Detecting Communities with Different Sizes for Social Network Analysis

Lihua Zhou
Department of Computer Science, Yunnan University, Kunming 650091, China
Kevin Lü
Brunel University, Uxbridge, UB8 3PH, UK

Online community detection is essential for social network analysis. Modularity is a quality function used to measure the strength of the community structure discovered with social networks. Existing methods detect community structures through modularity analysis. However, the existing modularities are unable to identify community structures correctly when communities are very different in size, in particular, when the size of some communities is very small compared to others. To address this problem, we propose the concept of a *coupling coefficient* between two communities and define a new modularity, *MC modularity*, to evaluate the quality of the discovered community structures. This method can provide adequate measures for the quality of community structures. In addition, we develop the *DC_MC algorithm* to detect community structures based on the concept of *MC modularity*. An algorithm for shifting a group of nodes, instead of one node at a time, amongst communities also is designed to achieve optimal results. The effectiveness and efficacy of the approach we proposed have been demonstrated through a set of experiments involving real data benchmarks and synthetic data sets that are purpose-built for evaluating different types of networks.

Keywords: social network; community detection; modularity; modularity optimization

1. Introduction

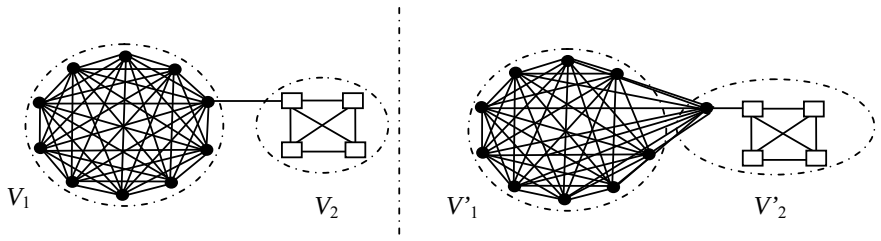
Online community studies in social networks have gained significant attention recently due to the popularity of online social media. Social networks can be represented by graphs where vertices represent individuals and edges represent relationships and interactions amongst individuals. Meanwhile, to understand community structures within social networks it is essential to have both a visual and mathematical analysis of relationships [1],[2],[3]. From the perspective of topological structures of a network, communities are groups (or clusters) of vertices that are densely interconnected, but only sparsely connected with the rest of the network [4][5]. The understandings of communities from large networks have great implications because they are closely related to the behaviour of social groups in a social network. Therefore, community detection is critical to gain an understanding of the features and other aspects of networks, and to reveal insightful functions and properties [6].

One difficulty in the identification of communities within online social networks is that there could be many ways to recognise the existence of a community; it is often unknown beforehand, thus it is difficult to determine a rational way to recognise communities within a network. In order to evaluate the quality of the discovered community structures, the concept of modularity has been introduced [7]. It is based on the idea that a random graph is not expected to have a cluster structure, so the possible existence of clusters is revealed by comparisons between the actual density of edges in a subgraph and the density one would expect to have in the subgraph if the vertices of the graph were attached regardless of community structure [8]. Modularity is one attempt to understand the clustering problem, and it embeds in its compact form all the essential ingredients and questions, from the definition of "community", to the choice of a null model, to the expression of the "strength" of communities and partitions [8].

The modularity of Newman and Girvan [9], *NG modularity* for short, has been widely recognised by academic communities and it is "the most popular quality function" [8] for measuring the strength of the community structures detected. In addition, other quality functions have also been proposed for evaluating the quality of discovered community structures from different aspects of consideration, such as the *generalization of modularity* suggested by Arenas *et al*. [10], *influence-based modularity* suggested by Ghosh and Lerman [11], and *modularity density* suggested by Li *et al*. [12].

However, as observed in this study, and also as has been pointed out in a recent review of [8], the *NG modularity* [9] is not sensitive enough for detecting clusters that are comparatively small compared

to other communities in a network, even when they are actually well defined communities such as cliques (subsets whose vertices are all adjacent to each other). Let us examine the example illustrated in Figure 1. Figure 1(a) and Figure 1(b) respectively show two community structures (*Structure_1* and *Structure_2*) of a network consisting of 104 vertices. In Figure 1(a), community V_1 and V_2 contain 100 and 4 vertices that form two cliques respectively. In Figure 1(b), community V'_1 contains 99 vertices that form a clique, community V'_2 consists of 5 vertices, but they do not form a clique. Clearly, *structure_1* in Figure 1(a) is more rational than *structure_2* in Figure 1(b), but the value of *NG modularity* (its definition is shown in Section 2.1) of *structure_1* is smaller than that of *structure_2*. This shows that *NG modularity* does not provide an adequate measure for partitioning a network where communities are very different in size. Good et al. [13] mentioned that the maximum modularity of a graph generally grows if the size of the network and/or the number of (well-separated) clusters increases. Therefore, *NG modularity* is not suitable for comparing the quality of the community structure of networks where communities with very different sizes exist [8].



NG modularity of *structure_1* =0.0024 *NG modularity* of *structure_2* =0.0026

NG modularity of *structure_1* < *NG modularity* of *structure_2*

Figure 1(a). *structure_1* Figure 1(b). *structure_ 2*

Figure 1. An example to illustrate the motivation of our work

In fact, it is not uncommon that communities in a social network have varied sizes; just look at the size and populations of all the countries in the world, and so community structures within a social network usually contain communities that are very diverse in size [14][15][16]. Therefore, it is indispensable to provide a quality function that can effectively measure the community structures of networks, no matter if the communities are similar or very different in size.

In this paper, we propose a concept of a *coupling coefficient* between two communities. It is the ratio of the *link density* between two communities over the sum of the densities of these two communities. The smaller the *coupling coefficient* is, the more independent these two communities are. Also, based on the *coupling coefficient* we define a new modularity, which we name *MC modularity*, for

evaluating the quality of the discovered community structures. A unique feature of *MC modularity* is that it is suitable for measuring the partitioning quality of networks with different sizes.

Moreover, in order to obtain maximal value of *MC modularity*, we propose a community detection algorithm based on *MC modularity*, which is referred to as the *DC_MC algorithm*, to extract groups of vertices that are densely interconnected, but only sparsely connected with the rest of the network; these groups will be identified as communities. A new method for shifting a group of nodes, instead of one node at a time, amongst communities also is designed to achieve optimal results.

Further, we have implemented the *DC_MC algorithm*, and experiments on synthetic and real data sets have shown that *MC modularity* can measure correctly the quality of community structures no matter communities are similar or different in size.

The details of this study are introduced as follows. Section 2 reviews the related work. In Section 3, we define *MC modularity* to measure the partition quality of networks. In Section 4, we present the *DC_MC algorithm* to automatically detect communities, and we propose some optimized methods to shift vertices. In order to verify our approach, we conducted extensive experiments on synthetic and real data sets. The experimental design and results analysis are given in Section 5. Finally, we conclude the paper in Section 6.

2 Related works

There are several algorithms that have been designed to analyse community structures in complex networks. Methods and principles of physics, artificial intelligence, graph theory, and even matrix factorization have been applied for this purpose [1][2]. Most of these algorithms therefore detect community structures via maximizing modularity to obtain optimised solutions, and differ in terms of the variations of the definitions of modularity designed and the methods used to achieve maximal values of modularity. This section presents a review on definitions of modularity and methods to obtain values of modularity.

2.1 Definitions of Modularity

Modularity of Newman and Girvan (*NG modularity* for short). The modularity of Newman and Girvan [9], was originally introduced to define a stopping criterion for the algorithm of Girvan and Newman for identifying communities within a network, and has rapidly become an essential element of many clustering methods. *NG modularity* is defined as $Q = \sum_i (e_{ii} - a_i^2)$, where e_{ii} is the fraction of edges in the network that link vertices within community i; and a_i is the fraction of edges that connect

to vertices in community i. *NG modularity* measures the fraction of the edges in the network that connect vertices of the same type (i.e., within community edges) minus the expected value of the same quantity in a network with the same community divisions but random connections between the vertices. In this way, the more the number of internal edges of the community exceeds the expected number, the better defined the community will be.

Motif modularity. Arenas et al. [10] suggested that high edge densities inside clusters usually imply the existence of long-range topological correlations between vertices, which are revealed by the presence of *motifs*, i.e., connected undirected subgraphs, such as cycles. They suggested that modularity can be generalized by comparing the density of *motifs* inside clusters with the expected density in the modularity's null model.

Influence-based modularity. Ghosh and Lerman [11] believed that edges do not give a true measure of network connectivity. Instead, they defined the number of paths, of any length, that exist between two vertices as the measure of network connectivity. They redefined modularity in terms of the influence metric and use the new definition of modularity, *influence-based modularity*, to partition a network into communities.

Modularity density. Li et al. [12] have introduced *modularity density*, which consists in the sum over the clusters of the ratio between the difference of the internal and external degrees of the cluster and the cluster size. The *modularity density* does not require a null model.

Network community profile plot. The *network community profile plot*, defined by Leskovec et al. [17], is defined as the conductance value (the ratio of the number of "cut" edges between a set and its complement divided by the number of "internal" edges inside that set) of the minimum conductance set of cardinality k in the network. Based on the concept of *network community profile plot*, Leskovec et al. [17] observed that good network communities exist only up to a size scale of ≈ 100 nodes.

2.2 Methods for obtaining maximal values of modularity

Once modularity has been derived, the community structures can be identified and detected through modularity analysis. Brandes et al. [18] have shown that the process of detecting clusters of vertices with a high modularity, and therefore identifying communities within a network, is an NP-complete problem. In order to complete this process within a reasonable and acceptable time, several different techniques can be used:

Greedy method. Greedy method was first devised by Newman [7] to identify communities with a high modularity. It is an agglomerative hierarchical clustering method, where groups of vertices are

successively joined to form larger communities such that the modularity increases after the merging. Danon et al. [15] found that the performance of Newman's Fast algorithm [7] is affected by inhomogeneities in community sizes considerably, so they modified the algorithm such that they can treat the communities of different sizes equally. Ciglan and Nørvåg [19] proposed a greedy algorithm for detecting size-constrained communities in large networks. The algorithm allows a user to specify the upper size limit of the communities being produced.

Duch method. Duch and Arenas [20] proposed to achieve maximal modularity via shifting the vertices between two initial groups with the same number of vertices. After the bipartition, each cluster is considered as a graph on its own and the procedure is repeated, as long as the modularity increases for the partitions found.

Simulated annealing method. Guimerà et al. [5] employed simulated annealing to obtain maximal value of modularity. Two types of moves are combined to achieve this: local moves, where a single vertex is shifted from one cluster to another, taken at random; and global moves, consisting of mergers and splits of communities.

In addition to these, spectral method [21], mathematical programming [22], and genetic algorithm [23] are also used to achieve maximal values of modularity for community detection.

3 Modularity based on the *coupling coefficient*

In this section, we introduce the concept of a *coupling coefficient* between two communities. This concept is proposed to reflect the edge density between two communities, and the edge density within these two communities, simultaneously. Based on the average *coupling coefficients* over all pairs of communities, a new modularity can be defined, which is referred to as *Modularity based on the coupling coefficient (MC modularity)* for analyzing and measuring the strengths of discovered community structures. We will firstly define the *link density within a community* and the *link density between two communities* before we give definitions of the *coupling coefficient* and *MC modularity*.

3.1 The definition of the modularity based on the *coupling coefficient*

Assume that a social network is denoted by graph $G=(V,E)$, where V is a set of vertices that represent underlying social entities, and E is a set of edges that represent interactions between pairs of entities. $|V|$ denotes the number of vertices in V. V_i is a subset of V, and $P(k)=\{V_1,V_2,...,V_k\}$ is a

community structure of the network, i.e. a division of V, where $\bigcup_{\forall V_i \in P(k)} = V$, and $\bigcap_{\forall V_i \in P(k)} = \Phi$ (an empty set). The lowercase letters x, y, z denote vertices in V_i and V_{i-x} represents $V_i - \{x\}$. E_i is a subset of E, and each edge in E_i connects two vertices of V_i. $e(V_i, V_j)$ denotes the number of edges in $G = (V, E)$ that link vertices in community V_i to vertices in community V_j.

Definition 1 *Link Density within a community.*

Link density $D(V_i)$ within community V_i is defined as: $D(V_i) = \begin{cases} \dfrac{2e(V_i, V_i)}{|V_i|(|V_i|-1)}, & |V_i| > 1 \\ 0, & |V_i| = 1 \ \ or \ \ |V_i| = 0 \end{cases}$.

$D(V_i) \in [0,1]$, $D(V_i)$ measures the density of edges linking vertices within community V_i. When the number of vertices in V_i is fixed, the more link edges exist between vertices, the larger the $D(V_i)$ is; when all of vertices in V_i are fully connected to each other, $D(V_i) = 1$; when there is only one vertex in V_i, or V_i is empty, $D(V_i) = 0$.

Definition 2 *Link Density between two communities.*

Link Density $L(V_i, V_j)$ between community V_i and V_j is defined as: $L(V_i, V_j) = \begin{cases} \dfrac{e(V_i, V_j)}{|V_i| \times |V_j|} & |V_i| \geq 1, \ \ |V_j| \geq 1 \\ 0 & |V_i| = |V_j| = 0 \end{cases}$.

$L(V_i, V_j) \in [0,1]$, $L(V_i, V_j)$ measures the density of edges linking vertices in community V_i to vertices in community V_j. When the numbers of vertices in community V_i and V_j are fixed, the greater $e(V_i, V_j)$ is, the larger the $L(V_i, V_j)$ is. When $|V_i|$ vertices in V_i are fully connected with $|V_j|$ vertices in $|V_j|$, $L(V_i, V_j) = 1$; when two communities are mutually independent from each other (there is no edge between V_i and V_j), or V_i and V_j are two empty sets, $L(V_i, V_j) = 0$.

Definition 3 *Coupling Coefficient between two communities.*

Coupling Coefficient $C(V_i, V_j)$ between community V_i and V_j is defined as: $C(V_i, V_j) = \dfrac{L(V_i, V_j)}{D(V_i) + D(V_j) + 1}$.

$C(V_i, V_j)$ measures the proportion of link density between community V_i and V_j. The lower the link density between community V_i and V_j is, and the higher the link density between communities

is; the smaller $C(V_i,V_j)$ is and stronger the feature (vertices are densely connected within a community and have much sparser connections between the communities) will be. When there is no edge between V_i and V_j, V_i and V_j are independent from each other, and $C(V_i,V_j)=0$.

Definition 4 *Modularity based on coupling coefficient– MC modularity*. Let $P(k)=\{V_1,V_2,...,V_k\}$ is a community structure with k communities in $G=(V,E)$, then the *MC modularity* with respect to $P(k)=\{V_1,V_2,...,V_k\}$ is defined as:

$$MC(P(k))=\begin{cases}1-\dfrac{2}{k\times(k-1)}\sum_{i=1}^{k}\sum_{j>i}C(V_i,V_j), & k>1\\ D(V), & k=1\end{cases}.$$

Where $\dfrac{2}{k\times(k-1)}\sum_{i=1}^{k}\sum_{j>i}C(V_i,V_j)$ is the average coupling coefficient over all pairs of communities. The lower the coupling coefficient is, the larger $MC(P(k))$ is. The *MC modularity* of $P(1)$, a special community structure in which all vertices of V form a community, is defined as $D(V)$.

Example 1. In Figure 1(a), $P(2)=\{V_1,V_2\}$, $D(V_1)=D(V_2)=1$, $L(V_1,V_2)=0.0025$, $C(V_1,V_2)=0.00083$, $MC(P(2))=0.9992$. In Figure 1(b), $P'(2)=\{V'_1,V'_2\}$, $D(V'_1)=1$, $D(V'_2)=0.7$, $L(V'_1,V'_2)=0.2$, $C(V'_1,V'_2)=0.074$, $MC(P'(2))=0.926$. We can see that $MC(P(2))>MC(P'(2))$, which correctly reflects the strength of two community structures.

3.2 The properties of MC modularity

MC modularity has following properties:

(1) $MC(P(|V|))=1-MC(P(1))$

$P(|V|)$ is a special community structure in which each vertex in $V(|V|>1)$ forms an independent community, so $MC(P(|V|))=1-\dfrac{2e(V,V)}{|V|(|V|-1)}=1-MC(P(1))$. If $G=(V,E)$ is a complete graph, then $MC(P(1))=1$, $MC(P(|V|))=0$. If $|E|=0$ in $G=(V,E)$, then $MC(P(1))=0$, $MC(P(|V|))=1$.

(2) *MC modularity* has higher values compare to those of *NG modularity* when communities are similar in size.

We take *MC modularity* and *NG modularity* with respect to $P(2)=\{V_1,V_2\}$ as a representative case.

V_1 and V_2 are similar in size $\Rightarrow |V_1| \approx |V_2| \Rightarrow$. $C(V_1,V_2) \approx \frac{e(V_1,V_2)}{2e(V_1,V_1)+2e(V_2,V_2)+|V_1|(|V_1|-1)} \Rightarrow$

$MC(P(2)) = 1 - \frac{e(V_1,V_2)}{2e(V_1,V_1)+2e(V_2,V_2)+|V_1|(|V_1|-1)} > 1 - \frac{e(V_1,V_2)}{e(V_1,V_1)+e(V_2,V_2)}$.

According to the Definition of *NG modularity* [9], $NG(P(2)) = 1 - \frac{2e(V_1,V_2)}{e(V,V)} - (\frac{e(V_1,V_1)}{e(V,V)})^2 - (\frac{e(V_2,V_2)}{e(V,V)})^2$.

In general, $e(V_1,V_1)+e(V_2,V_2) > e(V_1,V_2)$, so $\frac{e(V_1,V_2)}{e(V_1,V_1)+e(V_2,V_2)} < \frac{2e(V_1,V_2)}{e(V,V)}$, and $MC(P(2)) > NG(P(2))$.

(3) *MC modularity* would not be affected by $\frac{e(V_i,V_i)}{e(V,V)}$ when communities are different in size.

We also take *MC modularity* and *NG modularity* with respect to $P(2) = \{V_1,V_2\}$ as a representative case. We assume that $e(V_1,V_1) >> e(V_2,V_2)$, then $\frac{e(V_1,V_1)}{e(V,V)}$ would approach to 1, so $NG(P(2))$ would approach to 0-the minimal value of *NG modularity*, no matter how much the $\frac{e(V_1,V_2)}{e(V,V)}$ is, thus in this case, *NG modularity* does not provide an adequate measure for measuring the quality of community structures. However, the link density ($D(V_1)$, $D(V_2)$ or $L(V_1,V_2)$) would not be affected by $\frac{e(V_1,V_1)}{e(V,V)}$, because they are the proportion of the actual number of edges over the maximal number of edges. Therefore, $MC(P(2))$ would not be affected by $\frac{e(V_1,V_1)}{e(V,V)}$ when communities are different in size.

Considering a threshold of $MC(P(2))$ be ε, then $MC(P(2)) > \varepsilon$, iff. $e(V_1,V_2) < \frac{2(1-\varepsilon)[|V_2|(|V_2|-1)e(V_1,V_1)+|V_1|(|V_1|-1)e(V_2,V_2)]}{(|V_1|-1)(|V_2|-1)} + (1-\varepsilon)|V_1||V_2|$. It means that the fewer the number of edges existing between two communities, the greater $MC(P(2))$ would be. For example, in the Example 1, $P(2) = \{V_1,V_2\}$, if $e(V_1,V_2) \le 12$, $MC(P(2)) \ge 0.99$.

The analysing of $MC(P(2))$ is straightforward for the case of $MC(P(k))$ ($k > 2$), which will not be presented here.

4 Community detection algorithm

In $G=(V,E)$, a fixed k (community number) corresponds to many kinds of $P(k)$ (community structure). $MC(P(k))$ varies with k and $P(k)$. $P(k^*)=\arg\max_{k,P(k)} MC(P(k))$, the community structure with maximal value of *MC modularity*, represents a rational community structure of G.

Figure 2 shows 5 values of *MC modularity* of a network with 11 vertices and 18 edges. From Figure 2, we can see that $P(3)$ shown in Figure 2(b) is a rational community structure and $MC(P(3))$ is larger than the others.

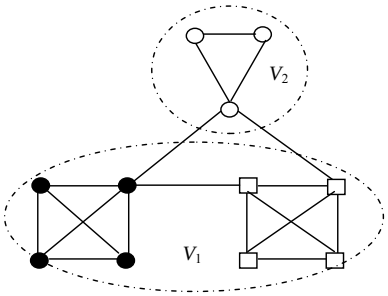


Figure 2(a). 2 communities $MC(P(2))=0.9662$

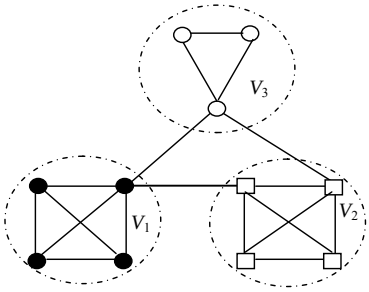


Figure 2(b). 3 communities $MC(P(3))=0.9746$

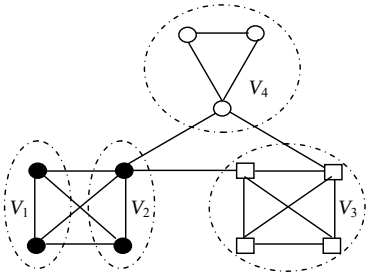


Figure 2(c). 4 communities $MC(P(4))=0.9236$

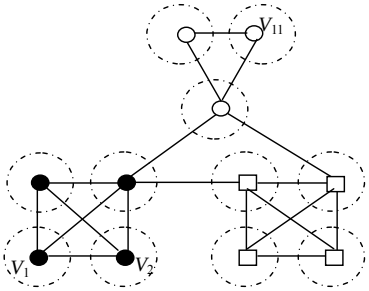


Figure 2(d). 11 communities $MC(P(11))=0.6727$

$$MC(P(1))=0.3273<MC(P(11))<MC(P(4))<MC(P(2))<MC(P(3))$$

Figure 2. The values of *MC modularity* under different number of communities

In order to detect communities in a network, we develop an algorithm that applies modularity (such as *MC modularity*, *NG modularity*) to evaluate the strength of discovered community structures. This algorithm is referred to as the *DC_MC algorithm*. The main idea of the *DC_MC algorithm* is initializing a community structure $P(k)=\{V_1,...,V_k\}$ according to the degrees of vertices in a network, and then optimizing $P(k)$ to maximize $MC(P(k))$ by shifting vertices amongst V_i ($i=1,...,k$); and further repeat this process with increased k until $MC(P(k))$ cannot be increased any more. At that stage, this process has identified k community structures; each of them corresponds to a maximal value of *MC modularity* and a special community structure. Then a rational community structure can be selected from these k community structures.

4.1 Description of the *DC_MC algorithm*

The *DC_MC algorithm* for detecting communities is as follows:

Input: a graph $G=(V,E)$, and k (the maximal community number)

Output: $P(k^*)=\{V_1,V_2,...,V_{k^*}\}$, i.e. a community structure of G

(1) $P(1)=\{V\}$, $MC(P(1))=D(V)=\dfrac{2|E|}{|V|(|V|-1)}$;

(2) $i=2$;

(3) while $i\le k$ do

(4) $V'=V$, $P(i)=\Phi$;

(5) for $j=1$ to i do

(6) $V_j=\Phi$;

(7) end for

(8) $j=0$;

(9) while $j<i$ do

(10) $j=j+1$;

(11) if $V'\neq\Phi$ do

(12) $x=\arg\max\limits_{y\in V'} e(\{y\},V')$, $V_j=V_j+\{x\}$, $V'=V'-\{x\}$;

(13) for each $y\in V'$ do

(14) if $e(\{y\},V_j)\neq 0$ do

(15) $V_j=V_j+\{y\}$, $V'=V'-\{y\}$;

(16) end if

(17) end for

(18) else if $V'=\Phi$ do

(19) $(x,l)=\arg\min\limits_{\substack{r=1,2,...,j-1\\ y\in V_r}} e(\{y\},V_r)$, $V_j=V_j+\{x\}$, $V_l=V_l-\{x\}$;

(20) end if

(21) end while

(22) while $V'\neq\Phi$ do

(23) $(x,l)=\arg\max\limits_{\substack{r=1,2,...,j\\ y\in V'}} e(\{y\},V_r)$, $V_l=V_l+\{x\}$, $V'=V'-\{x\}$;

(24) end while

(25) for $j=1$ to i do

(26) $P(i)=P(i)+V_j$;

(27) end for

(28) compute $MC(P(i))$, *keepflag*=0;

(29) while *keepflag* < |*V*| do

(30) for $j=1$ to i do

(31) for each $x \in V_j$

(32) for $l=1$ to i, $l \neq j$ do

(33) $P'(i)=\{V_1,...,V_j-\{x\},...,V_l+\{x\},...,V_i\}$;

(34) if $MC(P'(i))>MC(P(i))$ do

(35) $P(i)=P'(i)$, $MC(P(i))=MC(P'(i))$, *keepflag*=0;

(36) else *keepflag*= *keepflag*+1;

(37) end if

(38) end for

(39) end for

(40) end for

(41) end while

(42) $i=i+1$;

(43) end while

(44) $k^*=\arg\max_{j=1,...,k} MC(P(j))$

(45) output $P(k^*)$

In the *DC_MC algorithm*, (1) sets a community structure in which all vertices form a community; (4)~(28) initialize a community structure $P(i)$ according to the degrees of vertices, and their computational complexity is $O(|V|)$; (29)~(41) search the optimal community structure of G with respect to i communities by shifting vertex x. The computational complexity for calculating the value of *MC modularity* in $P(i)$ is $O(i^2)$; there are $|V|$ vertices shifting amongst $i-1$ communities, so the computational complexity (29)~(41) is $O(|V|i^3)$; the loop of (3)~(43) searches k optimal community

structures; (44) selects a community structure from $P(i)$ ($i = 1,...,k$). The computational complexity of the algorithm is $O(|V|^2 + k|V| + |V|(2^3 + 3^3 + \cdots + k^3)) = O(|V|^2 + k|V| + k^4|V|)$.

4.2 Calculation of modularity after moving a vertex

A crucial issue in the *DC_MC algorithm* described in Section 4.1 is the calculation of modularity $MC(P'(i))$ after moving vertex x in (34). In fact, when moving x from V_r to V_j, not only the coupling coefficient $C(V_r, V_j)$ between community V_r to V_j changes, but also coupling coefficients $C(V_r, V_s), s \neq r, j$ and $C(V_j, V_l), l \neq r, j$ will be changed, where at least there is an edge between V_s and V_r, between V_l and V_j. Figure 3 shows an example of links amongst community V_r, V_j, V_s, V_u, V_v and V_w, where vertex x belongs to V_r; in V_j and V_s, there is at least a vertex connecting to x; in V_u there is at least a vertex connecting to a vertex except for x in V_r; there is at least an edge between V_j and V_s, between V_j and V_v, between V_s and V_w, between V_w and V_v.

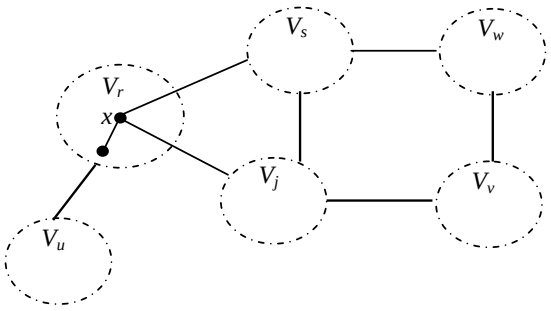


Figure 3. An example of links between communities

Now moving x from V_r to V_j, then $P(i) = \{V_1,...,V_r,...,V_j,...,V_i\}$ will change to $P'(i) = \{V_1,...,V_r - \{x\},...,V_j + \{x\},...,V_i\}$. Therefore, $|V_r|' = |V_r| - 1$, $|V_j|' = |V_j| + 1$, $e'(V_r, V_r) = e(V_r, V_r) - e(\{x\}, V_{r-x})$, $e'(V_j, V_j) = e(V_j, V_j) + e(\{x\}, V_j)$, $e'(V_r, V_j) = e(V_r, V_j) + e(\{x\}, V_{r-x}) - e(\{x\}, V_j)$, $e'(V_r, V_s) = e(V_r, V_s) - e(\{x\}, V_s)$, $e'(V_j, V_s) = e(V_j, V_s) + e(\{x\}, V_s)$, and we have:

$$D'(V_r) = \begin{cases} \dfrac{2[e(V_r, V_r) - e(\{x\}, V_{r-x})]}{(|V_r| - 1)(|V_r| - 2)} \\ = \dfrac{|V_r|}{|V_r| - 2} D(V_r) - \dfrac{2e(\{x\}, V_{r-x})}{(|V_r| - 1)(|V_r| - 2)}, |V_r| > 2 \\ 0 \qquad\qquad\qquad\qquad , |V_r| \leq 2 \end{cases} \tag{5}$$

$$D'(V_j)=\frac{2[e(V_j,V_j)+e(\{x\},V_j)]}{(|V_j|+1)|V_j|}=\frac{|V_j|-1}{|V_j|+1}D(V_j)+\frac{2e(\{x\},V_j)}{(|V_j|+1)|V_j|} \quad (6)$$

$$L'(V_r,V_j)=\frac{e(V_r,V_j)+e(\{x\},V_{r-x})-e(\{x\},V_j)}{(|V_r|-1)(|V_j|+1)}=\frac{|V_r||V_j|L(V_r,V_j)}{(|V_r|-1)(|V_j|+1)}+\frac{e(\{x\},V_{r-x})-e(\{x\},V_j)}{(|V_r|-1)(|V_j|+1)} \quad (7)$$

$$L'(V_r,V_s)=\frac{e(V_r,V_s)-e(\{x\},V_s)}{(|V_r|-1)|V_s|}=\frac{|V_r|}{|V_r|-1}L(V_r,V_s)-\frac{e(\{x\},V_s)}{(|V_r|-1)|V_s|} \quad (8)$$

$$L'(V_r,V_u)=\frac{e(V_r,V_u)}{(|V_r|-1)|V_u|}=\frac{|V_r|}{|V_r|-1}L(V_r,V_u) \quad (9)$$

$$L'(V_j,V_s)=\frac{e(V_j,V_s)+e(\{x\},V_s)}{(|V_j|+1)|V_s|}=\frac{|V_j|}{|V_j|+1}L(V_j,V_s)+\frac{e(\{x\},V_s)}{(|V_j|+1)|V_s|} \quad (10)$$

$$L'(V_j,V_v)=\frac{e(V_j,V_v)}{(|V_j|+1)|V_v|}=\frac{|V_j|}{|V_j|+1}L(V_j,V_v) \quad (11)$$

For communities without a vertex connecting to any vertex in V_r or V_j, the link densities within and between communities will not be affected by moving x. For example, in Figure 3, $D(V_w)$, $D(V_v)$ and $L(V_v,V_w)$ will not be changed after moving x. Therefore, after computing $C'(V_r,V_j)$, $C'(V_r,V_s)$, $C'(V_r,V_u)$, $C'(V_j,V_s)$ and $C'(V_j,V_v)$, the modularity $MC(P'(i))$ can be obtained.

4.3 How to choose a candidate vertex

Another crucial issue in the *DC_MC algorithm* (described in Section 4.1) is how to choose a candidate vertex *x*. A social network usually includes many vertices, if every vertex is going to be moved, the computational complexity will be high. In fact, if the movement of a vertex will not lead to an increase of *MC modularity*, then this movement is in vain. According to the definition of *MC modularity*, we know that reducing coupling coefficient means increment of *MC modularity* and $C(V_r,V_j)$ in Formulas (5)–(11) has most change, so we will focus on the change of $C(V_r,V_j)$. $C(V_r,V_j)$ is going to reduce after moving x from V_r to V_j if $D'(V_r)$ and $D'(V_j)$ increase, and $L'(V_r,V_j)$ reduces, i.e., $D'(V_r)-D(V_r)\geq 0$, $D'(V_j)-D(V_j)\geq 0$, $L'(V_r,V_j)-L(V_r,V_j)\leq 0$, therefore, we have:

$$e(\{x\},V_{r-x})\leq\frac{2}{|V_r|-2}e(V_{r-x},V_{r-x}) \quad (12)$$

$$e(\{x\},V_j)\geq\frac{2}{|V_j|-1}e(V_j,V_j) \quad (13)$$

$$|V_r||V_j|e(\{x\},V_{r-x})-(|V_r|-1)(|V_j|+1)e(\{x\},V_j)\leq(|V_r|-|V_j|-1)e(V_{r-x},V_j) \quad (14)$$

Formulas (12)–(14) describe the relationships of edges between candidate x and V_j, V_{r-x}. The vertices that satisfy Formulas (12)–(14) would be the most suitable candidates that have edges with vertices in V_r and V_j, and these vertices are referred to as *boundary vertices*. Conversely, the vertices that have only edges with vertices in a community are referred to as *internal vertices*. These internal vertices are not suitable for moving because they do not satisfy Formula (14) and their movement would result in incensement of $C(V_r, V_j)$.

Example 2. A social network shown in Figure 4(a) includes 9 social individuals $v_1 \sim v_9$ that form two communities. In the initial community structure $P(2) = \{\{v_1, v_2, v_3, v_4, v_6, v_7\}, \{v_5, v_8, v_9\}\}$, shown in Figure 4(b), $MC(P(2)) = 0.8397$, and the values of *MC modularity* after shifting boundary vertices v_5 and v_7 are respectively 0.9412 and 0.9781. The values of *MC modularity* have increased. According to Figure 4, these shifts are logical.

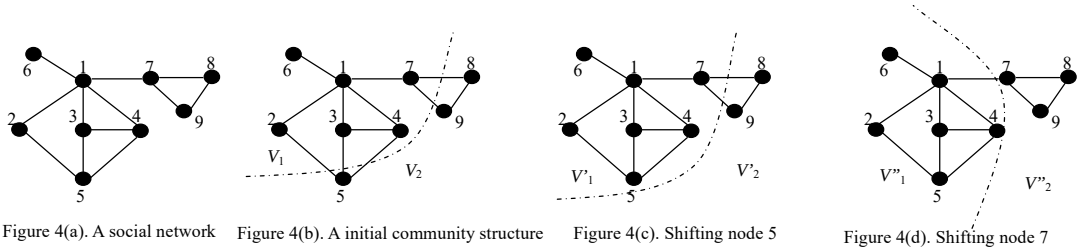


Figure 4(a). A social network

Figure 4(b). A initial community structure $MC(P(2)) = 0.8397$

Figure 4(c). Shifting node 5 $MC(P'(2)) = 0.9412$

Figure 4(d). Shifting node 7 $MC(P''(2)) = 0.9781$

Figure 4. The figures of Example 2

4.4 How to choose a candidate vertices group

The focus of Section 4.3 concerns an alternative vertex so that communities can be recognised in a more logical and rational way. However, sometimes considering an alternative vertex will not result in an improved community structure. For example, in Figure 5, the vertices group V_x composed of vertex x and the vertices connecting to it should be in community V_j, but during initial partition, V_x is assigned to V_i, So moving any vertex in V_x separately will not reduce $C(V_i, V_j)$, but if V_x is moved as a whole, the $C(V_i, V_j)$ will be reduced significantly, therefore moving a group of vertices should be considered. The conditions that the candidate vertex group should satisfy are given in Formulas (15)–(17), where $V_{i-V_x} = V_i - V_x$.

$$e(V_x, V_{i-V_x}) \le \frac{2|V_i||V_x| - |V_x|^2 - |V_x|}{(|V_i| - |V_x|)(|V_i| - |V_x| - 1)} e(V_{i-V_x}, V_{i-V_x}) \tag{15}$$

$$e(V_x, V_j) \ge \frac{2|V_j||V_x| + |V_x|^2 - |V_x|}{|V_j|(|V_j| - 1)} e(V_j, V_j) \tag{16}$$

$$|V_i||V_j| e(V_x, V_{i-v_x}) - (|V_i| - |V_x|)(|V_j| + |V_x|) \times e(V_x, V_j) \le (|V_i||V_x| - |V_j||V_x| - |V_x|^2) e(V_{i-V_x}, V_j) \tag{17}$$

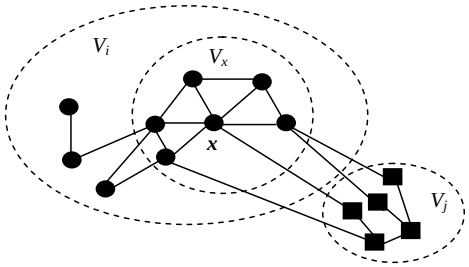


Figure 5. An example of vertices group

5. Experimental studies and results

The purposes of our experiments are to test (1) whether the *DC_MC algorithm* can correctly identify the community numbers in various types of social networks; (2) whether our *MC modularity* can correctly evaluate the strength of the discovered community structures; (3) whether our approach of choosing candidate vertices is effective. For the first and second purposes, we have implemented the approaches proposed in this study and applied them to detect community structures in four synthetic networks and three real networks [24][25][26]. In the synthetic networks, communities have very different sizes and topologies. The three real networks are well-known social networks used as a benchmark for testing community detection algorithms. We used *MC modularity* and *NG modularity* (the most popular approach [7]) respectively in the *DC_MC algorithm* and compared the community structures under these two different modularities. For the third purpose, we created a purpose-built synthetic network to verify the features of *MC modularity* and ran the *DC_MC algorithm* under two different strategies: the first chooses every vertex as a candidate vertex to shift, and the other only chooses vertices that satisfy Formulas (12)–(17) as candidates to shift; we then compared the running time of the *DC_MC algorithm* under these two different strategies. If the running time of the *DC_MC algorithm* under the second strategy is shorter than the one under the first strategy, then it shows that our approach of choosing candidate vertices is effective.

5.1 Identifying the number of communities

Four synthetic networks and three real networks were used to test whether the *DC_MC algorithm* can correctly identify the number of communities in social networks with different sizes and topologies. These networks are illustrated in Figure 6(a)~(g), *synthetic network_1* and *synthetic network_2* (shown in Figure 6(a) and Figure 6(b)) consist of one big clique and four small cliques. The big clique consists of 100 vertices and each small clique consists of 4 vertices, but these four small cliques are connected to different vertices of the big clique in Figure 6(a) while the four small cliques are connected to a same vertex of the big clique in Figure 6(b). In *synthetic network_3* and *synthetic network_4* (shown in Figure 6(c) and Figure 6(d)), small communities are set as cliques with 4 vertices and the big community consists of 100 vertices, but edges were placed independently at random between vertex pairs with probability $p_{in} = 0.9$ for an edge to fall between vertices in the big community and $p_{out} = 0.1$ to fall between vertices in different communities. The first real network, in Figure 6(e), is *Zachary's network* of karate club members [24], a well-known network used as a benchmark to test community detection algorithms. The *Zachary's network* consists of 34 vertices, concerning the members of a karate club in the United States, and presenting data collected during a period of three years. Edges connect individuals who were observed to interact outside the activities of the club. At some point, a conflict between the club president and the instructor led to the fission of the club into two separate groups, supporting the instructor and the president, respectively (indicated by squares and circles). The two groups (squares and circles in Figure 6(e)), one around vertices 33 and 34 (34 is the president), the other around vertex 1 (the instructor), can be easily distinguished in Figure 6(e). The second real network, in Figure 6(f), is *Lusseau's network* of bottlenose dolphins living in Doubtful Sound in New Zealand [25]. It is another graph often used to test algorithms for community detection. There are 62 dolphins and edges were set between animals that were seen together more often than expected by chance. The dolphins separated into two groups after a dolphin (vertex 31) left the place for some time (squares and circles in Figure 6(f)). Such groups are quite cohesive, and easily identifiable from the original network structure. The third real network, in Figure 6(g), is the network of interactions between major characters in the novel Les Misèrables by Victor Hugo [26]. In this network, 77 vertices represent characters and an edge between two vertices represents coappearance of the corresponding characters in one or more scenes. In this network, the interactions between major characters are complex than that in the *Zachary's karate network* and *Lusseau's dolphins network.*

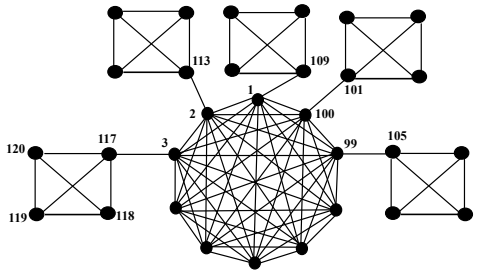


Figure 6(a). *synthetic network_1*

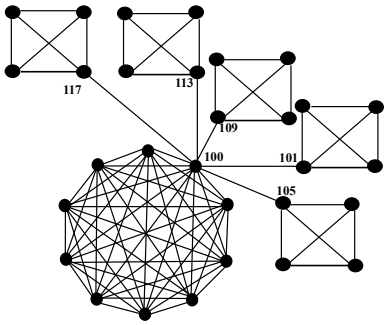


Figure 6(b). *synthetic network_2*

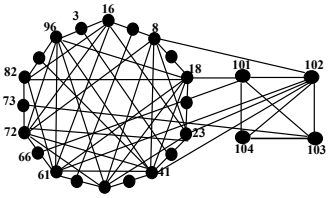


Figure 6(c). *synthetic network_3*

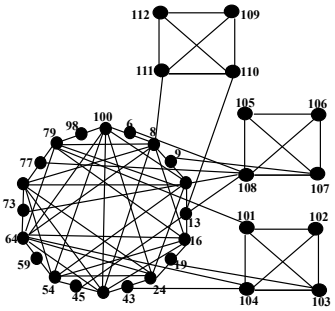


Figure 6(d). *synthetic network_4*

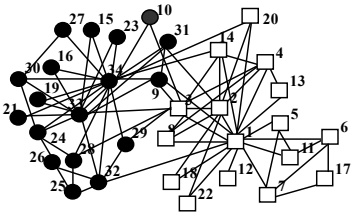


Figure 6(e). *Zachary's karate network*

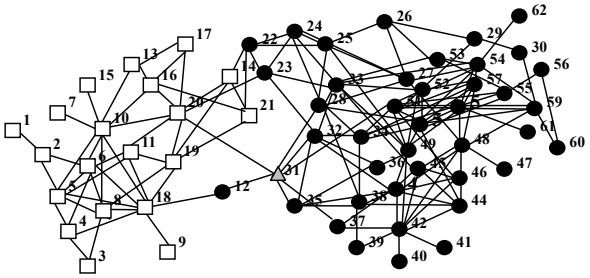


Figure 6(f). *Lusseau's dolphins network*

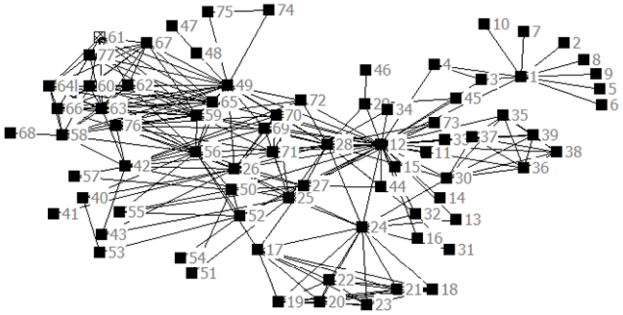


Figure 6(g). *Les Misèrables' characters network*

Figure 6. synthetic networks and real networks

The real numbers of communities in these networks of Figure 6, as we know, are less than 15, so in the algorithm we assign the maximal number k of communities to be 15 – it is assumed that the number of communities is unknown but is expected to be less than 15. In the experiments, the number of communities will be increased progressively from 1 to 15, so the algorithm will compute 15 values of modularity. We want to see if the numbers of communities in the community structure with maximal value of modularity would match with the real numbers of communities.

In the experiments, the *DC_MC algorithm* produced 15 maximal values of *MC modularity* and 15 maximal values of *NG modularity* respectively for each network in Figure 6. Table 1 shows the maximal values of *MC modularity* and *NG modularity* of seven networks of Figure 6 under different community numbers. The variation trends of maximal values of *MC modularity* and *NG modularity* of six networks are shown in Figure 7(a)~(g).

Table 1. The maximal values of *MC modularity* and *NG modularity*

Data sets		The number of communities														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
synthetic network_1	*MC*	0.6982	0.9992	0.9995	0.9996	0.9996	0.9997	0.9997	0.9782	0.9508	0.9495	0.9366	0.9268	0.9264	0.9177	0.9199
	NG	0	0.0026	0.0049	0.0074	0.0097	0.0122	0.0122	0.0112	0.0104	0.0101	0.0098	0.0095	0.0091	0.0088	0.0085
synthetic network_2	*MC*	0.6891	0.9969	0.9976	0.9988	0.9989	0.9997	0.9992	0.9868	0.9807	0.9775	0.9761	0.9688	0.9660	0.9614	0.9537
	NG	0	0.0055	0.0036	0.0067	0.0072	0.0130	0.0121	0.0120	0.0124	0.0116	0.0104	0.0095	0.0092	0.0088	0.0085
synthetic network_3	*MC*	0.6701	0.7565	0.9140	0.8865	0.8951	0.8922	0.8755	0.8638	0.8673	0.8594	0.8541	0.8527	0.8468	0.8416	0.8495
	NG	0	0	0.0376	0.0251	0.0155	0.0094	0.0159	0	0	0.0171	0.0127	0.0103	0.0160	0.0164	0
synthetic network_4	*MC*	0.6411	0.8593	0.9086	0.9532	0.9729	0.9525	0.9105	0.9195	0.8736	0.8998	0.8948	0.8545	0.8911	0.8790	0.8566
	NG	0	0.0186	0.0194	0.0226	0.0332	0.0209	0.0222	0.0235	0.0244	0.0228	0.0249	0.0238	0.0241	0.0237	0.0228
Zachary's karate network	*MC*	0.1381	0.9770	0.9743	0.9680	0.9389	0.9617	0.9481	0.9523	0.9603	0.9622	0.9651	0.9625	0.9717	0.9631	0.9595
	NG	0	0.3564	0.4006	0.3368	0.3618	0.3284	0.3057	0.3755	0.3786	0.3217	0.2804	0.2661	0.3919	0.2406	0.2219
Lusseau's dolphins network	*MC*	0.0843	0.9958	0.9814	0.9779	0.9791	0.9779	0.9828	0.9846	0.9868	0.9856	0.9872	0.9831	0.9851	0.9783	0.9723
	NG	0	0.4010	0.5147	0.4760	0.4861	0.4380	0.4383	0.4228	0.4177	0.4035	0.3730	0.3574	0.3529	0.3678	0.3291
Les Misèrables' characters network	*MC*	0.0868	0.9826	0.9826	0.9863	0.9900	0.9623	0.9702	0.9599	0.9727	0.9584	0.9565	0.9580	0.9645	0.9654	0.9543
	NG	0	0.3025	0.3276	0.3141	0.4595	0.3111	0.3983	0.3783	0.4040	0.2986	0.4107	0.4073	0.4064	0.3889	0.3985

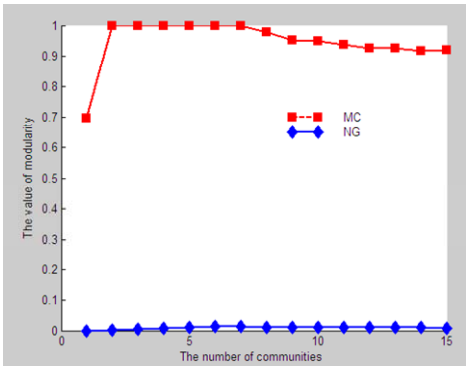


Figure 7(a). The values of *MC modularity* and *NG modularity* of *synthetic network_1*

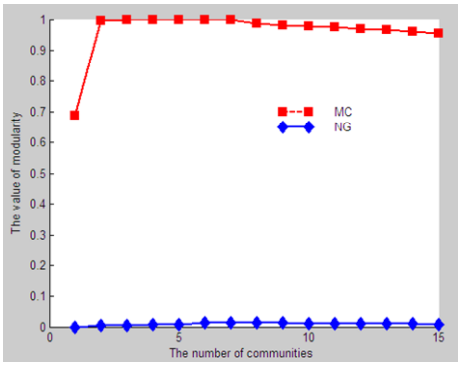


Figure 7(b). The values of *MC modularity* and *NG modularity* of *synthetic network_2*

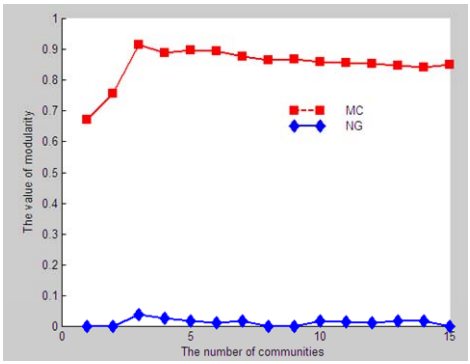


Figure 7(c). The values of *MC modularity* and *NG modularity* of *synthetic network_3*

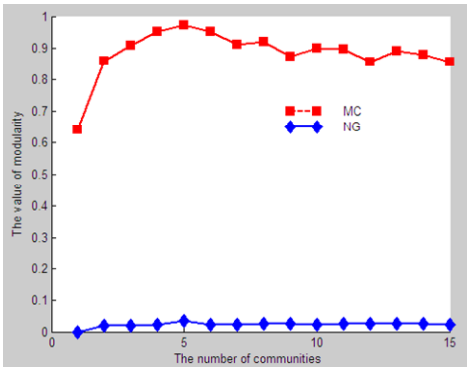


Figure 7(d). The values of *MC modularity* and *NG modularity* of *synthetic network_4*

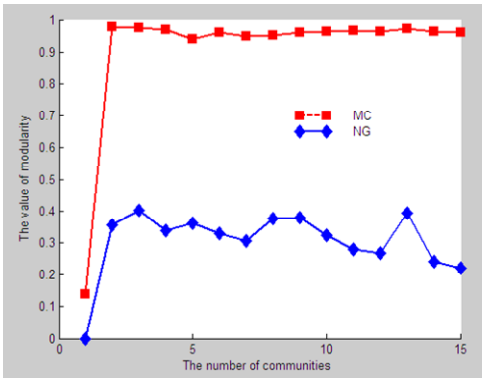


Figure 7(e). The values of *MC modularity* and *NG modularity* of *Zachary's karate network*

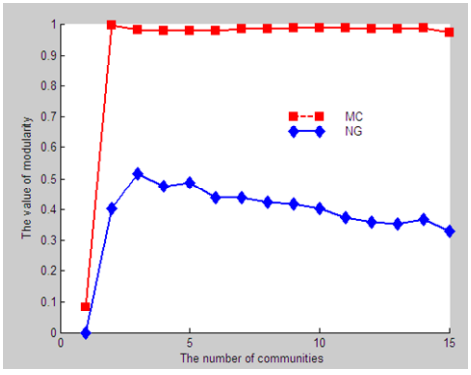


Figure 7(f). The values of *MC modularity* and *NG modularity* of *Lusseau's dolphins network*

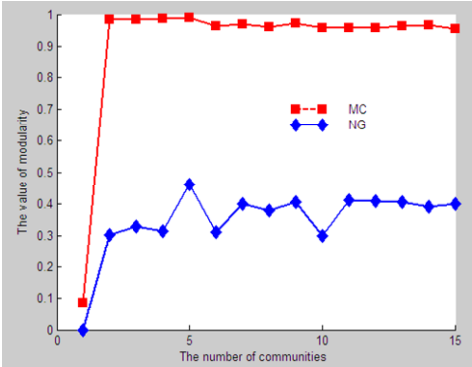


Figure 7(g). The values of *MC modularity* and *NG modularity* of *Les Misèrables' characters network*

Figure 7. The maximal values of *MC modularity* and *NG modularity* under different number of communities

In Figures 7(a) and (b) for *synthetic network_1* and *synthetic network_2*, the maximal values of *MC* modularity and *NG* modularity occur when the number of communities is 6. In Figure 7(c) for *synthetic network_3*, the maximal value of *MC* modularity and the maximal value of *NG* modularity occur when the number of communities is 3. In Figure 7(d) for *synthetic network_4,* the maximal value of *MC* modularity and the maximal value of *NG* modularity occur when the number of communities is 5. In Figure 7(e) for *Zachary's karate network* and in Figure 7(f) for *Lusseau's dolphins network*, both numbers of communities corresponding to the maximal value of *MC* modularity are 2, but the numbers of communities corresponding to the maximal value of *NG* modularity are 3. In Figure 7(g) for *Les Misèrables' characters network*, the maximal values of *MC* modularity and *NG* modularity occur when the number of communities is 5.

The results of Table 1 indicate that *DC_MC algorithm* is able to correctly identify the number of communities when *MC* modularity is used, no matter in the synthetic networks or in the real networks. Moreover, we can see that the maximal value in 15 values of *MC modularity* is significantly differ from the rest in Figures 7(c)~(d) for *synthetic network_3* and *synthetic network_4,* in which the edges in the big community and edges between communities were placed independently at random.

5.2 Evaluating the quality of discovered community structures

When *MC modularity* and *NG modularity* are used respectively by our algorithm to detect community structures, the community structures detected by *DC_MC algorithm* in networks of Figure 6 are shown in Figure 8, in which different colours represent different communities, and vertices with a same colour belong to the same community.

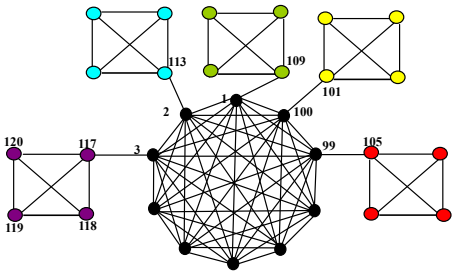


Figure 8(a). The community structure of *synthetic network_1* corresponding to the maximal value of *MC modularity*

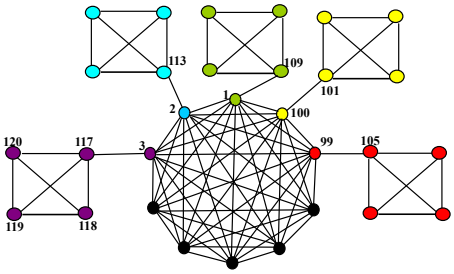


Figure 8(b). Community structure of *synthetic network*_1 corresponding to the maximal value of *NG modularity*

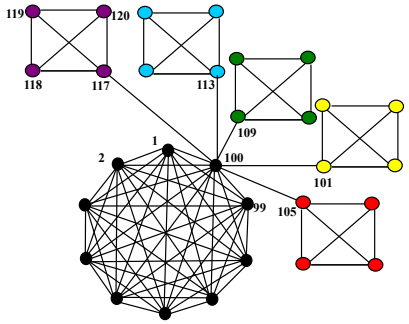


Figure 8(c). Community structure of *synthetic network*_2 corresponding to the maximal value of *MC modularity*

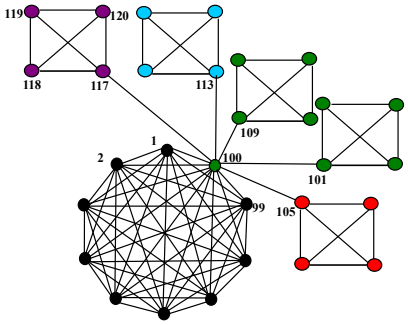


Figure 8(d). Community structure of *synthetic network* _2 corresponding to the maximal value of *NG modularity*

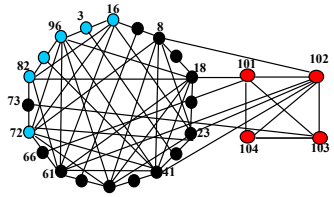


Figure 8(e). Community structure of *synthetic network*_3 corresponding to the maximal value of *MC modularity*

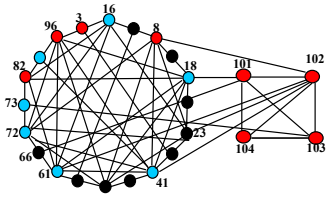


Figure 8(f). Community structure of *synthetic network*_3 corresponding to the maximal value of *NG modularity*

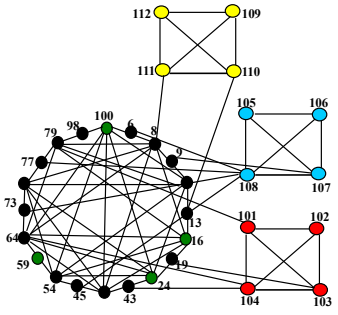


Figure 8(g). Community structure of *synthetic network*_4 corresponding to the maximal value of *MC modularity*

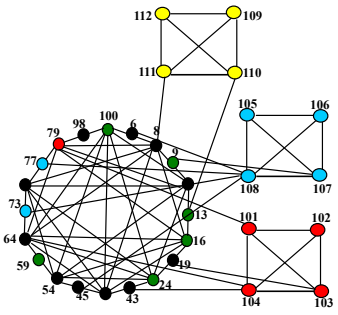


Figure 8(h). Community structure of *synthetic network*_4 corresponding to the maximal value of *NG modularity*

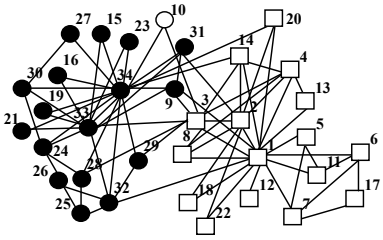


Figure 8(i). The community structure of *Zachary's karate network* corresponding to the maximal value of *MC modularity*

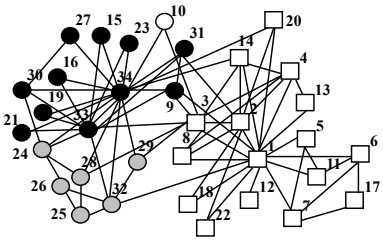


Figure 8(j). The community structure of *Zachary's karate network* corresponding to the maximal value of *NG modularity*

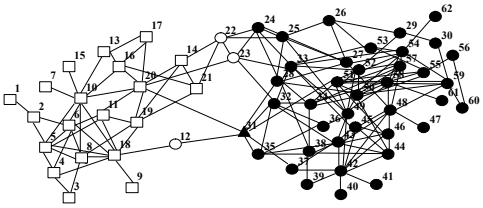


Figure 8(k). The community structure of *Lusseau's dolphins network* corresponding to the maximal value of *MC modularity*

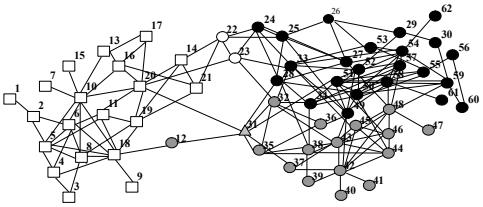


Figure 8(l). The community structure of *Lusseau's dolphins network* corresponding to the maximal value of *NG modularity*

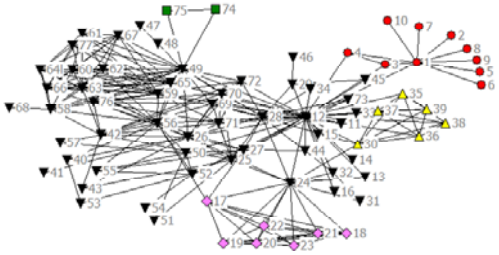


Figure 8(m). The community structure of *Les Misèrables' characters network* corresponding to the maximal value of *MC modularity*

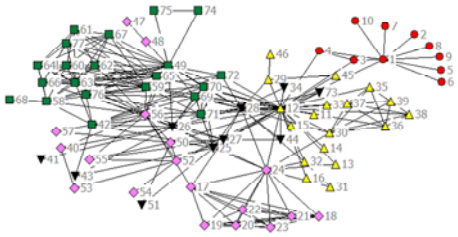


Figure 8(n). The community structure of *Les Misèrables' characters network* corresponding to the maximal value of *NG modularity*

Figure 8. Community structures of four *synthetic network*s and three real networks corresponding to the maximal value of *MC modularity* and *NG modularity*

Figure 8(a) and Figure 8(b) show the community structure of *synthetic network_1* detected by *DC_MC algorithm*. In Figure 8(a) (under *MC modularity*) all vertices are identified correctly, but in Figure 8(b) (under *NG modularity*) there are five vertices (vertices 1, 2, 3, 99, 100) which are misidentified.

Figure 8(c) and Figure 8(d) show the community structure of *synthetic network_2* detected by *DC_MC algorithm*. In Figure 8(c) (under *MC modularity)* all vertices are identified correctly, but in Figure 8(b) (under *NG modularity*) vertices 100, 101~104, 109~112 are considered as in one cluster.

Figure 8(e) and Figure 8(f) show the community structure of *synthetic network_3* detected by *DC_MC algorithm*. In Figure 8(e) (under *MC modularity*), the small community composed by vertices 101, 102, 103, 104 is identified correctly, vertices 1~100 are divided into two communities, one of them consists of vertices 3, 14, 16, 24, 34, 35, 72, 77, 82, 96. In Figure 8(f) (under *NG modularity*), 104 vertices are divided into three communities, but the clique composed by vertices 101, 102, 103, 104 is not identified.

Figure 8(g) and Figure 8(h) show the community structure of *synthetic network_4* detected by *DC_MC algorithm*. In Figure 8(g) (under *MC modularity*), three small community {101, 102, 103, 104}, {105, 106, 107, 108} and {109, 110, 111, 112} are identified correctly, vertices 1~100 forms two communities: one consists of {1, 72, 22, 36, 40, 63, 88, 90, 2, 7, 10, 16, 24, 41, 56, 59, 100}. In Figure 8(h) (under *NG modularity*), the clique {109, 110, 111, 112} is identified correctly, but the other two cliques are merged into {101, 102, 103, 104, 79} and {105, 106, 107, 108, 73, 77}. The other 97 vertices are divided into two communities.

Figure 8(i) and Figure 8(j) show the community structure of *Zachary's karate network* detected by *DC_MC algorithm*. In Figure 8(g) (under *MC modularity*), the classification of vertices except for vertex 10 fit well with the real one. In Figure 8(h) (under *NG modularity*), the algorithm identified a new community {24, 25, 26, 28, 29, 32}.

Figure 8(k) and Figure 8(l) show the community structure of *Lusseau's dolphins' network* detected by *DC_MC algorithm*. In Figure 8(k) (under *MC modularity*), the identification of vertices except for vertex 12, 22 and 23 fit well with the real one. In Figure 8(l) (under *NG modularity*), the algorithm identified a new community {12, 31, 32, 35~48} and vertices 22 and 23 are misidentified.

Figure 8(m) and Figure 8(n) show the community structure of *Les Misèrables' characters network* detected by applying *DC_MC algorithm*. In Figure 8(m) (under *MC modularity*), *DC_MC algorithm* have identified 5 communities, and they are different in size (the biggest community has 52 vertices, the smallest community only has 2 vertices). Vertices in these communities are densely interconnected, but sparsely connected with the rest of the network. In Figure 8(n) (under *NG modularity*), the differences in sizes amongst 5 communities detected by *DC_MC algorithm* are inapparent (the biggest community has 21 vertices, the smallest community has 10 vertices). In this community structure, vertices 12, 49, 56, 28 are appointed to different communities. In fact, vertices 12, 49, 56, 28 are connected to each other and their degrees are the largest, so it is not suitable to divide them into different communities from the point of view of the network structure.

The results of Figure 8 show that *MC modularity* can correctly evaluate the quality of discovered community structures.

5.3 Choosing candidate vertices

In order to test whether our approach of choosing candidate vertices is effective, we designed *synthetic network_5*, which consists of $(m-1)\times 1000+(m-1)\times 4$ vertices, of which $(m-1)\times 1000$ vertices form a large fully connected sub-graph, the other $(m-1)\times 4$ vertices form $m-1$ fully connected sub-graphs, and each contains 4 vertices. So m is the number of the fully connected sub-graphs. With the increase of m, the number of vertices and communities also increases. The structure of a network is shown in Figure 9. With respect to a fixed m, we run the *DC_MC algorithm* under two different strategies: the first chooses every vertex as candidate vertex to shift, and the other only chooses vertices that satisfy Formulas (12)–(17) as candidates to shift. It is intend to determine the difference of the running time of the *DC_MC algorithm* under the two different strategies.

Figure 10 shows the comparisons of the running times of the *MC_DC algorithm* on the *synthetic network_5*, in which the vertical coordinates represent the running times of the *MC_DC algorithm* corresponding to two different strategies under different numbers of vertices. From Figure 10, we can see that the running times of the *MC_DC algorithm* under the second strategy (only chooses vertices that satisfy Formulas (12)–(17) as candidates to shift) demonstrate linear trend, which indicates that our approach of choosing candidate vertices is effective.

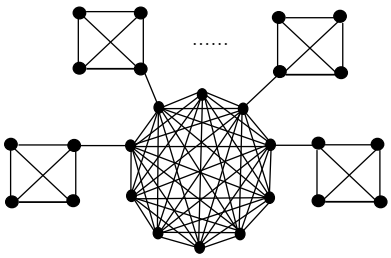


Figure 9. *synthetic network_5*

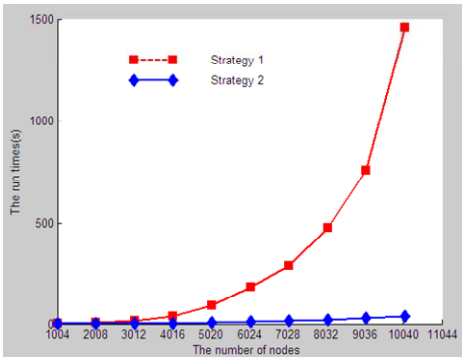


Figure 10. The log of running times of *MC_DC algorithm* corresponding to two different strategies under different nodes

6. Conclusion

To identify community structures within social networks is essential for further social network analysis. However, as we have demonstrated the existing approaches are not effective to detect community structures with significant size differences. We examined various possible approaches and proposed the concept of a coupling coefficient between two communities. This newly proposed concept can reflect the edge densities *between* two communities and the edge density *within* these two communities simultaneously. Based on this concept we defined a new modularity, *MC modularity*. *MC modularity* provides measures to evaluate the quality of the community structure detected even if the communities are very different in size. We further developed the *DC_MC algorithm* to detect community structures based on *MC modularity*, and a new algorithm for shifting nodes in a group, instead of one node at a time, amongst communities also is designed to achieve optimal results. Experiments on synthetic, computer-generated and real data sets have demonstrated that the *DC_MC algorithm* is capable to identify community structures with varied sizes from social networks, and revealed good performance features as well.

So far this study has only considered the basic networks. In the future, we plan to expend this work to directed, weighted and other more complicated social networks.

Acknowledgments

Lihua Zhou's work is supported by the National Natural Science Foundation of China under Grant No.61262069, No. 61272126, the Yunnan Educational Department Foundation under Grant No.2012C103, and Program for Young and Middle-aged Skeleton Teachers, Yunnan University.

References

[1] Lin, Y. R., Sun, J., Sundaram, H., Kelliher, A., Castro, P. and Konuru, R. (2011) Community discovery via metagraph factorization. ACM Transactions on Knowledge Discovery from Data, 5, 1-44.

[2] Wang, F., Li, T., Wang, X., Zhu, S. H. and Ding, C. (2011) Community discovery using nonnegative matrix factorization. Data Mining Knowledge Discovery, 22, 493-521.
[3] Duan, D. S., Li, Y. H., Li, R. X., Lu, Z. D. and Wen, A. M. (2013) MEI: Mutual enhanced infinite community-topic model for analyzing text-augmented social networks. The Computer Journal, 56, 336-354.
[4] Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. and Hwang, D. (2006) Complex networks: Structure and dynamics. Physics Reports, 424, 175-308.
[5] Guimerà, R. and Amaral, L. A. N. (2005) Functional cartography of complex metabolic networks. Nature, 433, 895-900.
[6] Zhao, Z. Y., Feng, S. Z., Wang, Q., Huang, J. Z., Williams, G. J. and Fan, J. P. (2012) Topic oriented community detection through social objects and link analysis in social networks. Knowledge-Based Systems, 26, 164-173.
[7] Newman, M. E. J. (2004) Fast algorithm for detecting community structure in networks. Physical Review E, 69, 066133.
[8] Fortunato, S. (2010) Community detection in graphs. Physics Reports, 486, 75-174.
[9] Newman, M. E. J. and Girvan, M. (2004) Finding and evaluating community structure in networks. Physical Review E, 69, 026113 .
[10] Arenas, A., Fernández, A., Fortunato, S. and Gómez, S. (2008) Motif-based communities in complex networks. Journal of Physics A: Mathematical and Theoretical, 41, 224001.
[11] Ghosh, R. and Lerman, K. (2010) Community detection using a measure of global influence. Advances in Social Network Mining and Analysis, 5498, 20-35.
[12] Li, Z. P., Zhang, S. H., Wang, R. S., Zhang, X. S. and Chen, L. N. (2008) Quantitative function for community detection. Physical Review E ,77, 036109.
[13] Good, B. H., Montjoye, Y. A. and Clauset, A. (2010) The performance of modularity maximization in practical contexts. Physical Review E, 81, 046106.
[14] Clauset, A., Newman, M. E. J. and Moore, C. (2004) Finding community structure in very large networks. Physical Review E, 70, 066111.
[15] Danon, L., Diaz-Guilera, A. and Arenas, A. (2006) Effect of size heterogeneity on community identification in complex networks. Journal of Statistical Mechanics: Theory and Experiment, P11010.
[16] Palla, G., Derényi, I., Farkas, I. and Vicsek, T. (2005) Uncovering the overlapping community structure of complex networks in nature and society. Nature, 435, 814-818.

[17] Leskovec, J., Lang, K. J., Dasgupta, A. and Mahoney, M. W. (2009) Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. Internet Mathematics, 6, 29-123.
[18] Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M, Nikolski, Z. and Wagner, D. (2008) On Modularity Clustering. IEEE Transactions on Knowledge and Data Engineering, 20, 172-188.
[19] Ciglan, M. and Nørvåg, K. (2010) Fast detection of size-constrained communities in large networks. Proceeding of the 11th international conference on Web information systems engineering, Hong Kong, China, 12-14 December, pp.~91-104. Springer-Verlag, Berlin.
[20] Duch, J. and Arenas, A. (2005) Community detection in complex networks using extremal optimization. Physical Review E, 72, 027104.
[21] Sun, Y., Danila, B., Josic, K. and Bassler, K. E. (2009) Improved community structure detection using a modified fine-tuning strategy. Europhysics Letters, 86, 1-6.
[22] Chen, W. Y. C., Dress, A. W. M. and Yu, W. Q. (2008) Community structures of networks. Mathematics in Computer Science,1, 441-457.
[23] Shi, C., Yan, Z. Y., Cai, Y. A. and Wu, B. (2012) Multi-objective community detection in complex networks. Applied Soft Computing, 12, 850-859.
[24] Zachary, W. W. (1977) An information flow model for conflict and fission in small groups. Journal of Anthropological Research, 33, 452-473.
[25] Lusseau, D. (2003) The emergent properties of a dolphin social network. Proceedings of the Royal Society B: Biological Sciences, 270, S186-S188.
[26] Knuth, D. E. (1993) The Stanford GraphBase: A Platform for Combinatorial Computing. ACM Press, New York.