularity tends to detect more communities than the real situation. We verify through experiments that extensive modularity is a better criterion than modularity when the real communities are unknown.

This paper proposes CDHC Algorithm. CDHC Algorithm includes four processes, which are initializing communities, expanding communities, merging little communities, and choosing the best result. We choose four algorithms to compare with CDHC on three datasets, which are Zachary, Dolphin, and Football. We find that CDHC is far better than GN and CNM and is slightly better than LCV and EACH.
According to F-measure and NMI, the communities detected by CDHC are very close to the real communities.

**Notation of Main Symbols**

- **G**: Network
- **V**: Set of nodes
- **E**: Set of edges
- **n**: Number of nodes
- **m**: Number of edges
- **d(v)**: Degree of node **v**
- **Adj(v)**: Adjacent nodes of node **v**
- **d[v][C]**: Edges between node **v** and community **C**
- **d_{in}(C)**: Edges in community **C**
- **d_{ext}(C)**: Edges between community **C** and other communities
- **c**: Number of communities
- **k**: Number of global central nodes
- **max()**: Maximal number in the brackets

**Conflict of Interests**

The authors declare that there is no conflict of interests regarding the publication of this paper.

**Acknowledgments**

The work is supported by National Natural Science Foundation of China (no. 61402028) and Science Foundation of Shenzhen City in China (JCYJ20140509150917445).

**References**

[1] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins, “The web as a graph: measurements, models, and methods,” in *Computing and Combinatorics*, vol. 1627 of *Lecture Notes in Computer Science*, pp. 1–17, Springer, Berlin, Germany, 1999.

[2] S. Redner, “Citation statistics from 110 years of physical review,” *Physics Today*, vol. 58, no. 6, pp. 49–54, 2005.

[3] C. Castellano, S. Fortunato, and V. Loreto, “Statistical physics of social dynamics,” *Reviews of Modern Physics*, vol. 81, no. 2, pp. 591–646, 2009.

[4] A. Lancichinetti, J. Saramäki, M. Kivelä, and S. Fortunato, “Characterizing the community structure of complex networks,” *PLoS ONE*, vol. 5, no. 8, Article ID e1976, 2010.

[5] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Paris, “Defining and identifying communities in networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 9, pp. 2658–2663, 2004.

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

[7] M. Arab and M. Afsharchi, "A modularity maximization algorithm for community detection in social networks with low time complexity," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT ’12)*, pp. 480–487, IEEE, December 2012.

[8] T. N. Dinh and M. T. Thai, "Community detection in scale-free networks: approximation algorithms for maximizing modularity," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 6, pp. 997–1006, 2013.

[9] V. D. Blondel, J.-L. 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.

[10] S. Fortunato and M. Barthélemy, "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.

[11] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell System Technical Journal*, vol. 49, no. 2, pp. 291–307, 1970.

[12] A. Pothen, H. D. Simon, and K.-P. Liou, "Partitioning sparse matrices with eigenvectors of graphs," *SIAM Journal on Matrix Analysis and Applications*, vol. 11, no. 3, pp. 430–452, 1990.

[13] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, no. 6, Article ID 066111, 2004.

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

[15] Q. Chen and T.-T. Wu, "A method for local community detection by finding maximal-degree nodes," in *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC ’10)*, vol. 1, pp. 8–13, IEEE, Qingdao, China, July 2010.

[16] S. Jia, Y. Gao, L. Gao, and H. Wang, “Anti-triangle centrality-based community detection in complex networks,” *IET Systems Biology*, vol. 8, no. 3, pp. 116–125, 2014.

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

[18] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of..."
doubtful sound features a large proportion of long-lasting associations: can geographic isolation explain this unique trait?" Behavioral Ecology and Sociobiology, vol. 54, no. 4, pp. 396–405, 2003.

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

[20] Q. Chen and M. Fang, "Community detection based on local central vertices of complex networks," in Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC '11), pp. 920–925, IEEE, Guilin, China, July 2011.

[21] X. Li, M. K. Ng, and Y. Ye, "MultiComm: finding community structure in multi-dimensional networks," IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 4, pp. 929–941, 2014.

[22] M. G. Quiles, E. R. Zorzal, and E. E. N. Macau, "A dynamical model for community detection in complex networks," in Proceedings of the International Joint Conference on Neural Networks (IJCNN '13), pp. 1–8, IEEE, August 2013.
Submit your manuscripts at http://www.hindawi.com