Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 9850927, 10 pages
http://dx.doi.org/10.1155/2016/9850927

Research Article

Semisupervised Community Detection by Voltage Drops

Min Ji,[1] Dawei Zhang,[1] Fuding Xie,[2] Ying Zhang,[3] Yong Zhang,[1] and Jun Yang[2]

[1]*School of Computer Science and Technology, Liaoning Normal University, Dalian, Liaoning 116081, China*
[2]*School of Urban and Environmental Science, Liaoning Normal University, Dalian, Liaoning 116029, China*
[3]*College of Business Administration, Dalian University of Finance and Economics, Dalian, Liaoning 116622, China*

Correspondence should be addressed to Fuding Xie; xiefd@lnnu.edu.cn

Received 3 November 2015; Revised 28 December 2015; Accepted 30 March 2016

Academic Editor: Pubudu N. Pathirana

Many applications show that semisupervised community detection is one of the important topics and has attracted considerable attention in the study of complex network. In this paper, based on notion of voltage drops and discrete potential theory, a simple and fast semisupervised community detection algorithm is proposed. The label propagation through discrete potential transmission is accomplished by using voltage drops. The complexity of the proposal is $O(|V| + |E|)$ for the sparse network with $|V|$ vertices and $|E|$ edges. The obtained voltage value of a vertex can be reflected clearly in the relationship between the vertex and community. The experimental results on four real networks and three benchmarks indicate that the proposed algorithm is effective and flexible. Furthermore, this algorithm is easily applied to graph-based machine learning methods.

1. Introduction

From the point of view of mathematics, many real-world systems in nature and society can be effectively modeled as complex networks or graphs. Specifically, the entities of the system are represented by the vertices and the interactions between the entities are represented by the edges. Examples include social relationships, spreading of viruses and diseases, the World Wide Web, author cooperation networks, citation networks, and biochemical networks. It has been shown that many real-world networks have a structure of modules or communities, where the nodes within a community are higher connected to each other than the nodes among communities. The community structures play an important role in the functional properties of complex network, and finding such a structure could be of significant practical importance.

Identifying community structure in special networks has a considerable merit of practice because it gives us insights to the structure-functionality relationship. In the past decades, plenty of techniques have been proposed to detect the community structure hidden in networks. The more typical algorithms for community detection can be found in [1]. Very recently, Chen et al. [2] defined the antimodularity as a quantitative measure of anticommunity partitioning on a network and showed the reliability of antimodularity as a measurement of the quality of an anticommunity partitioning. A vertices similarity probability model to find community structure without the prior knowledge of the type of complex network structure was presented [3]. By studying the community structure in Chinese character network, Zhang et al. [4] found that community structure was always considered as one of the most significant features in complex networks, and it played an important role in the topology and function of the networks. Palla et al. [5] revealed that complex network models exhibited an overlapping community structure, also called fuzzy community. These complicated structures actually make it harder to appropriately construct algorithms to uncover them. Along this way, researchers have made great contributions to the community detection [6–10].

The methods mentioned above belong to unsupervised community detection methods since the topological information of the network is used only and its background knowledge is ignored. In fact, some prior information is of great value in identifying the community structure. Based on the discussion of an equivalence of the objective functions of the symmetric nonnegative matrix factorization and the maximum optimization of modularity density, Ma et al. [11] introduced a semisupervised clustering algorithm for

community structure detection. In [12], Silva and Zhao presented a technique for semisupervised classification tasks, by using the modularity measure of complex networks, originally designed for unsupervised learning tasks. Zhang [13, 14] developed a method that implicitly encoded the pairwise constraints by modifying the adjacency matrix of the network, which could also be regarded as the denoising process of the consensus matrix of the community structures. A novel semisupervised community detection algorithm was proposed based on the discrete potential theory [15]. It effectively incorporated individual labels, the labels of corresponding communities, to guide the community detection process for achieving better accuracy. Although these existing semisupervised community detection methods can improve the community identification accuracy, some of them have limitations in high time complexity. Therefore, it is worthwhile to introduce the novel algorithm to identify community structures in complex network rapidly.

The application of discrete potential theory to detect community in network can be traced back to Wu and Huberman's work [16]. They presented a method that allowed for the discovery of communities within graphs of arbitrary size in times that scale linearly with their size. Their method was based on notions of voltage drops across networks that were both intuitive and easy to solve regardless of the complexity of the graph involved. Zhang et al. [17] applied it to directed networks; they presented a new mechanism for the local organization of directed networks and designed the corresponding link prediction algorithm. Wang and Zhang [18] came up with a semisupervised clustering method based on generalized point charge models for text data classification. Liu et al. [15] recently proposed a linear time algorithm to find the community in network based on discrete potential theory. As data sets get larger and larger, it is still necessary to develop the efficient semisupervised learning methods.

Motivated by Wu and Huberman's work [16] and Liu's method [15], in this paper, we present a simple and fast semisupervised algorithm for detecting the community in complex network by discrete potential theory and voltage drops. The complexity of the proposed algorithm is $O(|V| + |E|)$ for the sparse network with $|V|$ vertices and $|E|$ edges. Similar to the membership degree in fuzzy c-means algorithm, the voltage value of each vertex in network implies the relationship between this vertex and community. The main contributions of the proposal are as follows: (1) the proposed algorithm is a simple and fast semisupervised approach to discover community structures in complex network. (2) The proposal gets rid of the limitation of positive definite matrix which is needed to solve a linear system by conjugate gradient decent algorithm. To some extent, this approach remedies the deficiency of Liu et al.'s work [15]. (3) The unsupervised Wu-Huberman algorithm is extended to semisupervised learning case. The experimental results demonstrate the effectiveness of the proposed algorithm.

2. The Graph and Discrete Potential Method

The graph can be mathematically represented as $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices and $E \subset V \times V$ denotes the set of edges. Generally, the graph can be expressed by its adjacent matrix A, whose elements A_{ij} are equal to 1 if v_i points to v_j and 0 otherwise. We denote d_i as the degree of vertex v_i. The degree matrix D is a diagonal matrix containing the vertex degree d_i $(i = 1, 2, \ldots, n)$ of a graph on the diagonal. Then the Laplacian matrix L can be defined as

$$L = D - A. \tag{1}$$

Denote x_i^s as the potential of vertex v_i in the electrostatic field generated by vertices with label s. Assign the potentials of all labeled vertices with labels other than s to zero and the s-labeled vertices to have a unit potential. The process of potential transmission for each electrostatic field is a circuit theory problem and can be modeled by combinatorial Dirichlet [15]. By using the Laplacian matrix L, a combinatorial formulation of the Dirichlet integral is in the form [15, 19]

$$D[x] = \frac{1}{2} x^T L x, \tag{2}$$

where x is the potentials of all vertices minimizing (2). Reassigning the order of all vertices of the graph and putting the labeled vertices forward, (2) can be rewritten into

$$\begin{aligned} D[x] &= \frac{1}{2} \begin{bmatrix} x_L^T, x_U^T \end{bmatrix} \begin{bmatrix} L_L & B \\ B^T & L_U \end{bmatrix} \begin{bmatrix} x_L \\ x_U \end{bmatrix} \\ &= \frac{1}{2} \left(x_L^T L_L x_L + 2 x_U^T B^T x_L + x_U^T L_U x_U \right), \end{aligned} \tag{3}$$

where x_L and x_U are two vectors whose elements represent the potentials of labeled vertices and unlabeled vertices, respectively. Setting the derivative of $D[x]$ with respect to x_U equal to zero, one can obtain a system of linear equations

$$L_U x_U = -B^T x_L, \tag{4}$$

where x_U is a $|V_U|$ dimensional vector whose elements are unknown quantities needing to be solved. If the graph is connected, or if every connected component contains a seed, then (4) will be nonsingular.

For each label s, a system of linear equations can be established as

$$L_U x^s = -B^T p^s. \tag{5}$$

If one assigns a unit potential to the labeled vertices with label s and zero to other labeled vertices, it will generate an electrostatic field. The potentials of unlabeled vertices can be obtained by the solution of (5). By comparing the potentials of each unlabeled vertex, its label is assigned the same as the labeled vertex corresponding to the greatest potential. Thus the community structure is detected.

From the perspective of discrete potential theory, the solution to (5) can be interpreted as a circuit theory. Based on the three fundamental equations of circuit theory, Kirchhoff's Current Law, Ohm's Law, and Kirchhoff's Voltage Law, one can also get an equivalent system of (5) [15, 19].

In [15], the solutions of (5) have been obtained by conjugate gradient decent algorithm, and a novel semisupervised community detection algorithm was proposed. Several experimental results demonstrate the effectiveness of their approach.

3. The Proposed Algorithm

It should be noted that the coefficient matrix L_U in (5) must be a symmetric positive definite matrix while solving the nonhomogeneous linear equations (5) by conjugate gradient decent algorithm. Obviously, Laplacian matrix L is not a positive definite matrix since every row of the Laplacian matrix sums to zero, 0 is always its eigenvalue, and the corresponding eigenvector is $(1, 1, \ldots, 1)$. This fact compels us to develop a new method to detect communities in network while considering the network as an electric circuit. In [16], Wu and Huberman introduced an unsupervised method to solve the system like (5) to discover the communities in complex network in linear time. Since there is no class information in advance, they employed bipartite strategy and some superb skills for the case of multiple communities. In this work, we extend their work to the case of semisupervised community detection.

In what follows, we would like to present a novel method to find community structure in complex networks by the process of voltage transmission.

For a given network, we suppose each edge to be a resistor with the same resistance. One attaches all the labeled vertices with label s to anode of a battery and other labeled vertices to negative pole so that they have fixed voltages, say 1 and 0. Based on these assumptions, the network can be viewed as an electric circuit with current flowing through each edge (resistor). By solving Kirchhoff equations, one can obtain the voltage value of each unlabeled vertex which of course should be within $(0, 1)$. In this case, the voltage value of each vertex can be thought of as the membership degree similar as in FCM algorithm, which reflects clearly the relationship between a vertex and the sth community. In turn, we can get k voltages of a vertex for the different labels if there are k classes. In semisupervised learning methods, it is required that at least one sample must be labeled in each class. This indicates that the class parameter k is known previously.

Physically, if node v_i connects to m neighbors $v_1, v_2, \ldots, v_m$ in an electric circuit, the Kirchhoff equation [20] tells us that the total current flowing into v should sum up to zero; that is,

$$\sum_{j=1}^{m} I_j = \sum_{j=1}^{m} \frac{V_j - V_i}{R} = 0, \tag{6}$$

where I_j is the current flowing from v_j to v and V_j is the voltage at neighbor node v_j.

It is easy to rewrite (6) into the following form:

$$V_i = \frac{1}{m} \sum_{j=1}^{m} V_j. \tag{7}$$

That is to say, the voltage of a node is the average of those voltages of its neighbors.

Suppose the number of communities to be k; then the label set $K = \{1, 2, \ldots, k\}$. In addition, we also assume that there must be at least one labeled vertex in each community. Divide the vertex set V into two parts, $V_L = \{(v_1, y_1), (v_2, y_2), \ldots, (v_l, y_l)\}$ (labeled vertices), where $y_i \in K$ is the label of vertex v_i and $V_U = \{v_{l+1}, \ldots, v_n\}$ (unlabeled vertices) such that $V_L \cap V_U = \emptyset$. One also defines set $V_L^s = \{v_i \mid v_i$ with label $s\} = \{v_i \mid v_i \in V_L,\ y_i = s\}$. Denote V_i^s as the voltage of vertex v_i in the electrostatic field generated by vertices with label s and $N(v_i)$ as the set of neighbors of v_i.

If we reassign the order of all vertices of the graph and put the labeled vertices forward and labeled vertices with label s first, following (6), one can get the system:

$$V_i^s = 1, \quad \text{for } i = 1, 2, \ldots, r, \tag{8}$$

$$V_i^s = 0, \quad \text{for } i = r+1, r+2, \ldots, l, \tag{9}$$

$$V_i^s = \frac{1}{d_i} \sum_{v_j \in N(v_i)} V_j^s = \frac{1}{d_i} \sum_j V_j^s A_{ij}, \quad \text{for } i = l+1, l+2, \ldots, n. \tag{10}$$

Equation (10) is a linear system with $n - l$ variables and can be put into a symmetrical form as follows:

$$V_i^s = \frac{1}{d_i} \sum_{j=l+1}^{n} V_j^s A_{ij} + \frac{1}{d_i} A_{i1} + \cdots + \frac{1}{d_i} A_{ir}, \quad \text{for } i = l+1, l+2, \ldots, n. \tag{11}$$

Define

$$V^s = \begin{pmatrix} V_{l+1}^s \\ \vdots \\ V_n^s \end{pmatrix}, \quad B = \begin{pmatrix} \frac{A_{l+1l+1}}{d_{l+1}} & \cdots & \frac{A_{l+1n}}{d_{l+1}} \\ \vdots & & \vdots \\ \frac{A_{nl+1}}{d_n} & \cdots & \frac{A_{nn}}{d_n} \end{pmatrix}, \quad C = \begin{pmatrix} \frac{A_{l+11} + \cdots + A_{l+1r}}{d_{l+1}} \\ \vdots \\ \frac{A_{n1} + \cdots + A_{nr}}{d_n} \end{pmatrix}, \tag{12}$$

and then the matrix form of Kirchhoff equation is

$$V^s = BV + C, \tag{13}$$

which has the solution

$$V^s = (I - B)^{-1} C. \tag{14}$$

Generally, it will take $O((n-l)^3)$ time to solve this system. Wu-Huberman algorithm [16] skillfully avoids this difficulty by solving (8)–(10) for $r = 1$ and $l = 2$. This method seems naturally to be a semisupervised learning method. We now extend it to the case of semisupervised learning.

Specifically, we first set $V_i^s = 1$, for $v_i \in V_L^s$, and $V_i^s = 0$, for $v_i \in V - V_L^s$.

Starting from $N(v_i)$ ($v_i \in V_L^s$), one consecutively updates the voltages of $v_i \in V_U$ to

$$V_i^s = \frac{1}{d_i} \sum_{v_j \in N(v_i)} V_j^s = \frac{1}{d_i} \sum_j V_j^s A_{ij}. \tag{15}$$

The updating process adopts breadth-first search algorithm and it will end when we get voltages for all vertices in V_U. This process is called a round. One spends an amount of $O(d_i)$ time calculating neighbor voltage of vertex v_i and $|V|$ time setting initial voltages; therefore the complexity in one round is $O(|V|+|E|)$. After repeating the updating process for a finite number of rounds, one will reach an approximate solution of (14) within a certain precision which only depends upon the number of iteration rounds.

Unlike Wu and Huberman's method [20], we do not need to compute the ideal voltage gap and know roughly the size of each community. As a result, we get a $(n-l)$-dimensional voltage vector. The component V_i^s reflects the relationship of vertex v_i and sth community. For each label s in label set $K = \{1, 2, \ldots, k\}$, we repeat this process. Therefore, for each vertex v_i, we obtain a voltage vector $(V_i^1, V_i^2, \ldots, V_i^k)$. The element V_i^j can be considered as the membership degree which vertex v_i belongs to the jth community. The vertex v_i is within the jth community if $V_i^j = \max V_i^s$, $1 \le s \le k$. That is to say, largest voltage of each vertex indicates to which community the vertex v_i should belong.

4. Experiments

To validate the proposed algorithm, one would like to test it on four real networks and three benchmarks which are widely used to test the validity of various community division methods. The experimental platform is based on Windows 7 Ultimate Service Pack 1 with Intel® Core™ i5-3470 CPU 3.20 GHz, 4.00 GB memory, ×64 Operating system, and Java 1.8 Eclipse RCP Luna sr1.

4.1. Three Evaluation Indices of Clustering. To assess the quality of partition, we here use the F-measure, P-measure, and modularity Q to quantify the cluster results. The F-measure is a harmonic combination of the precision and recall values used in information retrieval [21].

If n_i is the number of the members of class i, and n_{ij} is the number of the members of class i in cluster j, then the precision P_{ij} and recall R_{ij} can be defined as

$$\begin{aligned} P_{ij} &= \frac{n_{ij}}{n_j}, \\ R_{ij} &= \frac{n_{ij}}{n_i}. \end{aligned} \tag{16}$$

F_{ij} is denoted by

$$F_{ij} = \frac{2 \times P_{ij} \times R_{ij}}{P_{ij} + R_{ij}}. \tag{17}$$

The corresponding F-measure (FM) of the whole clustering result is defined as

$$\mathrm{FM} = \sum_i \frac{n_i}{N} \max_j F_{ij}, \tag{18}$$

where N is the total number of the members in the data set.

In general, the high value of F-measure indicates the better cluster result.

The purity of a cluster represents the fraction of the cluster corresponding to the largest class of data assigned to that cluster; thus the purity of cluster j is defined as

$$P_j = \frac{1}{n_j} \max_i n_{ij}. \tag{19}$$

The purity of the whole clustering result is defined as

$$\mathrm{PM} = \sum_j \frac{n_j}{N} P_j. \tag{20}$$

In general, the larger the purity value is, the better the clustering result is.

In order to quantify the validity of community division of a complex network and to optimize the chosen splitting, we use, following [22], the concept of modularity. It is defined as follows: given a network division, Let e_{ij} be the fraction of edges in the network that connect vertices in group i to those in group j, and let $a_i = \sum_j e_{ij}$. Then the modularity Q is defined as

$$Q = \sum_i \left(e_{ii} - a_i\right)^2. \tag{21}$$

It measures the fraction of edges that fall between communities minus the expected value of the same quantity in a random graph with the same community division. Obviously, the larger Q corresponds to the ideal community structure.

4.2. Experiment on Four Real Networks. Testing an algorithm essentially means analyzing a network with a well-defined community structure and recovering its communities. In this subsection, four classical complex networks with known community structures are selected to test the introduced algorithm. The description of these four networks can be found everywhere [1, 11, 16, 23]. Taking Zachary Karate Club network with two communities, for example, we first choose randomly one node in each community and label it. Afterwards, the algorithm can work on this network and a community division is detected. The values of FM, PM, and Q can be computed according to the obtained partition. It is possible that the community division may be changed with the different selection of initial labeled notes. To evaluate the validity of the proposal objectively, we calculate the average values of three indices by choosing randomly 10 groups of initial labeled notes. Along this way, we also compute three indices values by adding the number of labeled notes in each community. In Table 1, we list the average values of three indices by selecting randomly 10 groups of different labeled

Table 1: The average values of indices for four real networks.

Network	Index	Label number									
		1	2	3	4	5	6	7	8	9	10
Dolphins	FM	0.8473	0.9571	0.9783	0.9902	0.9919	0.9952	0.9968	0.9903	0.9968	0.9936
	PM	0.8484	0.9613	0.979	0.9903	0.9919	0.9952	0.9968	0.9903	0.9968	0.9936
	Q	0.2476	0.338	0.3634	0.3757	0.3743	0.3805	0.3773	0.3743	0.3788	0.3774
Football	FM	0.8503	0.9179	0.9561	0.9667	0.9845	0.9913	0.9914	0.9939	0.9957	1
	PM	0.867	0.9235	0.9574	0.9669	0.9843	0.9913	0.9913	0.9939	0.9957	1
	Q	0.5544	0.5738	0.5763	0.5744	0.5657	0.5628	0.5608	0.5612	0.5582	0.554
Zachary	FM	0.764	0.8112	0.9409	0.9369	0.9615	0.9764	0.9853	0.9853	0.9882	0.9882
	PM	0.7235	0.7941	0.9412	0.9382	0.9618	0.9765	0.9853	0.9853	0.9882	0.9882
	Q	0.1655	0.2162	0.352	0.3449	0.3574	0.3642	0.3686	0.3705	0.3691	0.3704
Polbooks	FM	0.7003	0.772	0.8541	0.8588	0.8805	0.8825	0.898	0.8994	0.9021	0.9249
	PM	0.7095	0.7762	0.8676	0.8629	0.88	0.8838	0.8934	0.8971	0.8943	0.921
	Q	0.2732	0.3703	0.4593	0.4704	0.4581	0.472	0.4688	0.4632	0.4772	0.4623

Table 2: The average run time of the proposed algorithm for four real networks.

Network	Label number									
	1	2	3	4	5	6	7	8	9	10
Dolphins	0.7015	0.2953	0.239	0.161	0.1156	0.0812	0.0812	0.0672	0.0562	0.0515
Football	1.4094	0.6734	0.4562	0.3048	0.228	0.1826	0.1343	0.1107	0.0843	0.064
Zachary	0.1174	0.0704	0.0455	0.0312	0.0283	0.0171	0.0172	0.0109	0.0095	0.0093
Polbooks	1.7469	0.8718	0.5392	0.3938	0.2783	0.2688	0.2202	0.189	0.1735	0.1408

notes and the label number (number of labeled nodes in each community) varies from 1 to 10.

From Table 1, it is easy to see that we can detect an ideal community division for these four networks by the proposed algorithm when we label 3 nodes in each community. The accuracy of network partition is greater than or equal to 94% except polbooks network. Three indices values are ascending or varying slightly with the increasing of labeled nodes. These results also show that one can detect a good network partition by labeling a small quantity of nodes in each community. For football network, we can get the same partition accuracy as in [15] while the number of the labeled vertices randomly selected is from 1 to 4.

In Table 2, the average run times of the proposed algorithm for four real networks are presented. It is shown that the run times decrease with the increase of labeled nodes. This is reasonable because the number of nodes that need to be divided is reduced.

Figures 1 and 2 show the variety of run time of the proposed algorithm and the values of three indices for dolphins network and karate network, respectively.

4.3. Experiment on Three Benchmarks. For testing community detection algorithms on graphs with overlapping communities, several artificial networks or benchmarks are introduced. Among them, the most famous benchmark for community detection is a class of networks introduced by Girvan and Newman (GN) [24]. Each network has 128 nodes, divided into four communities with 32 nodes. The average degree of the network is 16 and the nodes have approximately the same degree, as in a random graph.

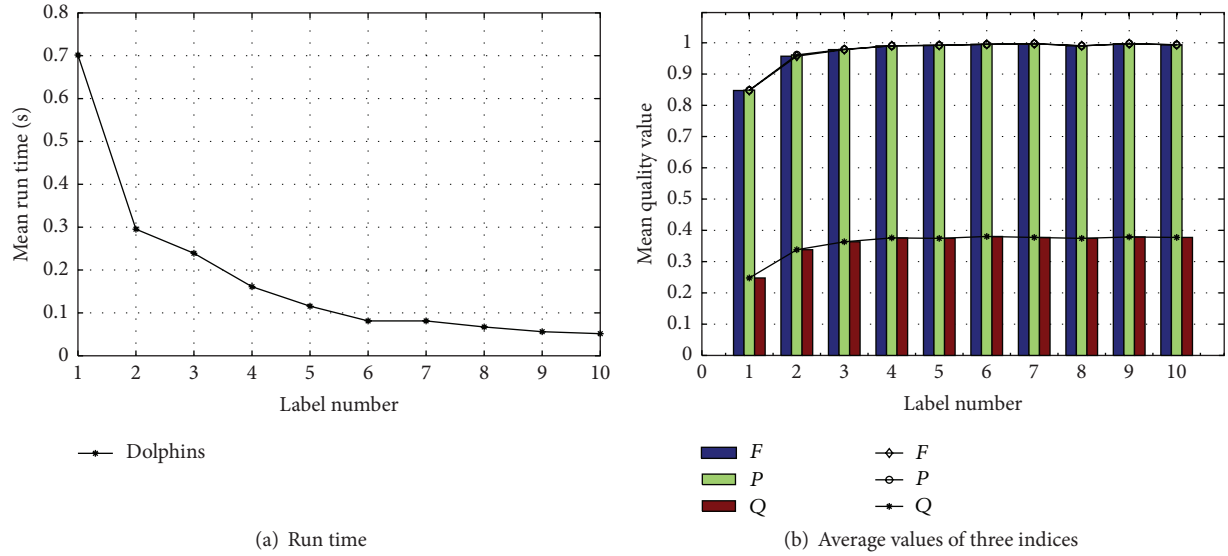
In what follows, we apply the proposed algorithm to detect the communities on this benchmark. For each fixed number of labeled nodes, one also selects randomly 10 groups of different initial labeled nodes to compute the average values of three indices. The benchmark can be thought of as the network with apparent community structure if mixing parameter $\mu < 0.5$. From Table 3, one can see that four communities in this benchmark are detected accurately when mixing parameter $\mu < 0.25$ and the number of labeled nodes is equal to or greater than 4. If we take $\mu = 0.5$ and label 10 nodes in each community, 90% of nodes in this benchmark can be partitioned correctly. When $\mu > 0.5$, this benchmark is with overlapping community structures. Although the partition accuracy becomes higher and higher with the increasing of number of labeled nodes, we can not find ideal communities in this network. Particularly, our algorithm fails to divide it into four groups when $\mu = 1$ and number of labeled nodes in each groups is less than 10.

Assuming that both the degree and the community size distributions are power laws, Lancichinetti et al. [25] designed a more general benchmark for testing community detection algorithms on graphs. Some parameters used in this benchmark are explained as follows:

N: number of nodes,

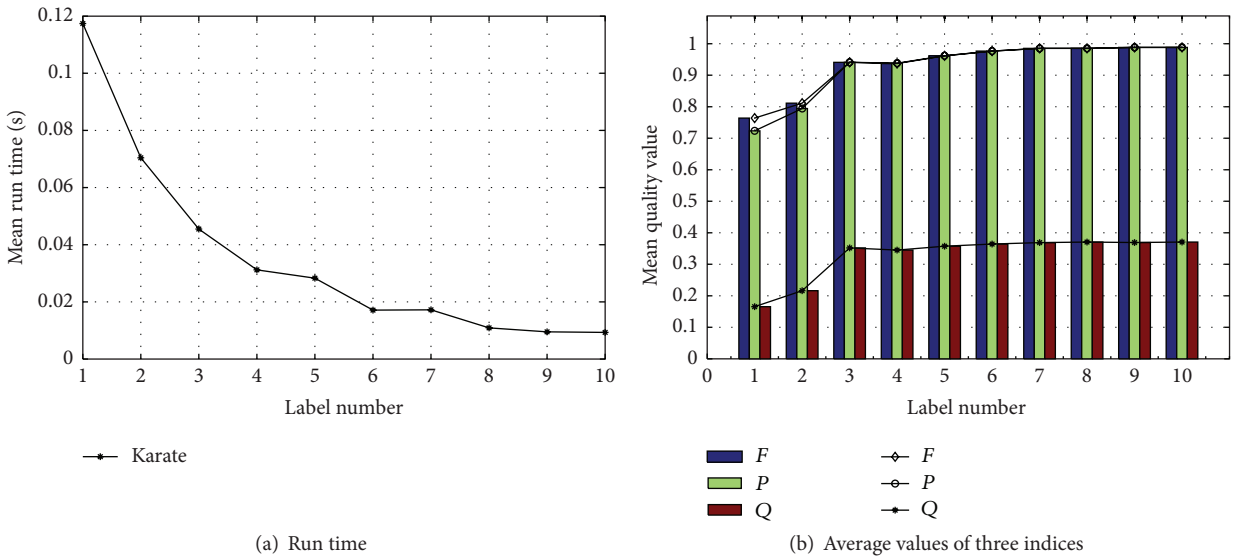
k: average degree,

max k: maximum degree,



(a) Run time (b) Average values of three indices

Figure 1: Dolphins network.



(a) Run time (b) Average values of three indices

Figure 2: Zachary network.

μ: mixing parameter,

t_1: minus exponent for the degree sequence,

t_2: minus exponent for the community size distribution,

$\min c$: minimum for the community sizes,

$\max c$: maximum for the community sizes,

on: number of overlapping nodes,

om: number of memberships of the overlapping nodes,

C: [average clustering coefficient] not mandatory.

In this benchmark, N, k, $\max k$, and μ have to be specified. For the others, the program can use default values: $t_1 = 2$; $t_2 = 1$; on = 0; om = 0; $\min c$ and $\max c$ will be chosen close to the degree sequence extremes.

If we set parameters $N = 128$, $k = 16$, $\max k = 16$, $\min c = 32$, and $\max c = 32$, a kind of Girvan-Newman benchmark will be obtained.

To test the validity of our algorithm on large network, we apply the proposed algorithm to this benchmark with parameters $N = 10^5$, $k = 20$, $\max k = 10^4$, and $t_2 = 1$. The mixing parameter μ is varied from 0.1 to 0.6. For each fixed μ, one takes $t_1 = 2$ and $t_1 = 3$, respectively. Unlike the GN benchmark, the community size is power laws in this network. Therefore, it is proper to label nodes in each community in terms of node proportion. The minimal proportion which we will take is 10% because of the requirement that there

Table 3: The average values of indices on GN benchmark.

μ	Index	Label number									
		1	2	3	4	5	6	7	8	9	10
0.0625	FM	1	1	1	1	1	1	1	1	1	1
	PM	1	1	1	1	1	1	1	1	1	1
	Q	0.6875	0.6875	0.6875	0.6875	0.6875	0.6875	0.6875	0.6875	0.6875	0.6875
0.1250	FM	1	1	1	1	1	1	1	1	1	1
	PM	1	1	1	1	1	1	1	1	1	1
	Q	0.625	0.625	0.625	0.625	0.625	0.625	0.625	0.625	0.625	0.625
0.1875	FM	0.9733	1	1	1	1	1	1	1	1	1
	PM	0.9735	1	1	1	1	1	1	1	1	1
	Q	0.5262	0.5625	0.5625	0.5625	0.5625	0.5625	0.5625	0.5625	0.5625	0.5625
0.2500	FM	0.9264	0.9734	0.9977	1	1	1	1	1	1	1
	PM	0.9266	0.9734	0.9977	1	1	1	1	1	1	1
	Q	0.4135	0.4686	0.4973	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.3125	FM	0.8592	0.8994	0.941	0.9789	0.993	0.9984	1	0.9992	0.9992	0.9992
	PM	0.8601	0.9	0.9414	0.9789	0.993	0.9984	1	0.9992	0.9992	0.9992
	Q	0.3112	0.3417	0.3819	0.4163	0.4311	0.4359	0.4375	0.4369	0.4368	0.4368
0.3750	FM	0.7277	0.8017	0.8659	0.8973	0.9514	0.9679	0.9796	0.9899	0.9946	0.9953
	PM	0.7336	0.8031	0.8664	0.8976	0.9516	0.968	0.9797	0.9899	0.9945	0.9953
	Q	0.2146	0.2411	0.2735	0.2965	0.3373	0.3498	0.3598	0.3668	0.3704	0.372
0.4375	FM	0.5788	0.6648	0.761	0.8031	0.8475	0.8907	0.913	0.9388	0.9501	0.9654
	PM	0.5922	0.6695	0.7617	0.8039	0.8492	0.8914	0.9133	0.9391	0.95	0.9656
	Q	0.1495	0.1614	0.1882	0.2119	0.2271	0.2495	0.2648	0.2757	0.2822	0.2923
0.5000	FM	0.4698	0.5567	0.6344	0.6806	0.7386	0.7615	0.8168	0.8546	0.8771	0.9054
	PM	0.4781	0.5695	0.6383	0.682	0.7391	0.7625	0.818	0.8554	0.8773	0.9062
	Q	0.1173	0.1265	0.1298	0.1456	0.1614	0.1701	0.1822	0.1931	0.2021	0.2135
0.5625	FM	0.41	0.4697	0.5389	0.5613	0.6142	0.6576	0.6784	0.7318	0.7464	0.7826
	PM	0.4086	0.4766	0.5406	0.5641	0.618	0.6594	0.682	0.7344	0.7469	0.7828
	Q	0.1142	0.1021	0.1133	0.1128	0.1194	0.1278	0.136	0.1401	0.1377	0.1509
0.6250	FM	0.3528	0.3844	0.434	0.4679	0.4989	0.5401	0.5776	0.5963	0.636	0.6725
	PM	0.3469	0.3891	0.4352	0.4719	0.5031	0.5422	0.5805	0.5984	0.6375	0.6726
	Q	0.125	0.099	0.1047	0.1099	0.1123	0.1075	0.1105	0.1144	0.1186	0.118
0.6875	FM	0.3559	0.3519	0.3789	0.4089	0.4179	0.462	0.4791	0.4971	0.5191	0.5641
	PM	0.3352	0.3406	0.3812	0.4078	0.4211	0.4656	0.4828	0.4984	0.5195	0.5656
	Q	0.1068	0.1042	0.1076	0.108	0.1116	0.1131	0.1183	0.1209	0.1142	0.1172
0.7500	FM	0.3547	0.3421	0.3459	0.3408	0.3613	0.3808	0.3933	0.4233	0.4498	0.4758
	PM	0.3211	0.3234	0.3406	0.3445	0.3633	0.3805	0.393	0.4227	0.4508	0.475
	Q	0.1012	0.1033	0.1064	0.1107	0.1206	0.1168	0.1148	0.1123	0.1106	0.1136
0.8125	FM	0.3297	0.3361	0.3107	0.323	0.3209	0.3415	0.3522	0.3618	0.3959	0.4166
	PM	0.3203	0.3148	0.3047	0.3141	0.3164	0.336	0.3484	0.3609	0.3961	0.418
	Q	0.1219	0.1035	0.1109	0.1065	0.1043	0.109	0.121	0.1119	0.1082	0.1042
0.8750	FM	0.3461	0.3324	0.3307	0.3255	0.3166	0.3255	0.3176	0.337	0.3417	0.3657
	PM	0.3156	0.3172	0.318	0.3187	0.3109	0.3172	0.3141	0.3352	0.3406	0.3649
	Q	0.1028	0.1044	0.108	0.1081	0.1151	0.1188	0.1158	0.1145	0.1055	0.1041
0.9375	FM	0.3314	0.3252	0.3442	0.3266	0.327	0.3187	0.3125	0.3107	0.3199	0.337
	PM	0.307	0.3133	0.3289	0.318	0.3195	0.3156	0.3102	0.3125	0.3211	0.3391
	Q	0.099	0.1073	0.1027	0.097	0.1086	0.1052	0.1029	0.1037	0.0981	0.1048
1.0000	FM	0.3286	0.3312	0.3455	0.3359	0.3318	0.3237	0.3136	0.3096	0.306	0.3257
	PM	0.3109	0.3156	0.3321	0.3297	0.325	0.318	0.3102	0.3062	0.3078	0.3242
	Q	0.1139	0.1018	0.0981	0.1067	0.102	0.1076	0.096	0.1002	0.1034	0.0967

Table 4: The mean values of indices on power law benchmarks.

Network	Index	Label proportion (%)				
		10	20	30	40	50
pw-0.1-2-1	FM	0.9585	0.9627	0.9707	0.9739	0.9795
	PM	0.958	0.9622	0.9703	0.9736	0.9792
	Q	0.8152	0.8155	0.8204	0.8233	0.8256
pw-0.1-3-1	FM	0.9602	0.9658	0.9709	0.9755	0.98
	PM	0.9524	0.9594	0.9657	0.9718	0.9775
	Q	0.789	0.7987	0.807	0.8148	0.8231
pw-0.2-2-1	FM	0.8968	0.9247	0.9332	0.9514	0.9579
	PM	0.8956	0.9225	0.9318	0.9499	0.9568
	Q	0.6756	0.6861	0.6922	0.701	0.7016
pw-0.2-3-1	FM	0.8923	0.9125	0.9241	0.9362	0.9465
	PM	0.8711	0.8953	0.9097	0.9252	0.9383
	Q	0.6079	0.6323	0.6478	0.6635	0.6772
pw-0.3-2-1	FM	0.8102	0.8582	0.8841	0.9066	0.9264
	PM	0.8125	0.8537	0.8798	0.9025	0.9231
	Q	0.5312	0.5539	0.5602	0.5739	0.5828
pw-0.3-3-1	FM	0.7064	0.7521	0.7933	0.8245	0.8574
	PM	0.6776	0.7228	0.7651	0.7995	0.8369
	Q	0.3901	0.4159	0.4466	0.4707	0.4988
pw-0.4-2-1	FM	0.7013	0.7532	0.81	0.8386	0.8712
	PM	0.7083	0.7537	0.8059	0.8354	0.8675
	Q	0.4218	0.4383	0.4562	0.4719	0.4814
pw-0.4-3-1	FM	0.7405	0.8253	0.8679	0.8994	0.9226
	PM	0.7125	0.7957	0.8401	0.8747	0.9025
	Q	0.351	0.4014	0.4291	0.4512	0.4698
pw-0.5-2-1	FM	0.5425	0.6709	0.7346	0.7663	0.8152
	PM	0.5603	0.671	0.7335	0.764	0.8119
	Q	0.2966	0.3432	0.3619	0.3702	0.385
pw-0.5-3-1	FM	0.5021	0.6245	0.6955	0.7542	0.8059
	PM	0.4752	0.5846	0.6519	0.7112	0.7689
	Q	0.2017	0.2365	0.2601	0.2851	0.3123
pw-0.6-2-1	FM	0.4258	0.5589	0.6464	0.7004	0.7551
	PM	0.4394	0.5607	0.6423	0.6964	0.7496
	Q	0.1988	0.2389	0.2642	0.2802	0.2882
pw-0.6-3-1	FM	0.3891	0.5211	0.612	0.6895	0.7537
	PM	0.3665	0.4811	0.5632	0.6383	0.7084
	Q	0.1494	0.1739	0.191	0.2112	0.2309

exists one labeled node at least in each community and the fact that there are small size communities in this network. Applying the proposed algorithm on this benchmark by labeling randomly of two groups of different initial nodes, one obtains some results reported in Table 4. There are nearly 90% of nodes which can be classified correctly in this network while $\mu < 0.2$ and 10% nodes in each community are labeled. In this case, there is no distinct variety of three indices values with the increasing of label proportion. This fact indicates that one can detect a good community division on the network with apparent community structure although a few nodes are labeled. The values in each column are descending with the increasing of mixing parameter μ. This shows that a good network partition will not be found by the proposed algorithm for the network which communities overlap seriously.

Figure 3 presents the comparison of run time of our algorithm on two benchmarks with different parameters and label nodes numbers or label proportions. The increasing of labeled nodes number or label proportions implies that the number of unlabeled nodes in benchmarks is descending, and therefore it needs less and less time to partition network into groups.

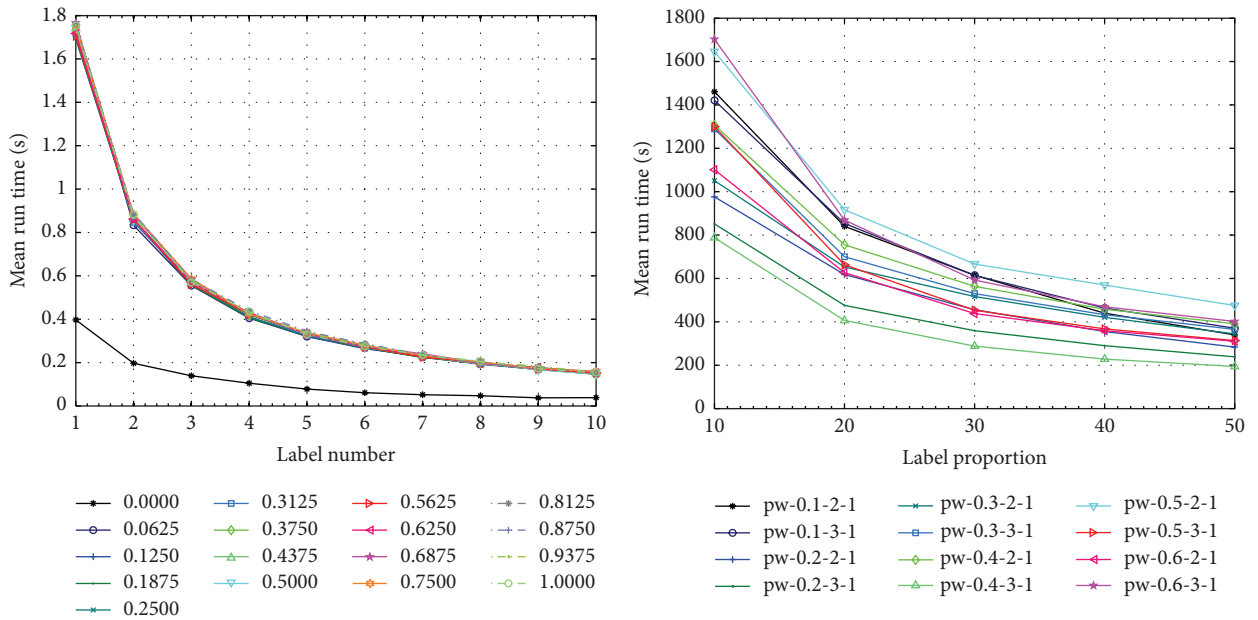
We now present our experimental results on the LFR benchmark and further compare our proposal with GN algorithm [24], spectral clustering algorithm [1], NMF algorithm [20], and SNMF-SS algorithm [11] by a normalized mutual information index (NMI).

The LFR benchmark is designed by Lancichinetti et al. [25] and widely employed to test the performance of community structure identification. It allows user to specify distributions for both the community sizes and the degree distribution and then generates vertices and communities by sampling from those distributions. The mix parameter μ represents the average ratio of intracommunity adjacencies to total adjacencies. The large μ corresponds to the network with apparent community structure. In this paper, the input parameters of the LFR benchmark are the same for our algorithm and the comparative algorithms. For the different values of $\mu \in \{0.50, 0.6, 0.7, 0.8, 0.9\}$, we generated 50 instances for each of LFR benchmark graphs whose node degree is taken from a power law distribution with exponent 2 and community size from a power law distribution with exponent 1. Each graph has 1000 vertices, average degree of 15, maximum degree of 50, maximum for the community sizes of 50, and minimum for the community sizes of 5. The definition of NMI can be found everywhere [11, 15, 26].

From Figure 4, we can see that the values of NMI obtained by our algorithm are bigger than those gotten by the other four algorithms. The peak value of our approach is 0.732 at $\mu_{\text{avg}} = 0.9$. This value is bigger than the one 0.7 computed by SNMF-SS algorithm. Because the decrease of μ means that the LFR benchmark is with the obscure community structure, it is difficult to detect communities correctly for five algorithms. It is reasonable that the NMI values obtained by five algorithms become smaller and smaller as μ decreases. The NMF algorithm seems to be stable since it has a small decrease speed. The performance of our proposal decreases greatly while μ is greater than 0.6. This fact implies that our algorithm can not apply the networks with nonapparent community structure. However, compared with other four algorithms, our algorithm can gain the best performance.

5. Conclusions

In this paper, we propose a semisupervised community detection algorithm for partitioning network into groups. This approach amalgamates the discrete potential theory and Wu-Huberman algorithm. The complexity $O(|V| + |E|)$ of the introduced approach indicates that it can be applied to detect



(a) GN benchmark

(b) Power law benchmark

Figure 3: The comparison of run time of our algorithm on two benchmarks.

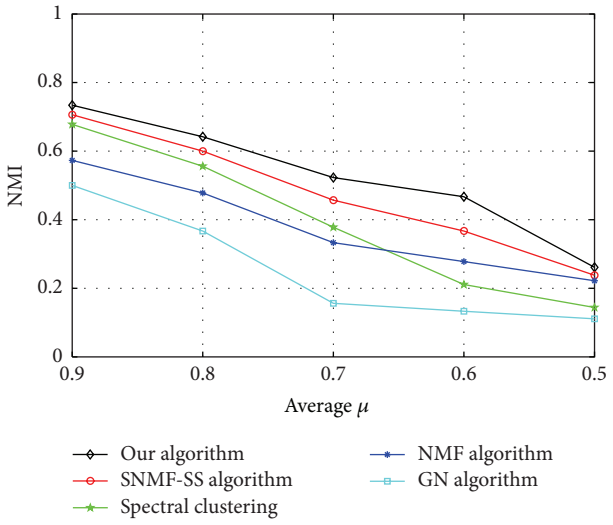


Figure 4: The comparative results of five algorithms on the LFR benchmark.

community on large network. The validity of our proposal is demonstrated by applying it to four real networks and three benchmarks. The experimental results show that a good community division of a complex network is obtained by labeling a small quantity of nodes in each community. However, it is difficult to classify correctly the network with heavily overlapping communities or obscure community structure by our method. This fact can be seen from the experimental result on LFR benchmark. Therefore, it is worthwhile to further introduce new and fast algorithm to deal with this case.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This work is supported by NSFC (under Grants nos. 61373127 and 41471140).

References

[1] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, 2010.

[2] L. Chen, Q. Yu, and B. Chen, "Anti-modularity and anti-community detecting in complex networks," *Information Sciences*, vol. 275, pp. 293–313, 2014.

[3] K. Li and Y. Pang, "A unified community detection algorithm in complex network," *Neurocomputing*, vol. 130, pp. 36–43, 2014.

[4] L. Zhang, J. Cao, and J. Li, "Complex networks: Statistical properties, community structure, and evolution," *Mathematical Problems in Engineering*, vol. 2015, Article ID 590794, 7 pages, 2015.

[5] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[6] W. Li, "A constrained power method for community detection in complex networks," *Mathematical Problems in Engineering*, vol. 2014, Article ID 804381, 6 pages, 2014.

[7] Z. Chen, Z. Xie, and Q. Zhang, "Community detection based on local topological information and its application in power grid," *Neurocomputing*, vol. 170, pp. 384–392, 2015.

[8] L. Zhou, K. Lü, P. Yang, L. Wang, and B. Kong, "An approach for overlapping and hierarchical community detection in social networks based on coalition formation game theory," *Expert Systems with Applications*, vol. 42, no. 24, pp. 9634–9646, 2014.

[9] K. Liu, J. Huang, H. Sun, M. Wan, Y. Qi, and H. Li, "Label propagation based evolutionary clustering for detecting overlapping and non-overlapping communities in dynamic networks," *Knowledge-Based Systems*, vol. 89, pp. 487–496, 2015.

[10] L. Ma, M. Gong, J. Liu, Q. Cai, and L. Jiao, "Multi-level learning based memetic algorithm for community detection," *Applied Soft Computing*, vol. 19, pp. 121–133, 2014.

[11] X. Ma, L. Gao, X. Yong, and L. Fu, "Semi-supervised clustering algorithm for community structure detection in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 1, pp. 187–197, 2010.

[12] T. C. Silva and L. Zhao, "Semi-supervised learning guided by the modularity measure in complex networks," *Neurocomputing*, vol. 78, no. 1, pp. 30–37, 2012.

[13] Z.-Y. Zhang, "Community structure detection in complex networks with partial background information," *Europhysics Letters*, vol. 101, no. 4, Article ID 48005, 2013.

[14] Z.-Y. Zhang, K.-D. Sun, and S.-Q. Wang, "Enhanced community structure detection in complex networks with partial background information," *Scientific Reports*, vol. 3, no. 11, p. 48005, 2013.

[15] D. Liu, X. Liu, W. Wang, and H. Bai, "Semi-supervised community detection based on discrete potential theory," *Physica A: Statistical Mechanics and its Applications*, vol. 416, pp. 173–182, 2014.

[16] F. Wu and B. A. Huberman, "Finding communities in linear time: a physics approach," *European Physical Journal B*, vol. 38, no. 2, pp. 331–338, 2004.

[17] Q.-M. Zhang, L. Lü, W.-Q. Wang, Y.-X. Zhu, and T. Zhou, "Potential theory for directed networks," *PLoS ONE*, vol. 8, no. 2, Article ID e55437, 2013.

[18] F. Wang and C. Zhang, "Semisupervised learning based on generalized point charge models," *IEEE Transactions on Neural Networks*, vol. 19, no. 7, pp. 1307–1311, 2008.

[19] L. Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.

[20] S. Zhang, R.-S. Wang, and X.-S. Zhang, "Identification of overlapping community structure in complex networks using fuzzy c-means clustering," *Physica A: Statistical Mechanics and its Applications*, vol. 374, no. 1, pp. 483–490, 2007.

[21] C. Luo, Y. Li, and S. M. Chung, "Text document clustering based on neighbors," *Data & Knowledge Engineering*, vol. 68, no. 11, pp. 1271–1288, 2009.

[22] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, Article ID 026113, 2004.

[23] D. W. Zhang, F. D. Xie, Y. Zhang, F. Y. Dong, and K. Hirota, "Fuzzy analysis of community detection in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 22, pp. 5319–5327, 2010.

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

[25] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, no. 4, Article ID 046110, 2008.

[26] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics*, vol. 406, Article ID P09008, 2005.