Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

Eigenspaces of networks reveal the overlapping and hierarchical community structure more precisely

Xiaoke Ma[1], **Lin Gao**[1] **and Xuerong Yong**[2]

[1] School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi 710071, People's Republic of China
[2] Department of Mathematical Sciences, University of Puerto Rico at Mayaguez, PO Box 9018, PR 00681, USA
E-mail: maxiaoke8218@163.com, lgao@mail.xidian.edu.cn and xryong@math.uprm.edu

Received 29 April 2010
Accepted 15 July 2010
Published 12 August 2010

Online at stacks.iop.org/JSTAT/2010/P08012
doi:10.1088/1742-5468/2010/08/P08012

Abstract. Identifying community structure is fundamental for revealing the structure–functionality relationship in complex networks, and spectral algorithms have been shown to be powerful for this purpose. In a traditional spectral algorithm, each vertex of a network is embedded into a spectral space by making use of the eigenvectors of the adjacency matrix or Laplacian matrix of the graph. In this paper, a novel spectral approach for revealing the overlapping and hierarchical community structure of complex networks is proposed by not only using the eigenvalues and eigenvectors but also the properties of eigenspaces of the networks involved. This gives us a better characterization of community. We first show that the communicability between a pair of vertices can be rewritten in term of eigenspaces of a network. An agglomerative clustering algorithm is then presented to discover the hierarchical communities using the communicability matrix. Finally, these overlapping vertices are discovered with the corresponding eigenspaces, based on the fact that the vertices more densely connected amongst one another are more likely to be linked through short cycles. Compared with the traditional spectral algorithms, our algorithm can identify both the overlapping and hierarchical community without increasing the time complexity $O(n^3)$, where n is the size of the network. Furthermore, our algorithm can also distinguish the overlapping vertices from bridges. The method is tested by applying it to some computer-generated and real-world networks. The experimental results

1742-5468/10/P08012+26$30.00

indicate that our algorithm can reveal community structure more precisely than the traditional spectral approaches.

Keywords: analysis of algorithms, random graphs, networks

Contents

1. Introduction

In recent years, complex networks have emerged as a powerful tool for describing and analyzing many systems, such as social networks, technological networks, and biological networks [22, 20, 24, 46] etc. It has been shown that many real-world networks have a structure of modules or communities. Generally, a community is a subgraph with many edges connecting nodes of the same group and comparatively fewer links joining vertices of different groups. Modules or communities reflect topological relationships between the elements of the underlying system and functional entities. This is, for example, confirmed in the case of the social networks, where communities correspond to social groups with similar interests or backgrounds. In protein–protein interaction (PPI) networks, it is widely believed that the modular structure can be a functional unit or protein complex. Thus, accurately extracting community structures has a considerable merit in practice because it allows us to infer special relationship between nodes that may not be easily discovered by direct empirical tests.

Over the years, researchers have introduced a large number of algorithms, for example, the betweenness-based method [22], nonnegative matrix factorization (NMF) algorithms [50, 55], spectral clustering algorithms [39, 52, 36, 53], fuzzy clustering approaches [56, 37], the simulated annealing method [29], the semi-supervised clustering algorithm [28], information-theoretic based algorithm [45], fast algorithms [10, 42, 48], local search algorithm [3] etc. More algorithms for community detection can be referred to in [34, 13]. However, designing an efficient algorithm for identifying community is still highly nontrivial for many reasons.

The solution is hampered by the fact that there is no consensus criteria for measuring the community structure, which is a main drawback in many algorithms. To tackle this difficulty, Newman *et al* [33] introduced a modularity function Q which measures the quality of a given partition of a network, and it can also be utilized to select an automatically optimal number of communities based on the maximum Q value. Many algorithms based on such a strategy are proposed in [52, 14, 35], although finding the optimal Q value is an NP-hard problem [14, 7]. Recently, Fortunato *et al* [19] pointed out the serious resolution limit of the widely used Q function and claimed that the size of a detected community depends on the size of the whole network. Specifically, modularity maximization algorithms may fail to resolve communities with fewer than $\sqrt{L/2}$, where L is the number of edges in the entire network. To overcome the shortcomings of Q value: Li *et al* [23] addressed the problem by defining an alternative, called modularity density or D value; Medus *et al* [30] also tackled this issue by presenting new merit factors based on the weak and strong community definitions; Arenas *et al* [2] handled it by providing the user with a parameter that directs modularity maximization algorithms to search for communities of certain natural sizes; The newest work for this topic is presented by Berry *et al* [4]. They extended Fortunato and Barthélemy's argument with weighted edges and concluded that weighted modularity algorithms may fail to resolve communities with fewer than $\sqrt{\hbar\varepsilon/2}$ total edge weight, where $\hbar$ is the total edge weight in the network and ε is the maximum weighted of an inter-community edge. However, all these efforts are still far away from being a satisfactory answer.

Another difficulty is caused by the fact that communities may be in complicated shapes. Palla *et al* [38] revealed that complex network models exhibit an overlapping community structure, also called fuzzy community. Furthermore, recent studies [9, 41] suggested that many networks exhibit hierarchical organization. These overlapping and hierarchical communities are much more realistic than average ones. For instance, a protein in PPI networks has more than one function. Current algorithms [56, 34, 13, 43, 25, 37] are proposed for such realistic structures. However, only two of them can identify both the overlapping and hierarchical communities: it was Lancichinetti *et al* [25] who presented the first work dealing with this issue by optimizing a fitness function; another effort is proposed in the literature [49] based on finding the maximal clique problem. Both approaches achieve a good performance on some well-known data sets, but the randomness in the first algorithm and the high computational complexity in the latter algorithm are unacceptable.

Among these efforts on community, spectral algorithms have been shown to be powerful. Results obtained by spectral methods very often outperform those of other approaches, such as the K-means algorithm. The spectral approach is very simple to implement and can be solved by standard linear algebra methods. In this paper, we focus our attention on developing new algorithms to identify overlapping and hierarchical communities by extending traditional spectral methods. Traditional spectral methods use eigenvalues or eigenvectors of graph matrices. So far, most published spectral algorithms have operated with symmetric matrices, such as the adjacency matrices or Laplacian of an undirected graph. However, such symmetric matrices have the disadvantage that some topological information becomes invisible. To overcome this difficulty, a novel spectral approach was introduced [53], using complex eigenvalues and eigenvectors of the Laplacian matrix of a directed network corresponding to the original undirected one. It indicates that there is much information about the spectrum, besides the eigenvalues and eigenvectors, that can be extracted to identify the communities.

In many cases, the eigenvalues and eigenvectors of a network do not provide sufficient structural information. For example, the fact that a graph is reconstructible from its spectrum and a set of corresponding linearly independent eigenvectors points to the important role of eigenspaces in algebraic graph theory. Generally speaking, a basis of eigenvectors is far away from being an algebraic invariant, but eigenspaces themselves are invariant to within a permutation of the vertices of a graph. More explicitly, if two G and G' are co-spectral graphs with adjacency matrices A, A' respectively then G and G' are isomorphic if and only if there exists a permutation matrix P such that $P\xi_A(\mu) = \xi_{A'}(\mu)$ for each eigenvalue μ, where $\xi(\mu)$ is the eigenspace of μ. In the situation where the spectrum of a graph is not enough to grasp the topological structure, a natural way to overcome this fault is to bring into the consideration the eigenspaces of networks. This is one of the main motivations for the investigations reported in this paper, with an immediate purpose to identify communities by making use of the geometric attributes of eigenspaces. Another important reason is that there is no spectral method can detect both the overlapping and hierarchical communities. Thus, it is necessary and critical to extend the traditional spectral algorithms for complicated community structures.

This paper has three main contributions:

(i) We show that the communicability between a pair of vertices can be rewritten in term of the eigenspaces of networks, which means that the eigenspaces of networks have a direct connection with the communities in complex networks. It also serves as the theoretic foundation for our algorithm. By using the similarity based on the eigenspaces of networks, a hierarchical clustering is adopted to reveal the hierarchical communities.

(ii) A novel algorithm is presented to identify the overlapping vertices between different communities. The overlapping vertices are detected by counting the number of cycles starting from them, based on the fact that vertices more densely connected amongst one another are more likely to be linked through short cycles.

(iii) The complexity analysis indicates that our algorithm is equivalent to the traditional spectral algorithms in the computational time, while it can recognize the overlapping and hierarchical communities, simultaneously.

The paper is structured as follows: a short review of the concept of the eigenspaces of a network is introduced in section 2. In section 3, the concrete algorithm is presented. An typical example and the effects of parameters are discussed in section 4. The experimental results and conclusion are presented in sections 5 and 6, respectively.

2. Eigenspaces of a network: a short review

In order to introduce the concept of the eigenspaces of a graph, we should first give some basic definitions concerning graph theory. Let $G = (V, E)$ be a graph with a vertex set V and an edge set E. By a walk of length k in a graph we mean any sequence of (not necessarily different) vertices $v_1, \ldots, v_{k+1}$ such that for each $i = 1, 2, \ldots, k$ there is an edge from v_i to v_{i+1}. The walk is closed if $v_1 = v_{k+1}$. A walk is a path if all the vertices are distinct. A closed path is a cycle and by k-cycle denotes a cycle of length k. Given an undirected network $G = (V, E)$, an $n \times n$ adjacent matrix $A = (A_{ij})$ is constructed with $A_{ij} = 1$ if vertex v_i is connected to vertex v_j, 0 otherwise ($n = |V|$, i.e. n is the size of G). The characteristic polynomial $\det(xI - A)$ of the adjacency matrix A of G is called the *characteristic polynomial* of G, represented by $P_G(x)$. The eigenvalues of A, i.e. the zeros of $\det(xI - A)$, and the spectrum of A (which consists of the n eigenvalues) are also called the *eigenvalues* and the *spectrum* of G, respectively. The eigenvalues of G are denoted by $\lambda_1, \ldots, \lambda_n$; they are real because A is symmetric. Unless indicated otherwise, we shall assume that $\lambda_1 \geq \cdots \geq \lambda_n$. The eigenvalues of A are the numbers λ satisfying $A\mathbf{x} = \lambda\mathbf{x}$ for some non-zero vector $\mathbf{x} \in R^n$. Each such vector $\mathbf{x}$ is called an eigenvector of the matrix A belonging to the eigenvalue λ. If λ is an eigenvalue of A then the *eigenspace* of eigenvalue λ is $\{\mathbf{x} \in \Re^n : A\mathbf{x} = \lambda\mathbf{x}\}$, a *subspace* of $\Re^n$, denoted by $\xi(\lambda)$.

Because A is a symmetric matrix with real entries there exists an orthogonal matrix U such that $U^{\mathrm{T}}AU = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ and the columns of U are corresponding eigenvectors which form an orthonormal basis of $\Re^n$. Then A has the *spectral decomposition* [11]

$$A = \mu_1 P_1 + \mu_2 P_2 + \cdots + \mu_l P_l, \tag{1}$$

where $\mu_1, \ldots, \mu_l$ are the distinct eigenvalues of A and $P_i = UE_iU^{\mathrm{T}}$ with $E_i = \mathrm{diag}(0, \ldots, 0, I, 0, \ldots, 0), i = 1, \ldots, l$. For fixed i, if $\xi(\mu_i)$ has $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ as an orthonormal basis then

$$P_i = \mathbf{x}_1\mathbf{x}_1^{\mathrm{T}} + \cdots + \mathbf{x}_d\mathbf{x}_d^{\mathrm{T}}, \tag{2}$$

and P_i represents the orthogonal projection of $\Re^n$ onto $\xi(\mu_i)$ with respect to the standard orthonormal basis of $\Re^n$. Moreover, it is easy to verify that $P_i^2 = P_i = P_i^{\mathrm{T}}$.

Graph angle is an invariant of eigenspaces. Supposed that $\mu_1 > \cdots > \mu_l$, let $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ be the standard orthonormal basis of $\Re^n$ and β_{ij} be the angle between $\xi(\mu_i)$ and $\mathbf{e}_j$. Let $\alpha_{ij} = cos\beta_{ij} (i = 1, \ldots, l; j = 1, \ldots, n)$, usually, we abuse terminology and refer to the numbers α_{ij} as the *angle* of G. The $l \times n$ matrix (α_{ij}) is called the *angle matrix*. The angles with $\xi(\mu_i)$ can be computed from an orthonormal basis $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ as follows

$$\alpha_{ij} = \left(\sum_{h=1}^{d} x_{hj}^2\right)^{1/2}, \tag{3}$$

where $\mathbf{x}_h = (x_{h1}, x_{h2}, \ldots, x_{hn})^{\mathrm{T}}$. More information for eigenspaces of graphs can be referred to in [11]. In the next section, we will draw the relationship between eigenspaces of networks and their communities, and also show how the eigenspaces of a network can be used to identify the hierarchical and overlapping communities.

3. Algorithm

In this section, we first show that the communicability between a pair of vertices can be rewritten in term of eigenspaces of a network in section 3.1. A hierarchical clustering algorithm is proposed in section 3.2 to discover the hierarchical community. The procedures to characterize and detect the overlapping vertices are proposed in sections 3.3 and 3.4. Finally, the computational complexity is also proved.

3.1. Similarity measure based on eigenspaces

The reason why we pull in the concept of communicability is that it is a well-known tool to characterize the dynamical properties of complex networks. The communicability between a pair of vertices in a network is usually defined as the shortest path connecting them by assuming that most of the transportation on the network flows along the shortest paths. The moment $a_{ij}^k = (A^k)_{ij}$ gives the number of walks of length k starting at the vertex v_i and ending at the vertex v_j [11]. In [16], the definition is generalized as a weighted sum of the number of all walks connecting the pair of vertices. Specifically, C_{ij}, the communicability between v_i and v_j, is defined as a weighted sum of the moments a_{ij}^k [16]

$$C_{ij} = \sum_{k=0}^{\infty} \frac{a_{ij}^k}{k!} = \sum_{t=1}^{n} x_{ti} x_{tj} \mathrm{e}^{\lambda_t}, \tag{4}$$

where x_{ti} is the ith component of the tth eigenvector of A associated with the eigenvalue λ_t. Recently, a generalized communicability is presented as [17]

$$GC_{ij} = \sum_{k=0}^{\infty} \frac{a_{ij}^k \beta^k}{k!} = \sum_{t=1}^{n} x_{ti} x_{tj} \mathrm{e}^{\beta\lambda_t}, \tag{5}$$

where β is a variable parameter. For $\beta \ll 1$, the longer walks between two nodes are more severely penalized than in equation (4): only very short walks are accounted in the generalized communicability. When $\beta \gg 1$, the long walks receive greater weights.

Here, we rewrite the communicability in term of the eigenspaces of a graph. Since P_i is the orthogonal projection of $\Re^n$ onto $\xi(\mu_i)$, then we have $\alpha_{ij} = \|P_i \mathbf{e}_j\|$. From the spectral decomposition of A^k in equation (1), we have

$$a_{ij}^k = \sum_{t=1}^{l} \mu_t^k P_t \mathbf{e}_i \cdot P_t \mathbf{e}_j. \tag{6}$$

Substituting the above expression into equation (4), we obtain that

$$C_{ij} = \sum_{k=0}^{\infty} \frac{a_{ij}^k}{k!} = \sum_{t=1}^{l} P_t \mathbf{e}_i \cdot P_t \mathbf{e}_j \mathrm{e}^{\mu_t}. \tag{7}$$

In the same way, we can prove that the generalized communicability also can be rewritten in term of eigenspaces of networks as

$$GC_{ij} = \sum_{k=0}^{\infty} \frac{a_{ij}^k \beta^k}{k!} = \sum_{t=1}^{l} P_t \mathbf{e}_i \cdot P_t \mathbf{e}_j \mathrm{e}^{\beta\mu_t}. \tag{8}$$

Note that the equation (8) reduces to equation (7) at $\beta = 1$.

The equations (3), (8) show that the eigenspaces of a graph can be used to characterize the communicability of networks. In the following, we use the communicability between pairs of vertices as the similarity measure to discover communities. Different from others such as the NMF [55, 50, 28], fuzzy clustering [56, 37] and traditional spectral algorithms [39, 52, 36, 53], our algorithm consists of two steps: first, a partition $\{V_i\}_{i=1}^m$ of the V into m groups is obtained by using a hierarchical clustering algorithm based on the communicability matrix. Second, the overlapping vertices are uncovered by the density of short cycles.

3.2. Detecting the hierarchical communities

After constructing the similarity of a pair of vertices by using the eigenspaces of networks, a $n \times n$ communicability matrix GC is acquired with elements GC_{ij}.

With GC, we can group vertices of the network into communities using a hierarchical clustering algorithm [51]. Hierarchical clustering algorithms are classified into two categories: agglomerative and divisive. Agglomerative algorithms, also called bottom-up approaches, start with each vertex as a separate community and then progressively merge them into larger and larger groups. In contrast, divisive approaches, also called top-down algorithms, start with the whole network and proceed to divide it into smaller groups. Thus, the graph progressively splits into smaller and smaller disconnected subgraphs or clusters.

In addition to the similarity between pairs of vertices, another important factor in the hierarchical clustering algorithm is a similarity measure that determines which groups should be merged if the agglomerative algorithm is adopted, or where a cluster should be split if the divisive method is employed. There are three methods, such as single linkage, average linkage, and complete linkage, used to determine the similarity between two groups. In these measures the similarity of two groups is defined as the minimum, mean and maximum similarity of vertices of each group. Also, the similarity between groups is updated at each step of the algorithms. Finally, a dendrogram is constructed to visualize the procedure. The dendrogram of the network in the left panel of figure 2 is shown in figure 3. In this paper, the agglomerative algorithm with the average linkage is employed based on the communicability matrix GC.

To obtain a partition of V from the dendrogram, a measure for community should be chosen with the immediate purpose to guide the final partition. The modularity function Q [33] is the most popular choice. However, due to the resolution limit problem of Q, a new measure, called modularity density D [23], is selected because it can overcome the resolution limit to a large extent.

In the next section, we will utilize the eigenspaces of a graph to extract the overlapping communities from the hard partition $\{V_i\}_{i=1}^m$ obtained by the hierarchical clustering approach.

3.3. Detecting the overlapping communities

To identify the overlapping communities, there are two tasks that should be done: first, one should determine a subset $S \subseteq V$ in which each $v \in S$ is a candidate for overlapping. Second, for each $v \in S$ one should determine to which groups v belongs.

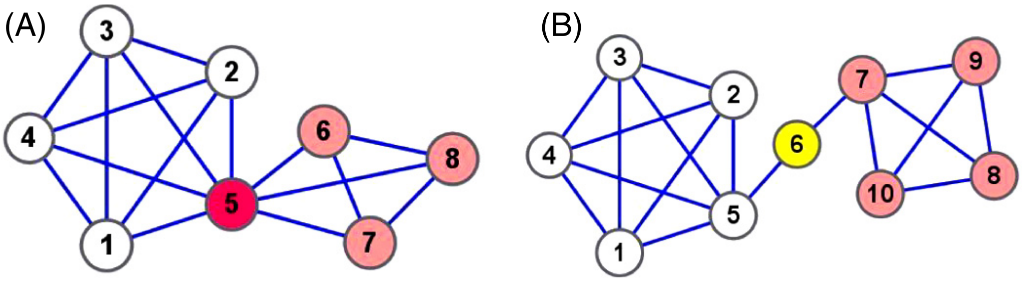

Figure 1. Schematic examples for overlapping and non-overlapping communities. Panel (left): there are two communities in the network distinguished by different colors and vertex 5 is overlapping; panel (right): a network consists of two communities connected by a bridge.

3.3.1. Characterize the overlapping vertex. To construct S, how to characterize the overlapping vertex is the prime objective. Perhaps, the simplest and most intuitive way is to set $S = V$, i.e., to check each vertex in the networks. This is, however, unacceptable for two reasons. The first one is that the computational complexity is too expensive. Actually, the overlapping vertices are much less than the size of G, i.e. $|S| \ll |V|$. Another reason is that setting $S = V$ throws away all the structure information of the networks, which results in many unnecessary blind search in the algorithm. Fortunately, there exists some prior knowledge that can be used to narrow $|S|$. An intuitive rule can be stated as

Rule 1. *If a vertex v is overlapped by two communities, then there is at least one adjacent vertex of v in each community.*

For example, in figure 1(A) there are two communities (one is 5-clique, the other is a 4-clique) distinguished by different colors, and the node 5 with the red color is overlapped. Only these vertices whose adjacent vertices are in other communities are selected. In this way, $|S|$ is much less than $|V|$. However, such a property is not sufficient to meet the requirement of our algorithm since it is too simple to draw a clear dividing line between the overlapping vertices and the 'bridge' ones (for the clear definition of bridge vertex refer to [37]). We take the simple graph shown in figure 1(B) as an example. A visual inspection of this graph most likely suggests two densely connected communities distinguished by different colors, the node 6 with the yellow color would be a 'bridge' one. To overcome this fault we take into account other properties based on eigenspaces of G to refine S obtained based on rule 1.

One of the most important features concerning complex networks is that cycles of different length underlie the connectivity of networks [44]. In this paper, a k-cycle is short if $k \leq \tau$, long otherwise, where τ is a threshold. Actually, the statistical distribution of cycles has been acknowledged as an important factor for defining the topology of the networks and the dynamic of systems running on such frameworks [1]. Generally, the density of cycles tends to increase as more edges are incorporated into networks, with longer cycles being observed earlier than shorter ones [12]. Therefore, the density of cycles of different lengths can be used as an indicator of the connectivity between any subset of nodes. In other words, the larger the number of shortest cycles among a subset of nodes, the more connected such nodes are to one another. For example, as shown in

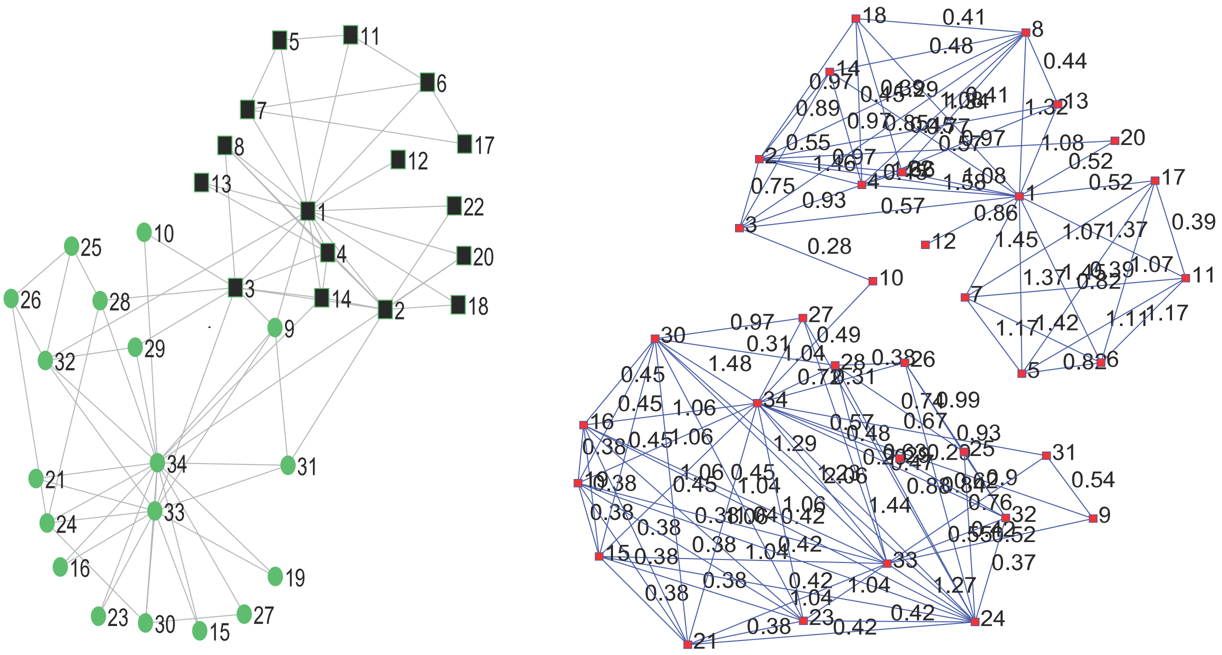

Figure 2. Left: Zachary's karate club network. Square nodes and circle nodes represent the administrator's faction and the instructor's faction, respectively. This figure is from Newman and Girvan [22]. Right: the communicability graph with threshold $\sigma = 0.28$ for the karate club network.

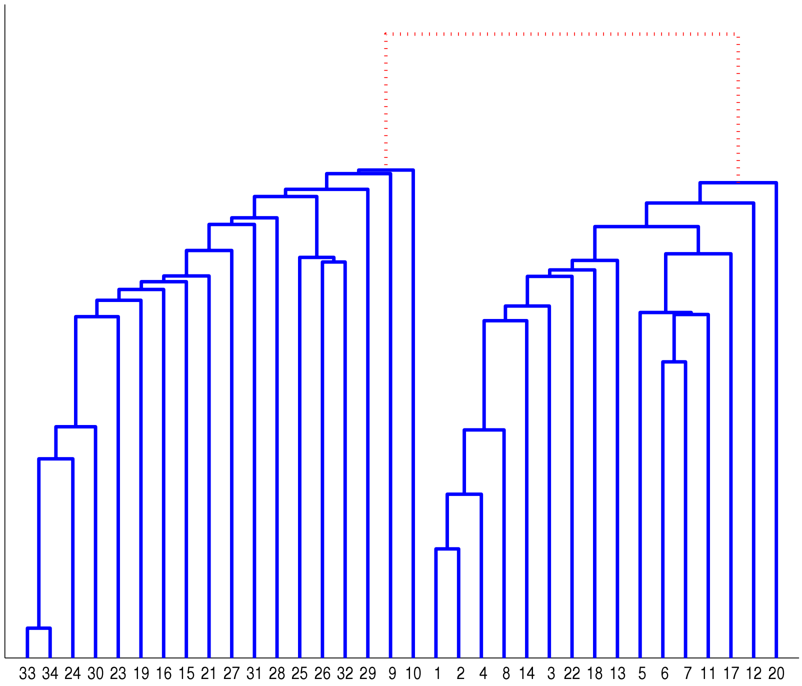

Figure 3. Dendrogram for Zachary's karate club using the generalized communicability matrix (with $\beta = 0.618$) between pairs of nodes and the average linkage method.

figure 1(A), the cycle concerning vertex 5 has 9 3-cycles while other vertices in the same community have only 6 3-cycles. At the same time, the bridge 6 in figure 1(B) has no cycles. So, this points out the direction to build the candidate set S, named Rule 2. Before giving the detailed description, the density of short cycles of a community V_i can be defined as

$$F(V_i) = \frac{\sum_{k=3}^{\tau} \sum_{v_j \in V_i} N_j^{(k)}}{|V_i|}, \tag{9}$$

where $N_j^{(k)}$ denotes the number of cycles at length k for each vertex v_j and τ is the upper bound for the length of cycles (we will discuss them later).

Rule 2. *Given a community V_i and a vertex $v \notin V_i$, v is overlapped by V_i if the density of short cycles in new community $V_i \cup v$ is larger than a scale of $F(V_i)$, i.e. $F(V_i \cup v) \geq \imath F(V_i)$ where $\imath$ is a scale, and the number of short cycles in new community increases, i.e. $F(V_i \cup v)(|V_i| + 1) > |V_i| F(V_i)$.*

Since it may take a long time to calculate the $N_i^{(k)}$ for each vertex v_i, we now propose a generalization of the $N_i^{(k)}$ by accounting for all the closed walks starting at vertex v_i. There are two reasons that can explain this modification. First, the cycles are not very sensitive with respect to the appearance of structural bottlenecks in a network. However, the number of walks can be affected significantly. Second, communication between two nodes in a network does not always take place through a path but it can follow non-path routes. Then the $N_i^{(k)}$ can be obtained from the graph angle as [11]

$$N_i^{(k)} = \sum_{j=1}^{m} \mu_j^k \alpha_{ji}^2. \tag{10}$$

There is another important parameter $\tau \in [3, d(G)]$ to distinguish the long cycles from the short, where $d(G)$ is the diameter of G. Notice that the choice of the threshold τ is very critical for the overlapping communities. We consider the length τ to be a tunable parameter, used to get the desired result when applied to a specific network. (In our experiment, we find that $3 \leq \tau \leq \sqrt{2} d(G)/2$ is a good choice, the possible explanation is that many complex networks are small world patterns. How the parameters $\beta, \tau, \imath$ effect the performance of algorithm is investigated in section 4.).

3.3.2. Merging the overlapping communities. With the two rules, a fuzzy partition $\{V_i^{(s)}\}_{i=1}^m$ of V can be obtained from the hard partition $\{V_i\}_{i=1}^m$. Sometimes, two soft communities share so many overlapping vertices that it is necessary to merge them into a larger one, which is related to the similarity between the communities in question. We adopt the Sørensen similarity index to characterize the degree of overlapping between communities $V_i^{(s)}$ and $V_j^{(s)}$, defined as $OV_{ij} = 2|V_i^{(s)} \cap V_j^{(s)}|/(|V_i^{(s)}| + |V_j^{(s)}|)$, where the numerator is the number of nodes in common in the two communities and the denominator gives the sum of the number of nodes in both communities. Obviously, the index is bounded as $0 \leq OV_{ij} \leq 1$. We can calculate the similarity index OV_{ij} for each pair of soft communities in $\{V_i^{(s)}\}_{i=1}^m$ and then represent it as a matrix $OV(\{V_i^{(s)}\}_{i=1}^m)$. The communities for which

$OV_{ij} \geq \eta$ need to be merged together into a larger one, where $0 \leq \eta \leq 1$ is a threshold. In this paper, we set $\eta = 0.5$.

The procedure of our algorithm is presented in the algorithm 1.

```
Algorithm 1: algorithm for the hierarchical and overlapping communities

Input:
    $A$: the adjacent matrix of $G = (V, E)$;
    $\beta$: the weight for cycles;
    $\tau$: the upper bound for the length of a short cycle;
    $\imath$: the threshold for the density of cycles in a community.

Output:
    $\{V_i\}_{i=1}^{m}$: a hard partition of $V$, $m$ is the number of clusters in hard partition;
    $\{V_i^{(s)}\}_{i=1}^{m'}$: a soft partition of $V$, $m'$ is the number of clusters in soft partition;

    Preparation work:
01. Compute the eigenvalues and eigenvectors of $G$ as
    $\begin{pmatrix} \mu_1 & \cdots & \mu_l \\ n_1 & \cdots & n_l \\ \{\mathbf{x}_1^{(\mu_1)}, \ldots, \mathbf{x}_{n_1}^{(\mu_1)}\} & \cdots & \{\mathbf{x}_1^{(\mu_l)}, \ldots, \mathbf{x}_{n_l}^{(\mu_l)}\} \end{pmatrix};$
02. For each eigenvalue $\mu_i$, construct the orthogonal projection $P_i$ of $\Re^n$ onto $\xi(\mu_i)$ according to equation (2);
03. Construct the angle matrix according to equation (3);

    Sub-algorithm 1: detecting hierarchical community:
04. Compute the communicability matrix $GC = (GC_{ij})$ according to equation (8);
05. Using the agglomerative hierarchical clustering algorithm with the average linkage to obtain a dendrogram based on $GC$;
06. A hard partition $\{V_i\}_i^m$ is attained with the maximum $D$ value [23];

    Sub-algorithm 2: detecting overlapping community:
07. Construct the candidate set $S$ according to the Rule 1 and Rule 2 based on $\{V_i\}_{i=1}^{m}$;
08. For each vertex $v \in S, v \in V_i$, for each different community $V_j$, if the new community induced by $V_j \cup v$ has more short cycles than that of original one, we set $V_j^{(s)} = V_j \cup v$;
09. Compute the similarity matrix $OV$ to merge these communities whose overlapped ratio $\geq \eta$ together into a larger one.
10. Return $\{V_i\}_{i=1}^{m}$ and $\{V_i^{(s)}\}_{i=1}^{m'}$;
```

3.4. Computational complexity

The computational complexity of our algorithm is investigated in this subsection. First of all, the space complexity is analyzed. Given a network G, a $n \times n$ adjacency matrix A is needed to store G. It needs space $O(n^2)$. We also need two $2 \times m$ vectors to store the spectrum of G with space $O(m)$. Since $\sum_{i=1}^{l} n_i = n$, the space for eigenvectors is

$O(n^2)$. There are also l projection matrices P_i, which require space $O(ln^2)$ bounded by $O(n^3)$ since $l \leq n$. The space for angle matrix GC and communicability matrix are $O(ln)$ and $O(n^2)$, respectively. We also need $O(\tau n)$ to store $N_i^{(k)}(1 \leq i \leq n, 3 \leq k \leq \tau)$. Since $\tau \leq d(G)$, the space for $N_i^{(k)}$ reduces to $O(d(G)n)$. The space for matrix OV is $O((m')^2)$, which is bounded by $O(m^2)$ since $m' \leq m$. Thus, the memory complexity of our algorithm is $O(n^2) + O(m) + (n^2) + O(d(G)n) + O(n^3) = O(n^3)$. It indicates that our algorithm is efficient in space.

As shown in algorithm 1, our algorithm consists of three major components. In preparation work, the time to determine the eigenvalues and eigenvector in step 1 is $O(n^3)$ [21], and that for orthogonal projection P_i $(1 \leq i \leq l)$ is $O(ln^2)$. Based on the $P_i(1 \leq i \leq l)$, one can quickly obtain the communicability matrix C in time $O(n^2)$. In the sub-algorithm for the hierarchical community, the only time is the hierarchical clustering algorithm. As we known, it is $O(n^2 \log n)$ [51]. In the third component, the time to construct S is $O(n^2)$ because it is linear to compute $N_i^{(k)}$ $(i = 1, 2, \ldots, n)$ based on angle matrix. Thus, the time complexity of our algorithm is $O(n^3)$, equivalent to that of the traditional spectral algorithms.

We would like to state that as the algorithm introduced in this paper uses not only the eigenvalues and eigenvectors of networks involved, but also uses the properties of the eigenspaces, this give us a better characterization of the community. A careful comparison indicates that there at least two aspects distinguishing our algorithm from the traditional spectral algorithm:

- In the traditional spectral clustering algorithms, each vertex is embedded into a spectrum space by using directly the eigenvectors of networks. Our algorithm, however, is based on the eigenspaces of networks by making use of the geometric attributes of eigenspaces. It is very reasonable to discover communities with the eigenspaces of networks for two reasons: first, eigenspaces embrace much more structure information than the eigenvectors and eigenvalues of networks. Second, the spectrum of networks is very sensitive to subtle change in networks, for example, deleting or adding a new edge may result in disconnection of networks associated with the λ_2. However, the eigenspaces are more stable than the spectrum of networks [11].
- Traditional spectral clustering algorithms for community detection only discover either the overlapping community or the hierarchical community, while our algorithm can identify both communities with equivalent time complexity.

4. An illustrative example: the karate club network

In the previous section we have shown how the eigenspaces of networks can be used to characterize the hierarchical and overlapping communities and prescribed the algorithm to extract these complicated communities. In the present section, we first demonstrate the algorithm. Since our algorithm employs several user-defined parameters β, τ and $\imath$, we will investigate how the variation of these parameters affects the performance of our algorithm with an example of a social network known as the Zachary karate club [54]. There are two reasons why the club network is selected: first, it has independent information about the partition, therefore we can judge whether the results obtained by our algorithm are reasonable or not. Second, since there are several vertices that have been acknowledged as overlapping, we can make use of them to demonstrate the performance of our algorithm to detect the overlapping communities. For these reasons, it has become a benchmark for all methods of community detection [22, 56, 14, 34, 28, 17, 36, 55, 50].

4.1. Procedure of our algorithm

The network consists of 34 members of a karate club as nodes and 78 edges representing friendship between members of the club which was observed over a period of two years. Due to a disagreement between the club's administrators and instructors, the club is split into two smaller ones. The network is shown in the left panel of figure 2, where the squares and the circles label the members of the two groups.

As shown in algorithm 1, for any fixed β the communicability matrix GC for the club network is obtained. Given a network $G = (V, E)$, to further reveal the communicability, the communicability graph is adopted [17] as $\Phi(G) = (V, E_{\Phi(G)})$ with $E_{\Phi(G)} = \{(i, j) : i, j \in V, GC_{ij} \geq \sigma\}$, where σ is a threshold. A typical example is shown in the right panel of figure 2 with $\beta = 0.5$ and $\sigma = 0.28$, in which the weight on each edge is the generalized communicability value between the corresponding vertex pairs. From the communicability graph, one may easily conclude that the communicability in the dense subgraph is much better than that of the sparse region, i.e. the communicability between vertex pair $(3, 10)$ is the minimum value in this graph since there is only one edge connecting them.

With the communicability matrix GC, an agglomerative hierarchical clustering algorithm with the average linkage is used to obtain a dendrogram, as shown in figure 3. A hard partition of V is then obtained with the help of the D value. From the dendrogram, we can easily figure out that our algorithm divides the network into two parts. Other algorithms often divide the karate club network into three [56] or four communities [23, 28, 29], which indicates that it is topological meaningful. But, our result is consistent with the natural communities shown on the left of figure 2.

After obtaining the hard partition, for each community we compute its density of short cycles according to equation (9). Then, for each vertex $v \in S$, we determine whether it is overlapping or not, based on the requirements of Rule 2. By setting $\tau = 6, \imath = 6.5$, a soft partition is constructed, as shown in figure 4. From it, we find that nodes 3, 9 are overlapped by two communities, and node 10 is a bridge. In other algorithms [34, 56], node 10 is overlapping. However, we have two reasons to consider that our result is more reasonable. The first reason is that since vertex 10 has two edges connecting each community with one edge, it makes no contribution to each community. Another is that vertex 10 has no 3-cycle in either community while both node 3 and 9 do have.

4.2. Effects of the parameters

Here, we study the effect of the parameters $\beta, \tau, \imath$, on the structure of communities in the karate club networks. Recall that the parameter β determines the hard partition of the networks since it takes charge of the weight on a walk connecting a pair of vertices. We now investigate how the variation of β affects the performance of our algorithm in discovering the hierarchical communities. Since the community structures in these networks are known, we employ the normalized mutual information (NMI) index as a measure of similarity between two partitions [15]. In detail, it is based on defining a

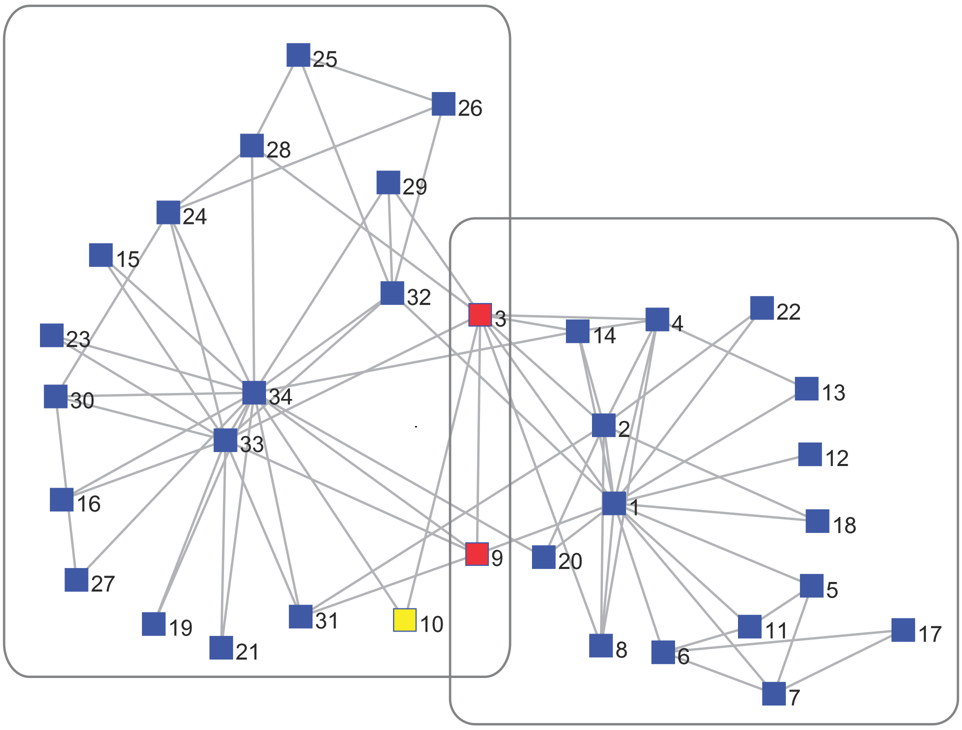

Figure 4. The result of our method applied to the karate club network with $\tau = 3, \imath = 6.5$. Two different communities are discovered, each enclosed by a square. The nodes 3 and 9 (with red color) are overlapped by two communities and node 10 (with yellow color) is identified as a bridge.

confusion matrix $\mathbf{N}$ whose rows correspond to the real community in standard partition P and the columns correspond to the communities in the obtained partition P'. The element of $\mathbf{N}$, N_{ij} is the number of vertices in the ith real community that appear in the jth found community. The NMI is defined as

$$\mathrm{NMI}(P, P') = \frac{-2\sum_{i=1}^{|P|}\sum_{j=1}^{|P'|} N_{ij}\log(\frac{N_{ij}N}{N_{i.}N_{.j}})}{\sum_{i=1}^{|P|} N_{i.}\log(\frac{N_{i.}}{N}) + \sum_{i=1}^{|P'|} N_{.j}\log(\frac{N_{.j}}{N})}, \qquad (11)$$

where $|P|$ and $|P'|$ denote the number of communities in P and P' respectively, $N_{i.}$ is the sum of the ith row of N and the sum over column j is denoted by $N_{.j}$.

Figure 5 shows various dendrograms with different values of β, and the plot of the NMI for different values of β. When $\beta = 0$, the dendrogram in figure 5(A) shows that there are no communities since the communicability between arbitrary vertex pair is zero. The value of NMI corresponding to this condition is 0.046, very close to 0. However, at $\beta = 0.2$, there are two obvious communities, as shown in figure 5(B), and the corresponding NMI value increases to 0.740. As the value of β increases, the value of NMI also starts to increase, i.e. when $\beta \geq 0.3$, the NMI is 1.000, as shown in figure 5(E) (within the shadow region). In an ideal situation, $\beta \to \infty$, there is only one community in the network. Although the NMI value is perfect when $\beta \geq 4$, carefully comparing figures 5(A) and (E) demonstrates that as β increases from 0 to 1, the interaction among vertices becomes stronger and stronger. For example, when β is small, say $\beta < 0.3$, the interaction among the nodes is

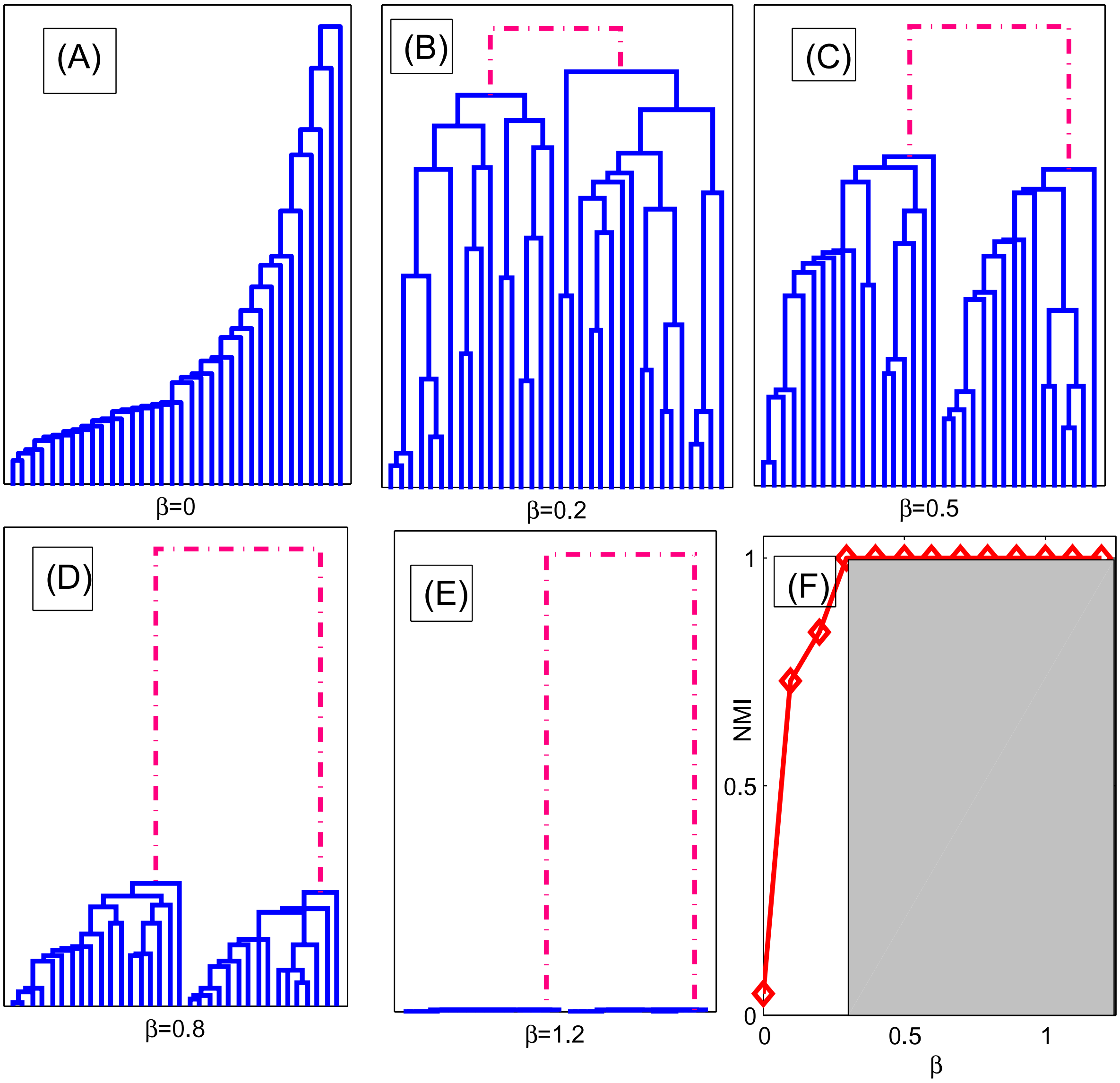

Figure 5. The effect of β. Parts (A)–(E) are various dendrograms with different β. Part (F) is the plot of the normalized mutual information for different values of β.

weak (shown in figure 5(B)), while when β is large, say $\beta \geq 1.2$, the interaction is strong (shown in figure 5(E)).

There is a good reason to explain the above result. As shown in the section 3.1, when β is small, only these short walks between a pair of nodes receive enough weights and hence dissect the loosely bound communities. As β increases, the contributions of long walks dominate in communicability between a vertex pair; therefore, wide-ranging communities which are loosely bound internally are detected.

After discussing the effect of β on the performance of the hard partition, we investigate the effects of parameters $\tau, \imath$ on the performance of extracting the overlapping nodes. Before giving the discussion in detail, we first introduce an evaluation measure for two-

Table 1. A confusion matrix for a two-class classification.

		True Classification	
		$\mathbf{X}$	$\overline{\mathbf{X}}$
Algorithm	$\mathbf{X}$	a	b
Results	$\overline{\mathbf{X}}$	c	d

class classification problem to assess the performance of our algorithm [18]. In detail, suppose an algorithm is supplied $|W|$ objects, each is classified as either $\mathbf{X}$ or $\overline{\mathbf{X}}$ (not $\mathbf{X}$). Table 1 discloses the performance of the algorithm in terms of the 2×2 confusion matrix, augmented by its row and column totals. In truth, there are $a + c$ $\mathbf{X}$s and $b + d$ $\overline{\mathbf{X}}$s; the algorithm believes that there are $a+b$ $\mathbf{X}$s and $c+d$ $\overline{\mathbf{X}}$s. There are $a+d$ correct, and $b+c$ confused classifications. Based on table 1, the performance metrics are defined as:

- Accuracy Acc $= (a + d)/(a + b + c + d)$ denotes the probability of objects classified correctly.
- True Positive Rate $(\mathrm{Acc}^{+}) = a/(a + c)$: if the algorithm classifies the object as an $\mathbf{X}$, how likely is it to be an $\mathbf{X}$?
- True Negative Rate $(\mathrm{Acc}^{-}) = d/(b + d)$. If the object is an $\overline{\mathbf{X}}$, how likely is the algorithm to classify it as an $\overline{\mathbf{X}}$?
- Precision (Pre) $= a/(a + b)$ denotes the probability of a correctly classified object to be an $\mathbf{X}$.

In the karate club network, we set $W = N(V_1) \bigcup N(V_2)$, where $\{V_1, V_2\}$ is the partition of V shown in figure 3 and $N(V_i) = \{v \in V : v \notin V_i, \exists u \in V_i, (u, v) \in E\}(i = 1, 2)$. It is easy to find that $W = \{1, 2, 3, 9, 10, 14, 20, 28, 29, 31, 32, 33, 34\}$. Although there is no consensus on the overlapping vertices in the club network, nodes 3 and 9 are recognized as overlapping. Thus, it is reasonable to set $\mathbf{X} = \{3, 9\}$ in the true classification. Let S denote the set of overlapping vertices detected by our algorithm. Figure 6 shows various plots of Acc and Pre for different values of τ and $\imath$. When $\imath$ is small, say $i \leq 5$, our algorithm detects $S = \{3, 9, 14, 20, 34\}$, and as the τ increases from 3 to 7, S is unchanged (shown in figure 6(A)). However, from figure 6(B) we can see that when $\imath = 6.3$, nodes 3,9,34 are extracted as overlapping under $\tau = 3$. Furthermore, when $\tau \geq 4$, vertices 14, 20 are included in S. As the value of $\imath$ increases, the values of Acc and Pre start to increase. In other words, the set of the detected overlapping vertices is more and more close to $\{3, 9\}$ as $\imath$ increases. For instance, at $\imath = 0.65$ and $\tau = 3$, or 4, our algorithm can identify the overlapping vertices 3 and 9 precisely (as shown in figure 6(C)). If we fixed $\imath = 6.5$ and keep increasing the value of τ, the values of Acc and Pre decrease because other nodes, such as 14 and 20, can satisfy the requirement of rule 2. When $\imath$ is large enough, some of the overlapping vertices are filtered mistakenly. This is, for example, confirmed in the case shown in figure 6(D), where $\imath = 8$, and only vertex 9 is identified as overlapping. In an ideal situation where $\imath$ is large enough, $\imath \geq 8$, the set of overlapping vertices detected by our algorithm is empty, i.e. no overlapping vertex is discovered.

We can interpret the above results in two ways. First, as explained in the section 3.3, the density of the short cycles can be used to characterize the community structure. This

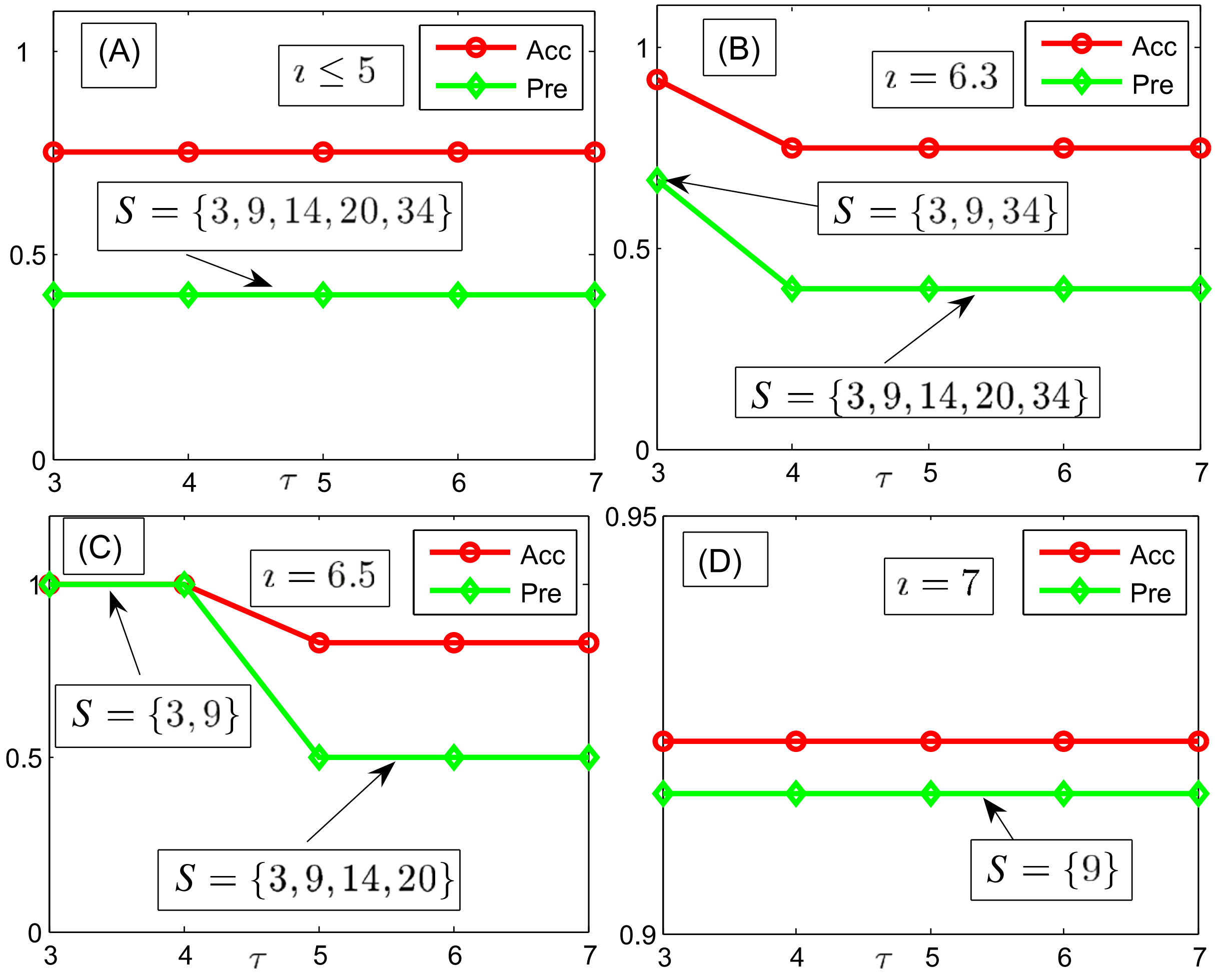

Figure 6. The effect of $\tau, \imath$. Parts (A)–(D) are the various plots of the Acc and Pre for different values of τ and $\imath$.

is confirmed by the fact that a careful contrast of the four plots in figure 6 indicates that when $\tau = 3$ or 4, our algorithm obtains a good performance in discovering the overlapping vertices. Second, given a community V_1 and a vertex $v \notin V_1$, the parameter $\imath$ represents the 'strength' of the interaction between v and V_1. If $\imath$ is too small, many non-overlapping vertices cannot be filtered. When $\imath$ is too large, many overlapping vertices are filtered since they do not make enough contribution to increase the density of short cycles in the new community.

5. Experiment results

In this section, we first test the performance of our method for computer-generated networks with hierarchical community structure. Then, we apply our method to a real network. Finally, the resolution limit problem is also investigated. The experimental results show that our algorithm is superior to traditional spectral clustering algorithms. The algorithm is coded using MATLAB version 7.1.

5.1. Artificial network

In many networks, communities join together to form larger ones, and these large communities are joined together and so on in a hierarchical fashion. The method introduced in this paper has the ability to reveal the hierarchical structure of the networks. We adopted a benchmark [25] similar to that recently proposed by Sale-Pardo *et al* [47]. It is a simple extension of the classical benchmark proposed by Girvan and Newman [22]. There are 512 nodes, arranged in 16 groups of 32 nodes each. The 16 groups are ordered into 4 supergroups. Each vertex has average of k_1 links with the remaining 31 partners of its group and k_2 links with the 96 nodes of the other three groups within the same supergroup. In addition, each node has a number k_3 of edges with the rest of network. In this way, two hierarchical levels emerge: one consisting of the 16 small groups, and one of the supergroups with 128 nodes. The degree of mixing of the four supergroups is expressed by the parameter k_3, which we tune freely. Also, the mixing of the small communities is varied by the ratio k_1/k_2. In this experiment, we set $k_1 = k_2 = 16$, so that the micro-communities are fuzzy and pose a hard test to our algorithm.

To test the performance of our algorithm, we considered different values of k_3: for each value we built 100 realizations of the network. Obviously, as k_3 increases, the community structure is more and more vague. Since the community structures in these networks are known, we employ the NMI as a measure of precision, as shown in equation (11). To make a comparative analysis, four algorithms are selected deliberately: the Infomap approach [40], Blondel algorithm [8], Spectral clustering algorithm [36] and NMF algorithm [50]. The recent paper [27] presents a detailed analysis of the most used algorithms, finding that the best performances on both GN and LFR benchmarks are obtained by Infomap[3] followed by the Blondel algorithm[4], that is why these two algorithms are selected. Since our algorithm is an extension of the spectral clustering algorithm, the spectral algorithm presented by Newmann is adopted with the immediate purpose to show the superiority of our algorithm. The reason why the NMF algorithm is employed is that it is a novel algorithm and has been extensively applied to community detection problems [28, 55], obtaining impressive performance.

In figure 7 we plot the average values of the NMI as functions of k_3 for the two hierarchical levels. The left plot of figure 2 is the result for the four supergroups or macro-communities, while figure 7(B) is for the 16 small communities. From figure 7(A), we can see that the Infomap algorithm achieves the best performance when $k_3 \leq 24$, while its performance drops dramatically when $k_3 \geq 25$. The possible reason is that, as k_3 increases, the minimum code length is not adequate to characterize the community structure. A similar problem also occurs in the Blondel algorithm. The spectral algorithm, NMF and our algorithm are inferior to the Infomap and Blondel algorithms when $k_3 \leq 34$, but superior to them when $k_3 \geq 26$. These three algorithms have competitive performance when $25 \leq k_3 \leq 31$. However, when $k_3 \geq 31$ the performance of the three algorithms decreases dramatically. The reason is that when $k_3 \leq 30$, the community structure is so obvious that can be easily detected. When $k_3 \sim 32$, each node has 32 links inside and 32 outside of its macro-community and it is difficult to detect these fuzzy communities. Furthermore, we can conclude that our algorithm is superior to the spectral algorithm

[3] The code is downloaded from http://www.tp.umu.se/∼rosvall/

[4] The code is downloaded from http://sites.google.com/site/findcommunities/

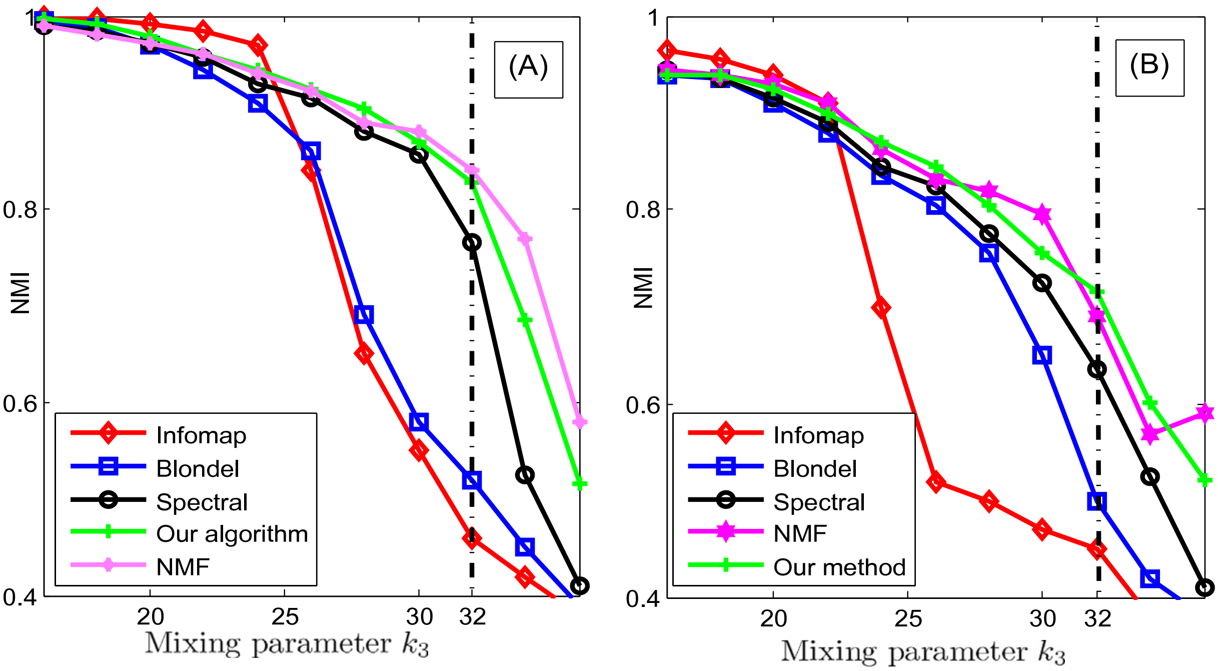

Figure 7. Test of the accuracy of our algorithm on a hierarchical benchmark. The normalized mutual information is used to compare the partition found by algorithms with the natural partition of the networks at each level. At the higher level, the communities are four clusters, each including four clusters of 32 vertices, for a total of 128 vertices per cluster. At the lower level, the communities are 16 clusters of 32 nodes each. (A) Comparison between the NMF algorithm, spectral algorithm, Blondel algorithm, Infomap algorithm and our algorithm on macro-communities. (B) Comparison among these five algorithms on micro-communities. The value of β in our algorithm is 0.5.

when $k_3 > 25$. The probable explanation is that the community structure is more and more difficult to detect when $k_3 \geq 26$, the spectrum cannot provide enough information to discover these predefined communities while the eigenspaces contain more information about network structure. Compared with the NMF algorithm, our algorithm is better when $k_3 \leq 31$ and inferior to it when $k_3 \geq 32$. The reason is that the graph eigenspaces may not characterize all these changes in networks. From figure 7(B), we can conclude that our algorithm has comparable performance with the NMF algorithm. A careful contrast of the two plots in figure 7 shows that all these methods have a much better accuracy in macro-communities than in micro-communities since $k_1 = k_2 = 16$ implies fuzzy communities.

5.2. Protein–protein interaction network

To expand the application of our algorithm, a PPI network of yeast S. cerevisiae is investigated with the immediate purpose to demonstrate that the presented method can also be used to detect functional units in a biological network. This network consists of

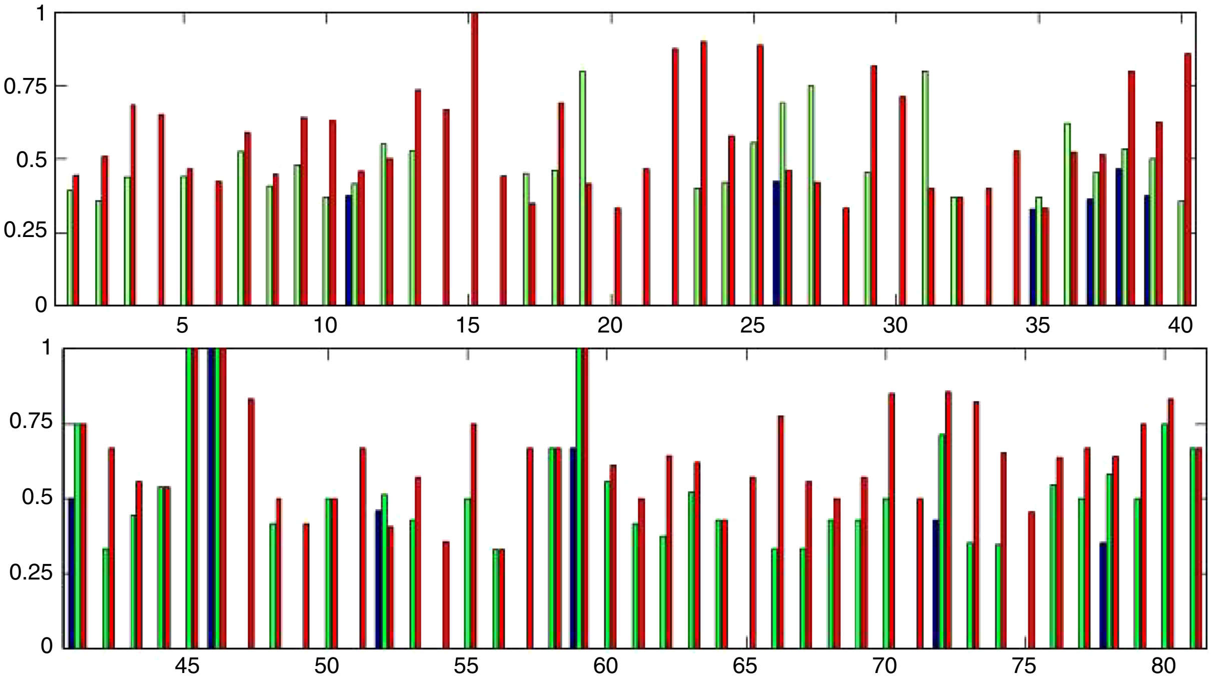

Figure 8. The matching ratio of the modules discovered by our algorithm with MIPS functional categories by analysis of constituent protein functions. Those modules with more than one matched functional categories have one ratio for each function category.

1257 proteins (vertices) and 6835 interactions (edges) which have a high confidence score not lower than 0.3. It is, however, suggested that the sizes of the obtained modules are too big and not very biologically reasonable in the PPI network [31]. Thus, we restrict the average size of discovered communities to roughly 12.

By setting $\beta = 0.6, \imath = 6.3, \tau = 4$, we applied our algorithm to the PPI network. The proposed method found 103 communities in the PPI network. Among the 103 modules discovered by our algorithm, there are some modules with no more than three nodes; these cannot be biological modules since they are common even in random networks, so these modules are excluded. Thus, our algorithm identifies 81 functional modules in total. The result concerning the matching ratio is summarized in figure 8. From figure 8, we can easily see that most of the remaining modules match MIPS function categories [32] with high consistency and most of the communities match with more than one function category. The main reason is that most proteins are involved in multiple biological processes. To further reveal the final results, we give some extracted modules (see figure 9) where vertices with the same color imply that those protein have an identical function and proteins with multiple colors have multiple functions. From figure 9, we can see that figure 9(A) involves one main function while figure 9(B) corresponds to multiple functions. We also present a large module in figure 9(C). Modules discovered without function annotation can be predicted according to the main functions in their communities. Thus, the module in figure 9(C) can be predicted to have a metabolic function. This experiment indicates that our algorithm can also be used to discover the modules in real PPI.

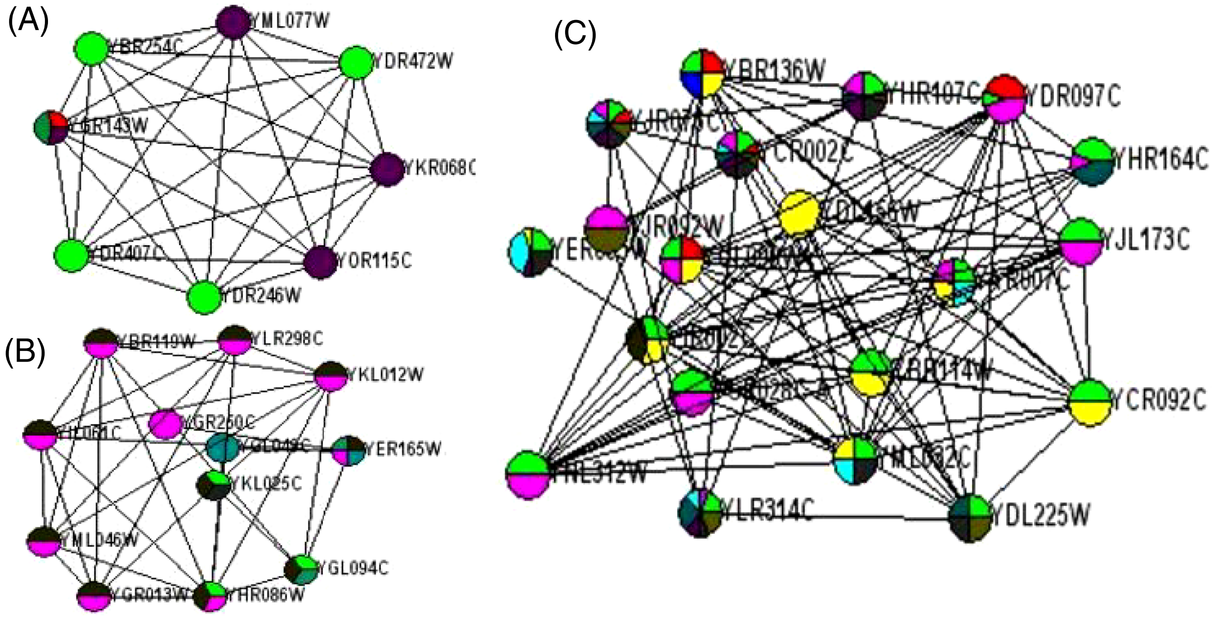

Figure 9. Modules of the PPI network discovered by our algorithm. (A) Most proteins in these modules involve transportation; (B) proteins in these modules have two functions: transcription and protein binding; (C) large module has three functions: metabolism, DNA processing, and cell rescue.

5.3. Resolution limit problem

Modularity optimization algorithms are criticized for a serious resolution limit problem [19], in which small communities are merged into a large community. The ability to escape the resolution limit problem is becoming an important aspect for an excellent algorithm for community structure [28, 23, 53, 26]. The main goal of this experiment is to test the performance of our algorithm to tolerate the resolution limit.

The GN benchmark is a popular choice for many algorithms. It is, however, not an appropriate alternative for resolution limit analysis because all the nodes of the network have essentially the same degree, the communities are all of the same size. The LFR benchmark [26] is adopted because it can produce communities with both small and large sizes. It is a similar spirit to the GN benchmark, but is much more realistic. It allows the user to specify distributions both for the community size and the degree distribution, then generates vertices and communities by sampling from those distributions. The LFR generators then rewire the graph in order to constrain the average ratio of intra-community adjacencies to total adjacencies. The ratio is denoted by μ. At $\mu = 0$, all the edges are intra-community. Of course, LFR is also not a perfect benchmark since the intended natural communities are often fractured into several unintended sub-communities at small values of μ. It is difficult to evaluate community detection algorithms with such a deceptive ground truth [4].

We set the input parameters for the FLR benchmark networks as: the average degree is 2, the maximum degree is 50, the exponent of the degree distribution is 2 and that of the community size distribution is 1. In this experiment, we set $\beta = 0.5$.

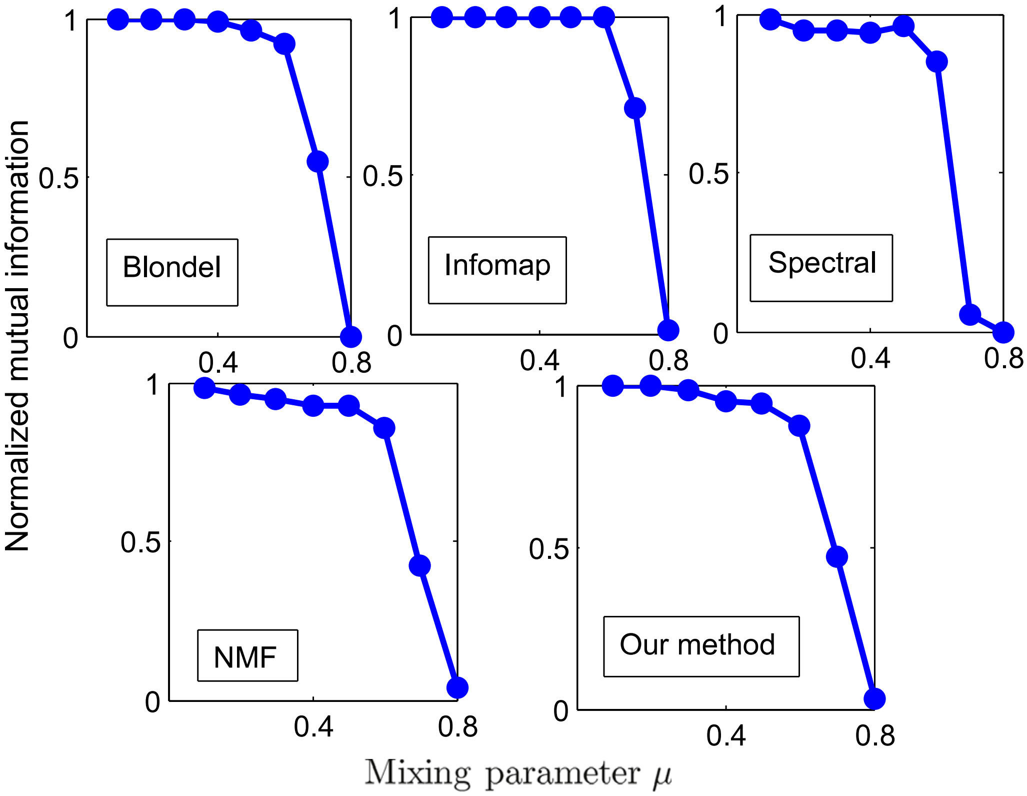

Figure 10. Comparison of Infomap, the Blondel method, NMF, the spectral clustering algorithm and our algorithm on the undirected and unweighted LFR benchmark, each point is an average over 100 instances.

The average results on 100 networks are summarized in figure 10, which are plots of normalized mutual information as a function of the mixing parameter μ with various algorithms. From figure 10, we can conclude that Infomap outperforms the others absolutely. In particular, it is capable of extracting all the communities in 100% of cases when $\mu \leq 0.5$. The Blondel algorithm also achieves an excellent performance, largely due to the maximum estimated modularity strategy. Compared with the Blondel algorithm, the spectral method is inferior, because the spectral methods achieve a good performance when the eigenvalue is a step function instead of a straight line. Our algorithm outperforms the spectral algorithm and obtains a comparable performance with the Blondel approach. There two reasons to explain why: first, compared to the spectrum and eigenvectors of networks, the eigenspaces of networks can grasp much more information about structure; second, the spectrum and eigenvectors are much more sensitive to perturbations of the network structure. The NMF algorithm does not have an impressive performance because the time complexity is very high when the number of clusters is large, thus affecting the final result.

In addition to the classification accuracy, we are also interested in comparing the distributions of the sizes of communities discovered by the algorithms to the original distributions used in the LFR generation. Figure 11 contains these results. The target line is the empirical cumulative distribution function (CDF) for the natural

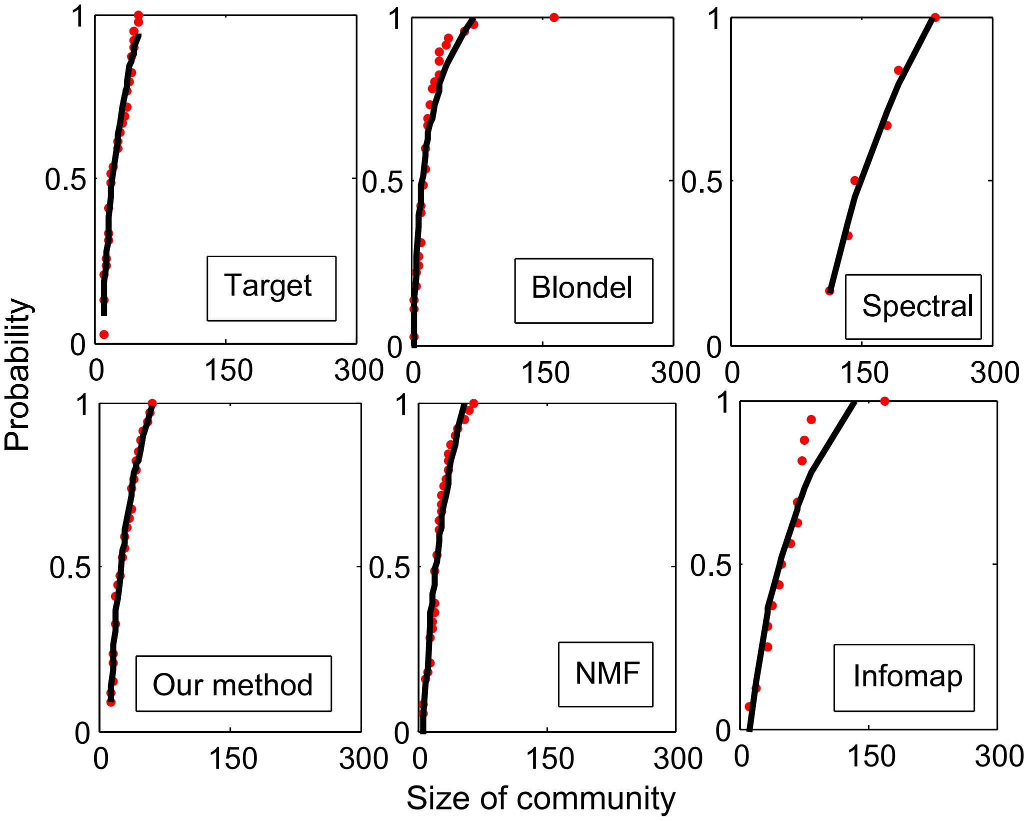

Figure 11. The distributions of natural community sizes and discovered community sizes with Infomap, the Blondel algorithm, spectral algorithm, NMF algorithm and our algorithm with $\mu = 0.7$.

community sizes sampled by LFR. The other lines are empirical CDFs for the sizes of communities discovered by the Blondel algorithm, spectral clustering, NMF, Infomap and our algorithm. To make the results clear, we also determine the curves fitting the distributions (see figure 11). Compared with the target, the lines of the Blondel and NMF algorithms are closer to 0, indicating that the two approaches find some communities that are smaller than the minimum natural size. In other words, the two algorithms split the natural community into smaller ones. Among the five algorithms, spectral clustering achieves a poor performance since the curved line deviates greatly from that of the target. There are two reasons to explain this: first, when $\mu = 0.7$, the eigenvectors of the networks are insufficient to characterize the community structure. Second, the spectral algorithm is a Q-driven method. Compared with the spectral algorithm, the Blondel algorithm can achieve a good performance although it is also a modularity optimization method, probably because the estimated modularity maximum is not a very good approximation of the real one. Finally, our algorithm and NMF are superior to the others as the curves of these two algorithms are approximately parallel to that of the target while the others deviate wildly from the target. Such an experiment indicates that our algorithm can tolerate the resolution limit in average networks to a large extent.

6. Conclusion

In this paper we proposed a novel spectral clustering algorithm for discovering the overlapping and hierarchical community structure of complex networks. Unlike previous spectral algorithms which use the eigenvalues or eigenvectors of graph matrices, such as the adjacent matrix or Laplacian matrix, to identify the community, our algorithm works by not only using the eigenvalues and eigenvectors but also the geometric attributes of the eigenspaces of the networks.

We showed that the generalized communicability between a pair of vertices can be rewritten in terms of the graph angle. It indicates that the eigenspaces of networks have direct connections with the community. Therefore, the generalized communicability is adopted as the similarity measure in networks. We then used the hierarchical clustering algorithm to obtain communities. Based on the fact that vertices more densely connected amongst one another are more likely to be linked through short cycles, we identified the overlapped nodes by counting the number of short cycles. We have tested our method on some computer-generated graphs and also on two real-world networks and found that the results are excellent, in agreement with our expectations. Compared with traditional spectral clustering methods, our algorithm is more accurate without increasing the computational complexity. Therefore, it is reasonable to assert that the eigenspaces of networks can be used to reveal communities.

We would like to close this paper by posing some further research directions:

- Even though the time complexity of our algorithm equals that of the traditional spectral methods, $O(n^3)$ is still unacceptable for large and dense networks. As shown in section 3.4, the eigenvector computation is still the most computationally expensive step of the method. For large and dense networks, it would be interesting to determine how to make the networks sparse without destroying the community structure in order to make use of our algorithm. Furthermore, studying the community structure predicted by our algorithm in fast growing networks like the Internet, where it is known that the spectral properties rescale with the network size, would also be very interesting [5].
- In this algorithm, the communities are uncovered by studying the loop structure of the networks. Recently, Bianconi *et al* [6] have pointed out that although community detection algorithms have a limit of detectability, entropy measures are useful to assess the relevance of different community assignments. In future work, we will check the significance of our algorithm with respect to others using entropy measures, particularly in the case where the correct communities are not known *a priori*.

Thus, designing effective and efficient methods which can solve these problems will be very important and interesting.

Acknowledgments

This work was supported by the National Key Natural Science Foundation of China (Grant No. 60933009), the Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 200807010013) and the National 863 Plan Project of China (Grant No. 2009AA01Z420). The authors thank Professor M E J Newman for providing the data of the karate club network. We also thank Professors X S Zhang, S H Zhang and

R S Wang for providing the PPI network data. The authors would like to thank the reviewers for their valuable comments and suggestions.

References

[1] Arenas A, Diaz-Guilera A and Perez-Vicente C J, *Synchronization reveals topological scales in complex networks*, 2006 *Phys. Rev. Lett.* **96** 114102

[2] Arenas A, Fernandez A and Gomez S, *Analysis of the structure of complex networks at different resolution level*, 2008 *New J. Phys.* **10** 053039

[3] Bagrow J P, *Evaluating local community methods in networks*, 2008 *J. Stat. Mech.* P05001

[4] Berry J W *et al*, *Tolerating the community detection resolution limit with edge weighting*, 2009 arXiv:0903.1072

[5] Bianconi G, Caldarelli G and Capocci A, *Loops structure of the internet at the autonomous system level*, 2005 *Phys. Rev.* E **71** 066116

[6] Bianconi G, Pin P and Marsili M, *Assessing the relevance of node features for network structure*, 2009 *Proc. Nat. Acad. Sci.* **106** 11433

[7] Brandes U *et al*, *On modularity clustering*, 2008 *IEEE Tran. Know. Data Eng.* **20** 172

[8] Blondel V D, Guillaume J L, Lambiotte R and Lefebvre E, *Fast unfolding of communities in large networks*, 2008 *J. Stat. Mech.* P10008

[9] Clauset A, Moore C and Newman, *Hierarchical structure and the prediction of missing links in networks*, 2008 *Nature* **453** 98

[10] Clauset A, Newman M E J and Moore C, *Finding community structure in very large networks*, 2004 *Phys. Rev.* E **70** 066111

[11] Cvetković D, Rowlinson P and Simić S, 1997 *Eigenspaces of Graphs* (Cambridge: Cambridge University Press)

[12] Da Fontoura Costa L, *The L-percolation of complex networks*, 2004 *Phys. Rev.* E **70** 056106

[13] Danon L *et al*, *Comparing community structure identification*, 2005 *J. Stat. Mech.* P09008

[14] Duch J and Arenas A, *Community identification using extremal optimization*, 2005 *Phys. Rev.* E **72** 027104

[15] Danon J *et al*, *Comparing community structure identification*, 2005 *J. Stat. Mech.* P09008

[16] Estrada E and Hatano N, *Communicability in complex networks*, 2008 *Phys. Rev.* E **77** 036111

[17] Estrada E and Hatano N, *Communicability graph and community structures in complex networks*, 2009 *Appl. Math. Comput.* **214** 500

[18] Forbes A D, *Classification-algorithm evaluation: five performance measures based on confusion matrices*, 1995 *J. Clin. Monit.* **11** 183

[19] Fortunato S and Barthélemy M, *Resolution limit in community detection*, 2007 *Proc. Nat. Acad. Sci.* **104** 36

[20] Gibson D, Kleinberg J and Raghavan P, *Inferring Web communities from link topology*, 1998 *Proc. Int. Conf. on Hypertext and Hypermedia* p 225

[21] Golub G H and Van Loan C F, 1996 *Matrix Computation* (Baltimore, MD: Johns Hopkins University Press)

[22] Girvan M and Newmann M E J, *Community structure in social and biological networks*, 2002 *Proc. Nat. Acad. Sci.* **99** 7821

[23] Li Z P *et al*, *Quanitative function for community detection*, 2008 *Phys. Rev.* E **77** 036109

[24] Lu H C *et al*, *The interactome as a tree-an attempt to visualize the protein–protein interaction networks in yeast*, 2004 *Nucl. Acid Res.* **32** 4804

[25] Lancichinetti A, Fortunato S and Kertész J, *Detecting the overlapping and hierarchical community structures of complex networks*, 2008 *New J. Phys.* **11** 033015

[26] Lancichinetti A, Fourtunato S and Radicchi F, *Bechmark graphs for testing community detection algorithm*, 2008 *Phys. Rev.* E **78** 046110

[27] Lancichinetti A and Fortunato S, *Community detection algorithms: a comparative analysis*, 2009 *Phys. Rev.* E **80** 056117

[28] Ma X K *et al*, *Semi-supervised clustering algorithm for community structure detection in complex networks*, 2010 *Physica* A **389** 187

[29] Medus A D, Acuña G and Dorso C O, *Detection of community structures in networks via global optimization*, 2005 *Physica* A **358** 593

[30] Medus A D and Dorso C O, *Alternative approach to community detection in networks*, 2009 *Phys. Rev.* E **79** 066111

[31] Muff S, Rao F and Caflsch A, *Local modularity measure for network clusterizations*, 2005 *Phys. Rev.* E **72** 056107

[32] Mewes H W *et al*, *MIPS: a database for genomes and protein sequences*, 2002 *Nucl. Acids Res.* **32** 31

[33] Newmann M E J and Girvan M, *Finding and evaluating community structure in networks*, 2004 *Phys. Rev.* E **69** 026113

[34] Newmann M E J, *Detecting community structure in networks*, 2008 *Eur. Phys. J.* B **38** 321

[35] Newmann M E J, *Modularity and community structure in networks*, 2006 *Proc. Nat. Acad. Sci.* **103** 8577

[36] Newmann M E J, *Finding community structure in networks using the eigenvectors of matrices*, 2006 *Phys. Rev.* E **74** 036104

[37] Nepusz T *et al*, *Fuzzy communities and the concept of bridgeness in complex networks*, 2008 *Phys. Rev.* E **77** 016107

[38] Palla G *et al*, *Uncovering the overlapping community structure of complex networks in nature and society*, 2005 *Nature* **435** 814

[39] Pujol J M, Béjar J and Delgado J, *Clustering algorithms for determining community structure in large networks*, 2006 *Phys. Rev.* E **74** 016107

[40] Rosvall M and Bergstrom C T, *Maps of random walks on complex networks reveal community structure*, 2008 *Proc. Nat. Acad. Sci.* **105** 1118

[41] Ravasz E *et al*, *Hierarchical organization of modularity in metabolic networks*, 2002 *Science* **297** 1551

[42] Radicchi F *et al*, *Defining and identifying communities in networks*, 2004 *Proc. Nat. Acad. Sci.* **101** 2658

[43] Reichardt J and Bornholdt S, *Detecting fuzzy community structures in complex networks with a Potts model*, 2004 *Phys. Rev. Lett.* **93** 218701

[44] Rozenfeld H D *et al*, *Statistics of cycles: how loopy is your network?*, 2005 *J. Phys. A: Math. Gen.* **38** 4589

[45] Rosvall M and Bergstron C T, *An information-theoretic framework for resolving community structure in complex networks*, 2007 *Proc. Nat. Acad. Sci.* **104** 7327

[46] Samanta M P and Liang S, *Predicting protein functions from redundancies in large-scale protein interaction networks*, 2003 *Proc. Nat. Acad. Sci.* **100** 12579

[47] Sales-Pardo M, Guimerá R and Amaral L A N, *Extracting the hierarchical organization of complex systems*, 2007 *Proc. Nat. Acad. Sci.* **104** 15224

[48] Sun S *et al*, *Faster and more accurate global protein function assignment from protein interaction networks using the MFGO algorithm*, 2006 *FEBS Lett.* **580** 1891

[49] Shen H W *et al*, *Detect overlapping and hierachical community structure in networks*, 2009 *Physica* A **388** 1706

[50] Wang R S *et al*, *Clustering complex networks and biological networks by nonnegative matrix factorization with various similarity measures*, 2008 *Neurcomputing* **72** 134

[51] Wasserman S and Faust K, 1994 *Social Network Analysis* (Cambridge: Cambridge University Press) p 26

[52] White S and Smyth P, *A spectral clustering approach to finding communities in graphs*, 2005 *SIAM ICDM*

[53] Zarei M, Samani K A and Omidi G R, *Complex eigenvectors of network matrices give better insight into the community structure*, 2009 *J. Stat. Mech.* P10018

[54] Zachary W W, *An information flow model for conflict and fission in small groups*, 1977 *J. Anth. Res.* **33** 452

[55] Zhang S H, Wang R S and Zhang X S, *Uncovering fuzzy community structure in complex networks*, 2007 *Phys. Rev.* E **76** 046103

[56] Zhang S H, Wang R S and Zhang X S, *Identification of overlapping community structure in complex networks using fuzzy c-means clustering*, 2007 *Physica* A **374** 483