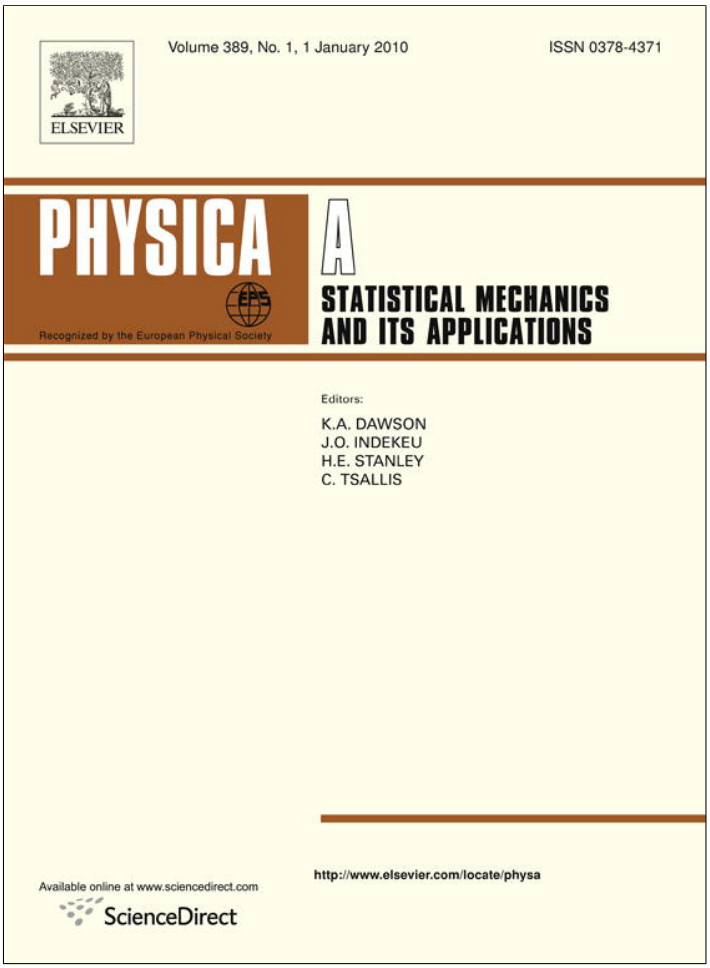

Physica A 389 (2010) 187–197

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Semi-supervised clustering algorithm for community structure detection in complex networks

Xiaoke Ma [a,*], Lin Gao [a], Xuerong Yong [b], Lidong Fu [a]

[a] School of Computer Science and Technology, Xidian University, 710071, PR China

[b] Department of Mathematical Sciences, University of Puerto Rico at Mayaguez, P.O. Box 9018, PR 00681, USA

ARTICLE INFO

Article history:
Received 4 June 2009
Received in revised form 31 August 2009
Available online 12 September 2009

Keywords:
Semi-supervised clustering
Complex networks
Community structure
Nonnegative matrix factorization

ABSTRACT

Discovering a community structure is fundamental for uncovering the links between structure and function in complex networks. In this paper, we discuss an equivalence of the objective functions of the symmetric nonnegative matrix factorization (SNMF) and the maximum optimization of modularity density. Based on this equivalence, we develop a new algorithm, named the so-called SNMF-SS, by combining SNMF and a semi-supervised clustering approach. Previous NMF-based algorithms often suffer from the restriction of measuring network topology from only one perspective, but our algorithm uses a semi-supervised mechanism to get rid of the restriction. The algorithm is illustrated and compared with spectral clustering and NMF by using artificial examples and other classic real world networks. Experimental results show the significance of the proposed approach, particularly, in the cases when community structure is obscure.

1. Introduction

Various complex networks, such as social networks [1], technological networks [2], and biological networks [3], can be effectively modeled as graphs by regarding each entity as a vertex and each link as an edge. It has been shown that many real world networks have a structure of modules or communities which are characterized by groups of densely connected nodes. Generally, a community is a subgraph whose nodes are more tightly connected with each other than with nodes outside the subgraph. However, the intuitive meaning of a community differs greatly in different networks. This is, for example, confirmed in the case of the social networks, where communities correspond to social groups with similar interest or background. In protein–protein interaction networks, it is widely believed that the modular structure can be a functional unit.

Identifying a community structure in special networks has a considerable merit of practice because it gives us insights into the structure-functionality relationship. Over the years, researchers have introduced a large number of algorithms, for example, the spectral clustering algorithm [4,5], betweenness-based method [1], NMF-based algorithm [6,7], fuzzy clustering approach [8] etc. More algorithms for community detection can be referred to Refs. [9,10]. Among these methods, the NMF-based algorithm is a recently introduced method for multivariate data [11,12]. It has received considerable attention in several disciplines immediately, particularly in the complex network. Literatures [6,7] employed the NMF-based algorithm for the community and both of them attained good performance. However, designing an efficient algorithm for community is still highly nontrivial largely due to the following three reasons.

1. Measure for community: 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. [13] introduced a modularity function Q which measures

* Corresponding address: P.O. Box 171, School of Computer Science and Technology, Xidian University, No.2 South TaiBai Road, Xi'an, Shaanxi, 710071, PR China.
E-mail address: maxiaoke8218@163.com (X. Ma).

0378-4371/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.physa.2009.09.018

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 Refs. [5,14,15], although finding the optimal Q value is an NP-hard problem [14,16]. Recently, Fortunato et al. [17] 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. To overcome the drawbacks of Q function, a new measure, named by modularity density or D, was presented in Ref. [18].

2. Structure of community: communities may be in complicated shapes. Palla et al. [19] revealed that complex network models exhibit an overlapping community structure, also called fuzzy community. Furthermore, Ravasz et al. [20] proved the existence of the hierarchical organization of modularity in metabolic networks. These overlapping and hierarchical communities are more realistic than average ones. For example, a person in social networks belongs to more than one community at the same time. However, these complicated structures actually make it harder to contrive appropriately algorithms for overlapping or hierarchical communities. Now, only a few efficient algorithms [8–10,21] can uncover such realistic structures.
3. Topology structure of complex networks: measuring the graph topological characteristics of complex network is difficult, although there are various similarity measures presented [22–26]. Usually, topological characteristics cannot be captured by one or two measure indexs. In Ref. [6], several similarity measures, such as diffusion kernel similarity, shortest path based similarity, on several well studied networks were evaluated. It is also indicated that different networks require different similarity or dissimilarity measures. Thus, how to make the most use of appropriate measures to grasp as much network structure information as possible is also crucial to community detection.

At present, most algorithms for community detection focus on the first and the second challenges, but few methods concentrate on the third one. Based on the discussion above, we will focus on the identification of the community in complex networks from the third aspect. In this paper, we will prove the equivalence of objective functions of symmetric NMF (SNMF) and the maximum optimization of modularity density. Such an equivalence has an immediate implication: we may use the NMF to optimize the modularity density D. Based on the equivalence, we propose a SNMF-based semi-supervised clustering (SNMF-SS) algorithm by a combination of SNMF and a semi-supervised clustering way. Particularly, our algorithm can simultaneously make use of several similarity measures presented in Refs. [22–26], and grasp topological properties from different perspectives. However, others can only use one similarity measure at one time. By applying the SNMF-SS algorithm to several widely studied networks and artificial synthesis networks, we can find that the algorithm has better performance in terms of classification accuracy, robustness and ability to large-scale networks. Comparison with other algorithms also indicates the superiority of SNMF-SS, particularly, in such graphs whose community structure is obscure.

The paper is organized as follows. In Section 2, we show the equivalence of the optimization of modularity density and the SNMF. The proposed SNMF-SS algorithm for community discovery is proposed in Section 3. Experimental results appear in Section 4. Finally, a conclusion is presented in Section 5.

2. Equivalence of modularity density and symmetric nonnegative matrix factorization

Let us begin with the definition of the modularity density function D. In detail, given an undirected network $G = (V, E)$ consisting of the vertex set $V = \{v_1, \ldots, v_n\}$ (n is the cardinality of V) and the edge set E, the modularity density function D is defined as [18]

$$D = \sum_{c=1}^{m} \frac{L(V_c, V_c) - L(V_c, \bar{V}_c)}{|V_c|}, \tag{2.1}$$

where $\{V_c\}_{c=1}^m$ is a partition of the V into m groups, $\bar{V}_c$ is the complement of V_c with respect to V, $L(V_c, \bar{V}_c) = \sum_{i\in V_c, j\in\bar{V}_c} A_{ij}$ ($A = (A_{ij})$ is the adjacent matrix of G) and $|V_c|$ is the cardinality of V_c.

The objective function of kernel k-means is to construct a m-way partition $\{V_c\}_{c=1}^m$ with the purpose to minimize the sum of squared errors [27]

$$F = \sum_{c=1}^{m} \sum_{v_i\in V_c} \|\phi(v_i) - m_c\|^2, \qquad m_c = \sum_{v_i\in V_c} \phi(v_i)/|V_c| \tag{2.2}$$

where $\|\cdot\|^2$ is the Frobeneus norm, m_c is the centre of the c-th cluster and ϕ is a kernel function mapping the vectors onto a generally higher dimensional space. The squared distance $\|\phi(v_i) - m_c\|^2$ can be rewritten as

$$\phi(v_i)\cdot\phi(v_i) - \frac{2\sum_{v_j\in V_c} \phi(v_i)\cdot\phi(v_j)}{|V_c|} - \frac{\sum_{v_j, v_l\in V_c} \phi(v_j)\cdot\phi(v_l)}{|V_c|^2}. \tag{2.3}$$

As a result, given a kernel matrix K with entry $K_{ij} = \phi(v_i)\cdot\phi(v_j)$, the above may be rewritten as

$$K_{ii} - \frac{2\sum_{v_j\in V_c} K_{ij}}{|V_c|} + \frac{\sum_{v_j, v_l\in V_c} K_{jl}}{|V_c|^2}. \tag{2.4}$$

In Ref. [18], the authors proved the equivalence of the objective functions of maximum optimization D and kernel k-means

$$\min F \propto \max D. \tag{2.5}$$

Now, we show the equivalence of D and SNMF by proving Eq. (2.2) can be solved approximately by the SNMF, similar to Ref. [28].

Setting Eq. (2.2) as a trace minimization, we can get

$$\min_{H^{\mathrm{T}}H=I, H\geq 0} F = Trace(K) - Trace(H^{\mathrm{T}}KH), \tag{2.6}$$

where $Trace(K)$ is the trace of matrix K, I is an identity matrix and

$$H = (h_1, h_2, \ldots, h_m) \tag{2.7}$$

is the solution to the clustering, consisting of m nonnegative indicator vectors

$$h_k = (0, \ldots, 0, \underbrace{1, 1, \ldots, 1}_{|V_k|}, 0, \ldots, 0)^{\mathrm{T}}/|V_k|^{1/2}, \quad \forall k = 1, 2, \ldots, m. \tag{2.8}$$

It is easy to validate that $h_k^{\mathrm{T}}h_l = 0$ when $k \neq l$, and 1 otherwise. Because the first term in Eq. (2.6) is a constant, we can conclude that

$$\min F \propto \max Trace(H^{\mathrm{T}}KH). \tag{2.9}$$

Now, what remains to be done is to show that $\max Trace(H^{\mathrm{T}}KH)$ can be solved by the SNMF. In detail, provided that the feature matrix K and the number of clusters m are given, the SNMF approximately factorizes the matrix K into a $n \times m$ matrix and its transpose matrix such that

$$K \approx HH^{\mathrm{T}}, \quad H \geq 0, \tag{2.10}$$

where K and H are nonnegative matrices. m is usually much smaller than n so that H is smaller than the original matrix K.

Actually, casting Eq. (2.10) as an optimization form, the objective function can be approximated by minimizing the sum of squared errors, defined as

$$\min_{H\geq 0} \|K - HH^{\mathrm{T}}\|^2. \tag{2.11}$$

Further, we can rewrite the right term of Eq. (2.9) as

$$\begin{aligned} \max Trace(H^{\mathrm{T}}KH) &\propto -\min_{H\geq 0} 2Trace(H^{\mathrm{T}}KH) \\ &\propto \min_{H^{\mathrm{T}}H=I, H\geq 0} \|K\|^2 - 2Trace(H^{\mathrm{T}}KH) + \|H^{\mathrm{T}}H\|^2 \\ &= \min_{H^{\mathrm{T}}H=I, H\geq 0} \|K - HH^{\mathrm{T}}\|^2. \end{aligned} \tag{2.12}$$

Relaxing the orthogonality $H^{\mathrm{T}}H = I$ (H is no longer the indicator matrix in Eq. (2.8) from now on), we can assert that SNMF is equivalent to kernel k means clustering by Eqs. (2.9)–(2.11), i.e.,

$$\max D \propto \min_{H\geq 0} \|K - HH^{\mathrm{T}}\|^2. \tag{2.13}$$

This implies that D attains its maximum if and only if the minimum of F is achieved, and F's minimum is reached only if $\min_{H\geq 0} \|K - HH^{\mathrm{T}}\|^2$ is obtained. Such a relation assures that algorithms based on SNMF can be applied to community discovery. Thus, we shift our focus to an efficient SNMF-SS algorithm for the community.

3. Method

In this section, the SNMF-SS algorithm, the parameters optimization strategies and complexity analysis are presented.

3.1. SNMF-SS algorithm

Based on the theoretic foundation presented in Section 2, we propose a SNMF-SS for the community detection problem. In SNMF-SS, domain knowledge is incorporated to guide the clustering. Supervision is provided as two sets of pairwise constrains on the data objects: *must-link* constraints C_{ML} and *cannot-link* constraints C_{CL}. Every pair $(i, j) \in C_{\mathrm{ML}}$ implies that object i and j must belong to the same cluster. Similarly, pair $(i, j) \in C_{\mathrm{CL}}$ indicates that the two different clusters. The constraints are accompanied by associated violation cost matrix W. An entry $w_{ij} \in W$ represents the cost of violating the constraints between object i and j.

According to Eq. (2.13), the objective function of the SNMF-SS algorithm is constructed as

$$J_{\mathrm{SNMF-SS}} = \min_{\bar{K}\geq 0, H\geq 0} \|\bar{K} - HH^{\mathrm{T}}\|^2, \tag{3.1}$$

where $\bar{K} = K - \alpha W_{\mathrm{ML}} + \beta W_{\mathrm{CL}}$ is the affinity or similarity matrix K with constraints $W_{\mathrm{ML}} = \{w_{ij} : (i,j) \in C_{\mathrm{ML}}\}$, $W_{\mathrm{CL}} = \{w_{ij} : (i,j) \in C_{\mathrm{CL}}\}$, and α, β are real numbers small enough to ensure $\bar{K}$ is positively definite (In our experiment, we find that $\alpha \in [0, 0.001]$ and $\beta \in [0.03, 0.3]$ are a good choice). Note that when $\alpha = \beta = 0$ SNMF-SS reduces to the NMF algorithm.

A common feature of the algorithms for solving Eq. (3.1) is iteratively updating matrix H to improve the approximation of HH^{T} to $\bar{K}$ while maintaining the non-negativeness of matrix entries throughout. SNMF-SS algorithm starts with random matrix H whose entries $H_{ij} \sim N(0, 1)$, where $N(0, 1)$ is a Gauss distribution. If an entry of the matrix is negative, its absolute value is adopted. The update rule for entries H_{ij} of H can be formulated as

$$H_{ij} = H_{ij} \left(\frac{(\bar{K}H)_{ij}}{(HH^{\mathrm{T}}H)_{ij}} \right). \tag{3.2}$$

Under such a rule, $J_{\text{SNMF-SS}}$ is non-increasing. Iteration continues until $J_{\text{SNMF-SS}}$ is lower than a predefined tolerant error, for example 10^{-2}, or the maximum iteration number, say 10^3, is reached. From the SNMF $\bar{K} \approx HH^{\mathrm{T}}$, a crisp partition $H = (H_{ij})$ can be constructed in the following manner: for each node i, $H_{ij^*} = 1$ for $l^* = \arg\max_{1 \le l \le m} H_{il}$, and 0 otherwise.

3.2. Parameters optimization

We explain the key techniques one should consider when choosing the appropriate values for parameters to enhance the performance of SNMF-SS.

• *Choosing the number of communities*

Like most other clustering algorithms, the first and most important parameter of the SNMF-SS approach is the number of communities m. Generally, determining m in a self-adaptive way without human intervention is a hard problem. There are various ways for m. Each algorithm has its own strong and weak points. The objective function driven method is a widely used alternative. It works as: First, a measure, say Q function or D function, is employed to determine an appropriate m. Second, starts with $m = 2$, then increasing m until the measure index reaches its peak. The value corresponding to the optimal of target function is selected as the best choice. One of the major advantages of such strategy is that it casts this problem in an constrained optimization framework where optimization theory is available. Its performance, however, is very sensitive to the employed measure. Unfortunately, as mentioned in introduction we have not reached a consensus on measure index for community. Spectral algorithms determine m via making use of the largest eigenvalue of the adjacency matrix A of G or the smallest eigenvalue of the Laplacian matrix $L = D - A$ ($D = diag(d_1, \ldots, d_n)$ with d_i is the degree of the i-th vertex). Spectral methods achieve a good performance when the eigenspectrum is a step function instead of a straight line.

Here, we propose a different divisive approach, which spares some computation in the early stage of algorithm. Initially, a m_l-way partition of a graph is computed, where $m_l \ge 2$ is derived from the given dissimilarity matrix. Then keep on increasing m_l until the new partitioning cannot improve the D function. Given a dissimilarity matrix $S = (S_{ij})$, m_l is constructed as: we first obtain the maximum dissimilarity degree in S, i.e., $S_{\max} = \max_{S_{ij} \in S}$. Based on the maximum dissimilarity degree, a set is builded within which each pairwise data has maximum dissimilarity, i.e., $S_{S_{\max}} \subseteq V, \forall i, j \in S_{S_{\max}} S_{i,j} = S_{\max}$. Finally, $m_l = |S_{S_{\max}}|$, clearly, $m_l \ge 2$. Our experimental results indicate that good performance will be attained by relaxing $DS_{\max}$ to $DS_{\max} - \delta$, where $\delta > 0$ is a fluctuation factor.

Note that SNMF-SS starts with random matrix H, so different implementations may not certainly return identical results. We determine m by repeating the algorithm for certain times, i.e., 50 times and select one solution with maximum D.

• *Similarity and dissimilarity*

Even though the appropriate choice of W_{CL} and W_{ML} based on similarity and dissimilarity matrices is not crucial to the final result of the SNMF-SS, it provides a way to incorporate additional a prior knowledge into algorithm. If there is no a priori knowledge to be used, then SNMF-SS is equivalent to SNMF i.e., $W_{\mathrm{CL}} = 0$ and $W_{\mathrm{ML}} = 0$. Given a similarity or dissimilarity matrix $S = (S_{ij})$ and a threshold $\hbar = \max_{S_{ij} \in S} S_{ij}$, $W_{\mathrm{ML}} = (W_{ij})$ is obtained as $W_{ij} = \mathrm{e}^{S_{ij} - \hbar}$ if $S_{ij} \ge \hbar$ and 0 otherwise. It is easy to see that $W_{ij} = 1$ if and only if $S_{ij} = \max_{S_{ij} \in S} S_{ij}$. The construction of W_{CL} is analogous to W_{ML}.

There are various similarity or dissimilarity measures which can be used [22–26], In SNMF-SS, we take *diffusion kernel feature matrix* [22] for K, the shortest path based similarity [23] for W_{CL}, and the adjacency matrix based similarity [24] for W_{ML}. Of course, other measures can also be adopted.

3.3. Computational complexity

The computational complexity of the SNMF-SS algorithm is investigated in this subsection. First, we focus on the space complexity. Given a graph $G = (V, E)$, a $|V| \times |V|$ adjacent matrix A is needed to store G. W_{ML} and W_{CL} can be directly obtained from matrix A, which does not increase the space complexity. The space complexity for the factor matrix H is $O(|V|m)$, where m is the number of clusters. Since $m < |V|$, the memory complexity of SNMF-SS is $O(|V|^2)$, linear to the number of edges.

The SNMF-SS algorithm contains two major components: computing a partition using SNMF with given m and determining the optimal m. To compute a partition, SNMF terminates when the $J_{\text{SNMF-SS}}$ is lower than a predefined threshold or its maximum iteration γ is reached. In each iteration, all these elements in factor matrix H are updated according to

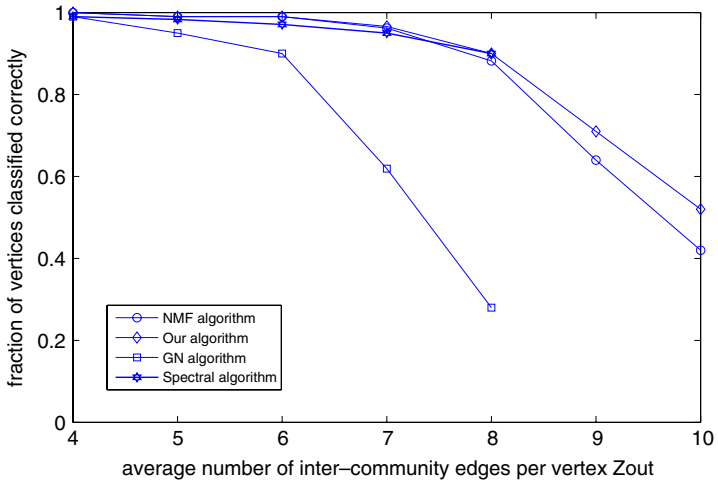

Fig. 1. Test of various algorithms on computer generated networks with fixed community number. Each points is an average over 100 realizations of the networks.

Eq. (3.2). To finish it, matrices $\bar{K}H$ and $HH^{\mathrm{T}}H$ should be computed beforehand. The time complexity for calculating $\bar{K}H$, $HH^{\mathrm{T}}H$ are $O(|V|^2m)$ and $\min\{O(|V|^4m^2), O(|V|^2m^4)\}$, respectively. Since $m \ll |V|$, then $\min\{O(|V|^4m^2), O(|V|^2m^4)\} = O(|V|^2m^4)$ holds. Thus, the time complexity for computing a partition based on SNMF is $O(|V|^2m^4\gamma)$. To determine the optimal m, we just keep increasing m to the point where the D value is never increased. In the worst case, m starts with 2. Thus, the time complexity of SNMF-SS is $O(|V|^2m^5\gamma)$, which shows that the computational complexity depends on the maximum iteration number, size of network and the number of clusters. When γ is a constant, the time complexity can be simplified to $O(|V|^2m^5)$, which demonstrates that SNMF-SS is a time-consuming approach. Actually, our algorithm runs much faster than the theoretic time because of the sparsity of complex networks.

There are two central features that distinguish our algorithm from previous NMF-based approaches [6,7]. First, these methods just directly implement NMF without detailed explanation as to why NMF can be used for the community detection problem, but our algorithm is based on the equivalence of D and SNMF. Second, Previous NMF-based algorithms make use of one similarity measure at a time to capture the topology structure of networks, which throws away useful information contained in other measures. Our algorithm removes such a restriction via a semi-supervised strategy.

4. Experimental results

In this section, the SNMF-SS algorithm is applied to a class of widely used artificial networks and well studied real-world networks with an immediate purpose to test the performance of SNMF-SS. Firstly, we put an emphasis on extracting reasonable communities and determining the correct number of communities; Secondly, we concentrate on the resolution limit problem. The SNMF-SS is coded using the MATLAB version 6.5.

4.1. Performance on discovering community structure

4.1.1. The GN benchmark

Firstly, as a controlled test of how well SNMF-SS performs, we have generated networks with a known community structure to see if the algorithm can recognize and discover the structure.

This network set was originally designed by Girvan and Newman (GN) and has been widely used to test a lot of community detection algorithms [1,6,8,9,18,29]. Each network has 128 nodes divided into 4 communities of 32 vertices each. The links are distributed randomly with a constant probability Z_{in} for a link to occur for each pair of intra-community nodes and another constant probability Z_{out} for each pair of outer-community nodes so as to keep the average degree of a node at 16. For each Z_{out}, 100 networks are generated. Obviously, as the Z_{out} increases, the community structure is more vague. The average results on 100 networks are summarized in Fig. 1, which is a plot of classification accuracy (the fraction of nodes that are classified into their correct communities) as function of Z_{out}. From Fig. 1, we can see that the NMF algorithm [6], spectral algorithm [22] and our algorithm have good performances when $Z_{out} \leq 7$. However, when $Z_{out} \geq 8$ networks are more and more difficult to be dealt with, our algorithm is superior to the NMF algorithm, for instance, for 100 random networks with $Z_{out} = 10$, on average about 53% of nodes are classified correctly by our algorithm, while only about 42% of nodes by the NMF algorithm. A possible explanation about the results in Fig. 1 is that when $Z_{out} \leq 7$, the community structure in networks is so obvious that one simple measure index, such as spectral index, adjacent index, can grasp the topology structure easily. When $Z_{out} \geq 8$, the community structure is not obvious enough to be measured by one measure index. That's why our algorithm obviously outperforms the NMF algorithm under $Z_{out} \geq 8$ only.

To test the robustness of SNMF-SS, we adopted the same method in Ref. [29] to construct asymmetric network; i.e., three of four groups in the benchmark test are merged to form a series of test networks, each with one large group of 96 nodes

Table 1
Benchmark performance for symmetric and asymmetric group detection measured as fraction of correct assignments, average over 100 network realizations with the standard deviation in parentheses.

Group	Z_{out}	Compression [29]	Q [36]	D [18]	SNMF-SS
Symm.	6	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)
	7	0.97 (0.01)	0.97 (0.02)	0.97 (0.02)	0.97 (0.02)
	8	0.87 (0.08)	0.89 (0.05)	0.91 (0.03)	0.90 (0.04)
Node asymm.	6	0.99 (0.01)	0.85 (0.04)	0.99 (0.01)	0.97 (0.01)
	7	0.96 (0.04)	0.80 (0.03)	0.98 (0.02)	0.93 (0.02)
	8	0.82 (0.10)	0.74 (0.04)	0.94 (0.03)	0.89 (0.05)
Link Symm.	2	1.00 (0.00)	1.00 (0.01)	1.00 (0.00)	1.00 (0.00)
	3	1.00 (0.00)	0.96 (0.03)	1.00 (0.00)	0.99 (0.01)
	4	1.00 (0.01)	0.74 (0.10)	0.99 (0.01)	0.96 (0.03)

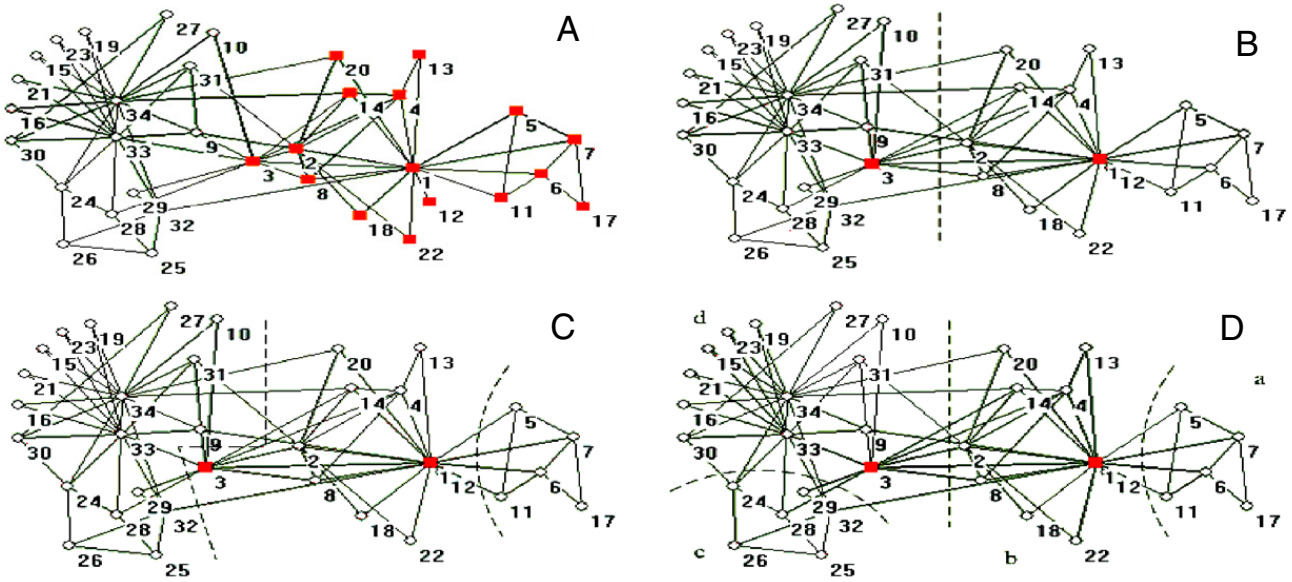

Fig. 2. The results of NMF-algorithm, k-means algorithm and our algorithm applied to karate club network. (A): Zacharyaŕs karate club network. Square nodes and circle nodes represent the instructoraŕs faction and the administratoraŕs faction, respectively. This figure is from Newman and Girvan [1]. (B): The NMF algorithm divides the network into 2 parts. (C): k-means discovers 3 communities. (D): Our algorithm found two kinds of partition with $m = 2$ and $m = 4$, respectively.

and one small group of 32 nodes. These asymmetrically sized networks are harder for both the Q optimization algorithm, clustering compression algorithm and the SNMF-SS algorithm. The results show that the fraction of correct assignments discovered by SNMF-SS is slightly lower than the D-value method [18], and larger than the Q optimization algorithm, which indicates that SNMF-SS can identify the underlying community structure. Finally, we conducted another set of benchmark test using the link asymmetric networks [29] composed of two groups each with 64 nodes but with different average degrees of 8 and 24 links per node. For these networks, we use $Z_{out} = 2, 3, 4$, for which the SNMF-SS algorithm has a comparable result with D-value algorithm and can detect community structure. Note that even though our algorithm is inferior to the D-value algorithm and the compression algorithm, our result is also acceptable since the maximum deviation is 0.003 (see Table 1).

4.1.2. Karate club network

The famous karate club network introduced by [30] is widely used as a test example for methods of identifying communities in networks [1,8,14,9]. 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 clubaŕs administrator and the clubaŕs instructor, the club split into two smaller ones.

The computational results for this experiment are summarized in Fig. 2, which shows the community structure found by NMF algorithm [8], k-means algorithm (www.fmt.vein.hu/softcomp) and our algorithm, respectively. The colored vertices are those critical ones may be misclassified. We can see that the NMF algorithm divides the network into two parts, and k-means the algorithm divides it into three parts. However, our algorithm finds that $m = 2(a+b, c+d)$ and $m = 4(a, b, c, d)$. Note that there are some nodes, say node 3, may belong to different communities with various algorithms. Maybe such nodes play a 'bridge' role in two or more communities in networks. Compared with the real network, our result is more reasonable. It is noted that our partition is consistent with the D-value algorithm [18].

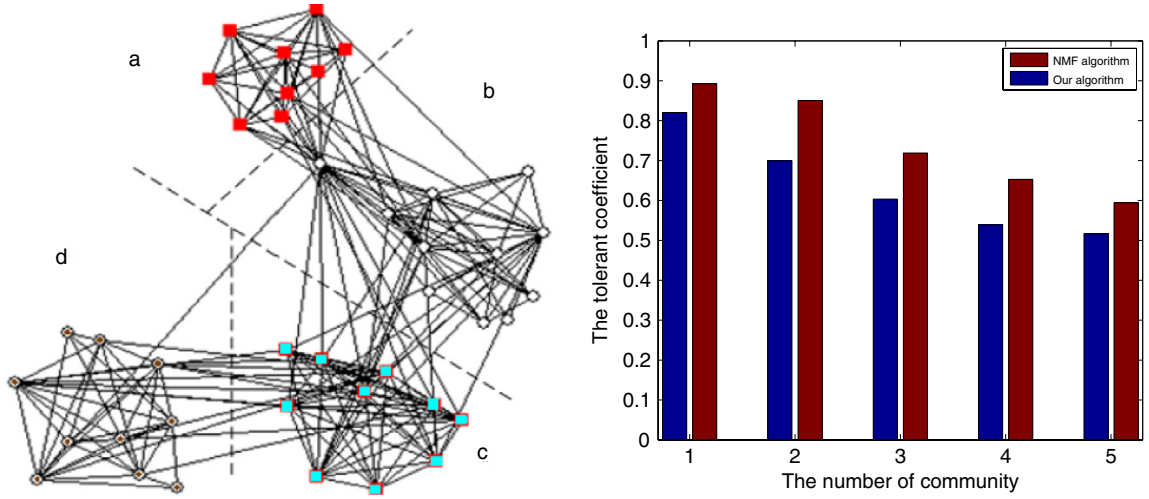

Fig. 3. (L): The optimal partitioning found by our algorithms and vertices with same color belong to the same community. (R): The tolerant coefficient for NMF algorithm (Red) and our algorithm (Blue) in network partitioned into 1–5 different modules. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.1.3. Journal index network

The journal index network employed in Ref. [17] consists of 40 journals as nodes from 4 different fields: physics, chemistry, biology, and ecology and 189 edges connecting nodes if at least one article from one journal cites an article in the other journal during 2004. Ten journals with the highest impact factor in the 4 different fields were selected.

Like literature [18], we also partition the network into 1–5 modules, but as such, $m = 4$ achieves the maximum Q value. Our experiment results for this network is presented in Fig. 3. Both the NMF algorithm [8] and our algorithm can attain the optimal partitioning in the left graph of Fig. 3, which is consistent with that in Ref. [17] where hub *Physical Review Letters* (PRL) is categorized into the chemistry cluster. For this mistake, there are two reasons possibly. First, the structural role that PRL plays in the un-weighted journal networks is that of a chemistry journal and it is closely linked to biology and somewhat ecology. Second, algorithms based on NMF are difficult to converge in definite iterations under the condition that m is small. As shown in the right one of Fig. 3, the minimum tolerant error ratio ($J_{\text{SNMF-SS}}/\|\bar{K}\|^2$) is >0.5, which indicates both the NMF algorithm and SNMF-SS do not converge to local optimal or a saddle with predefined maximum iterations. However, this does not tamper with the ability to recognize underlying structure.

4.2. Performance on tolerating resolution limit problem

Modularity optimization algorithms are criticized for the serious resolution limit problem [17]. However, a clear advantage of the SNMF-SS over modularity maximization methods is its higher resolution. There are three reasons to explain it. Firstly, literature [18] has showed that the D value improves the resolution limit at large extent. Thus, the equivalence relation between SNMF and D assures that SNMF-SS has the similar ability to escape from it; secondly, SNMF-SS can make use of other prior knowledge to grasp network topological properties; thirdly, compared with approaches based on modularity maximization, algorithm based on NMF is driven by the minimum error instead of Q value.

4.2.1. The LFR benchmark

The GN benchmark [1] is a popular choice for testing community structure identification. It is, however, not an appropriate alternative for resolution limit analysis due to several caveats: all nodes of the graph have essentially the same degree, the communities are all of the same size, and the network is small. Thus, the Lancichinetti, Fortunato, and Radicchi (LFR benchmark) [31] is adopted because it can produce communities with both small and large sizes. It has the similar spirit of the GN benchmark, but is considerably more realistic. It allows the user to specify distributions both for the community sizes 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 $\mu = 1$, 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 [32].

For several 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. Based on the algorithm presented in [31], we generate each graph whose node degree was taken from a power law distribution with exponent 2 and community size from a power law distribution and with exponent 1. Each graph has 1000 vertices, average degree 15, maximum degree 50, maximum for the community sizes 50 and minimum for the community sizes 5. Such a long-tailed distribution results in a much larger variance than the respective means.

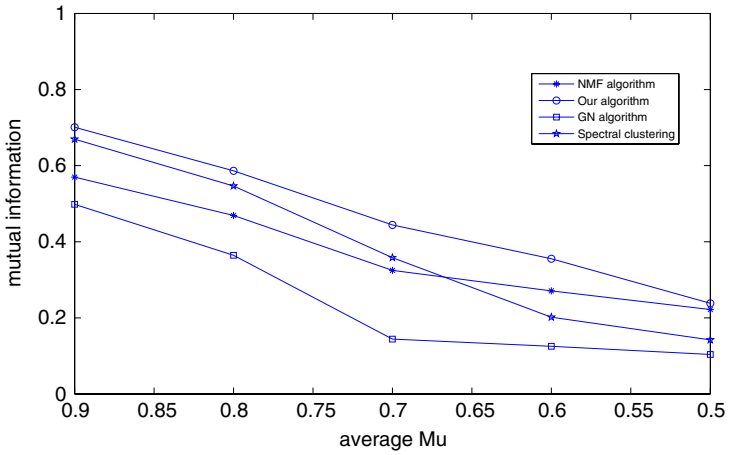

Fig. 4. Test of various algorithm on the LFR benchmark. Each point is an average over 50 instances.

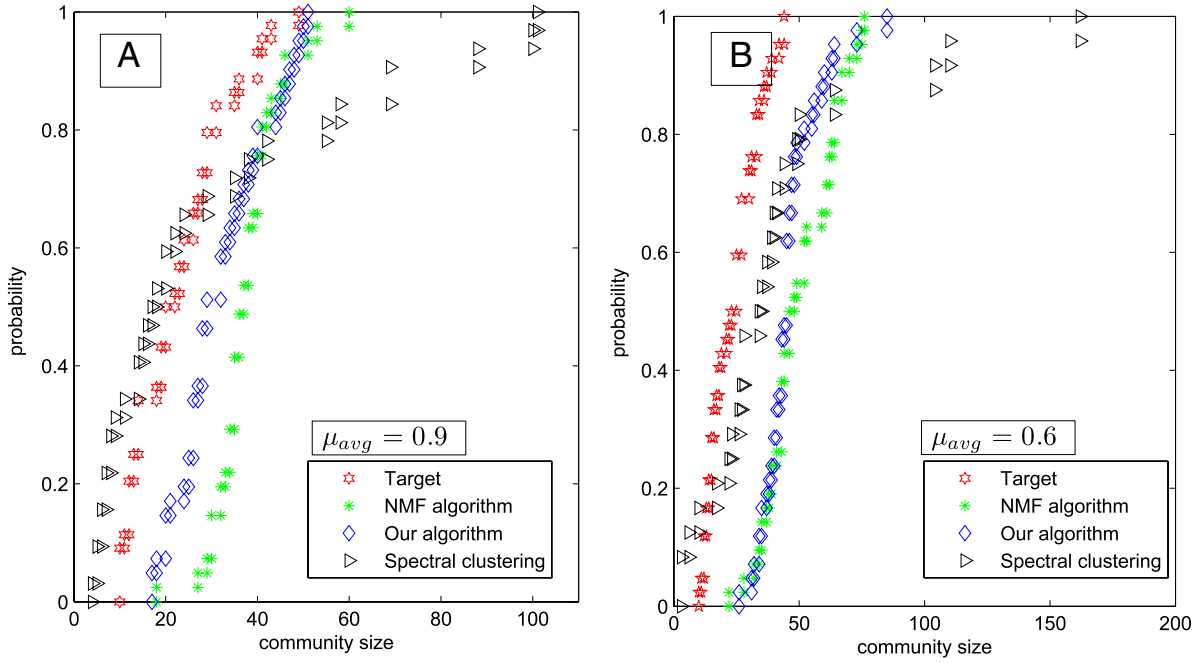

Fig. 5. The distributions of natural community sizes and discovered community sizes with different μ. (A): $\mu_{\text{avg}} = 0.9$, (B): $\mu_{\text{avg}} = 0.6$.

Since the community structure in these networks is known, we introduce a normalized mutual information index as a measure of similarity between two partitions P and P' [33]

$$I(P, P') = \frac{-2\sum_{i=1}^{|P|}\sum_{j=1}^{|P'|} N_{ij}\log\left(\frac{N_{ij}N}{N_{i.}N_{j}}\right)}{\sum_{i=1}^{|P|} N_{i.}\log\left(\frac{N_{i.}}{N}\right) + \sum_{i=1}^{|P'|} N_{.j}\log\left(\frac{N_{j}}{N}\right)}, \tag{4.1}$$

where N is a $|P| \times |P'|$ matrix with entry N_{ij} denotes the number of nodes in the real community i that appear in the found community j, $N_{i.}$ is the sum of the i-th row of N and the sum over column j is denoted by $N_{.j}$.

The average results on 50 networks are summarized in Fig. 4, which is a plot of mutual information as function of the μ_{avg}. Careful contrast of Figs. 1 and 4 shows that all these algorithms have a much better accuracy in the GN benchmark than that of the LFR benchmark. The reason is that since the number of clusters is given and the community size is fixed, the GN benchmark is friendly to many algorithms. However, when the size of the graph starts to dwarf the sizes of the natural communities, the problem becomes much harder. Agreeing with the resolution limit argument, algorithms based on the optimization of the Q value are not able to resolve most of the natural communities. From Fig. 4, we can see that the peak value of GN algorithm is around 0.5 at $\mu_{\text{avg}} = 0.9$, however, others can obtain a better performance. In detail, SNMF-SS algorithm can achieve its peak above 0.70 at $\mu_{\text{avg}} = 0.9$, spectral clustering roughly 0.68 and NMF 0.57. Since μ_{avg}

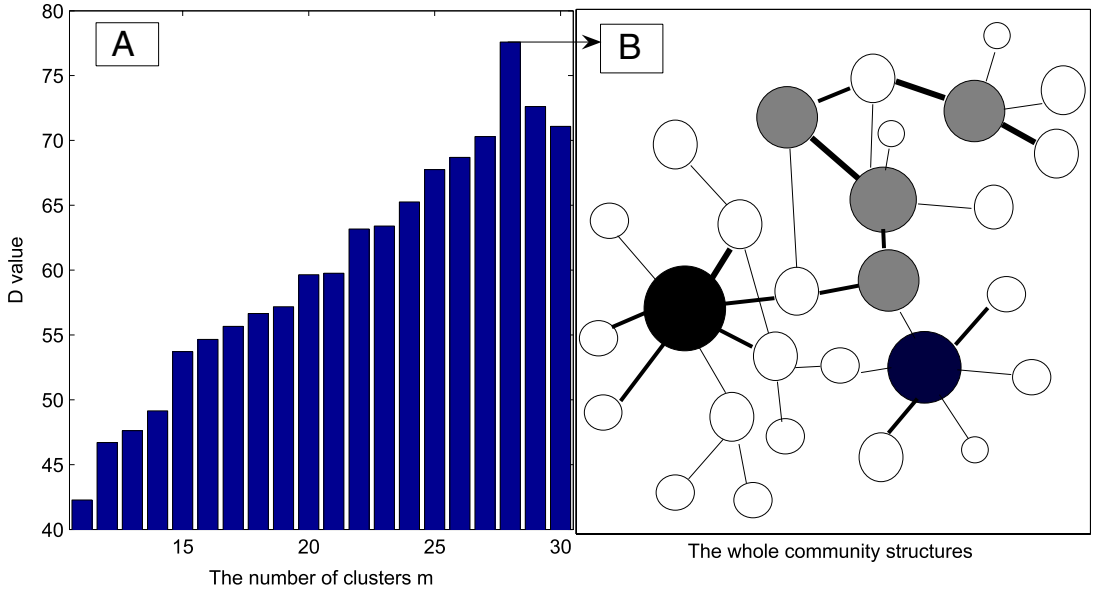

Fig. 6. The results of the SNMF-SS algorithm applied to the collaboration network. (A): D value versus m, (B): The communities found by our algorithm, where a circle denotes a cluster, with size varying roughly with the number of vertices in the cluster and the line represents the number of edges connecting different clusters.

determines the ratio of the inner edges to outer edges, it becomes harder and harder to detection communities correctly as μ_{avg} decreases. Spectral clustering has a good performance when the community structure is clear, while its performance decreases greatly as μ_{avg} decreases. Furthermore, we can also see that, compared with these four algorithms, the SNMF-SS algorithm gains the best performance.

In addition to the classification accuracy, we are also interested in comparing the distributions of the sizes of communities discovered by algorithms to the original distributions used in LFR generation. Fig. 5 contains these results. The target line is the empirical cumulative distribution function (CDF) for the natural community sizes sampled by LFR. The other lines are empirical CDF's for the sizes of communities discovered by NMF, spectral clustering and SNMF-SS. When $\mu_{\text{avg}} = 0.9$, communities in such networks are likely to be cliques, not necessarily cliques. However, the sizes of the communities are not equal, which makes it hard to be detected. Still, SNMF-SS can resolve 70.08% vertices correctly, while NMF and spectral clustering can only obtain 56.98%, 66.96% respectively. When $\mu_{\text{avg}} = 0.6$, the communities are not as well defined, which increases the difficulty for the algorithm in discovering the correct community. From the right panel of Fig. 5, we see that the spectral clustering can find much smaller community sizes than NMF and SNMF-SS, but it also discovers much larger community sizes. From Fig. 5, it is easy to discover that roughly 0.5% of the vertices are classified by spectral clustering into communities that are even smaller than the smallest natural community. This problem also appeared in Ref. [32]. However, both the SNMF-SS and NMF can get rid of this case. Furthermore, the CDF of community sizes found by our algorithm is much closer than those by NMF and spectral clustering.

4.2.2. Collaboration network

Finally, a large-scale co-authorship network of scientists working on network theory and experiment, as published in Ref. [34]. The network has a total of 1589 vertices and 2742 weighted, undirected edges. The weights are derived from the number of joint publications, if authors i and j share a paper where both authors and the paper has t total co-authors, this contributes by $\frac{1}{t}$ to the total weight of the edge. It makes no sense to find a partition based on the whole network since there are 396 connected subgraphs. Thus, we extracted the giant component of the networks consisting of 379 scientist and 914 links.

Applying the SNMF-SS algorithm to this network again, the modularity density D value varies with the cluster number m presented in Fig. 6(A). It is easy to find that the modularity density has a strong peak at 28 communities with a value of $D = 77.58419$, while the optimum m discovered by [35] is 30. The reason is that the main goal of [35] is to uncover the fuzzy communities.

Since the network is too large, it is not convenient to figure a diagram showing the concrete clustering results. To reduce the level of complexity to one that can be interpreted by the human eye, we have reduced the network to only groups in Fig. 6(B). In this panel, we have drawn each cluster as a circle, with their size varying roughly with the number of vertices in the cluster. Different colors on circles have no additional meanings just for convenience. The line between clusters i and j represents the number of edges connecting them with the thickness shows the number of edges. The thicker the line is, the more the edges are.

Since the actual communities are unknown in the collaboration network as the LFR benchmark, we cannot compute the mutual information and compare CDF as the last subsection. Here, we turn our focus to the small communities found

Table 2
Parameters about the 15 clusters whose edge numbers are smaller than the $\sqrt{914/2} \approx 20$, where V_{size}, E_{in}, E_{out}, D_c are the number of vertices, the number of edges in this cluster, edges leave from this cluster, contributing to the D value.

No	V_{size}	E_{in}	E_{out}	D_c
1	13	18	7	2.230769e+000
2	10	15	7	2.300000e+000
3	10	17	2	3.200000e+000
4	9	18	2	3.777778e+000
5	9	17	2	3.555556e+000
6	8	15	4	3.250000e+000
7	8	16	4	3.500000e+000
8	7	10	2	2.571429e+000
9	6	8	3	2.166667e+000
10	6	11	2	3.333333e+000
11	6	9	2	2.666667e+000
12	5	10	6	2.800000e+000
13	5	10	5	3.000000e+000
14	4	6	2	2.500000e+000
15	3	3	1	1.666667e+000
Sum	109 (0.2860)	183 (0.2002)	51 (0.0558)	42.5189 (0.5480)

by SNMF-SS. According to Ref. [17], modularity maximization algorithms may fail to resolve communities with fewer than $\sqrt{L/2}$ edges, where L is the number of edges in entire network. However, SNMF-SS can tolerate the resolution limit problem at large extent. From Fig. 6(B), we can easy find that there are few large communities while there are much more small communities. In detail, our algorithm can find 15 small communities whose edge numbers are less than $\sqrt{914/2} \approx 20$ as shown in Table 2. From Table 2, we can conclude that the 15 small clusters consume 28.60% vertices from the second column. There are 20.02% edges among the clusters and 5.58% between clusters, which indicates that the 15 small However, only 54.80% of the contribution is presented by the 15 ones that shows that the rest of the 13 large clusters are also valid. Such an experiment indicates that the SNMF-SS algorithm can tolerate the resolution limit problem by discovering communities smaller than $\sqrt{L/2}$.

5. Conclusion

In this paper, we introduce a semi-supervised clustering for community detection in complex networks by nonnegative matrix factorization, and this algorithm can choose the proper number of clusters and discovery community structure correctly. There are two central features in our algorithm. First, we further the theoretic result in Ref. [18] by showing the equivalence of the objective functions of SNMF and modularity density D. Second, previous methods make use only of one similarity measure to capture the topology structure of networks, which throws away useful information contained in other measures. Our algorithm removes such restriction via semi-supervised strategy.

We would like to close this paper by posing two interesting problems. First, algorithms based on NMF are time-consuming, especially for large-scale networks. Second, as we see in experiment 3, the absolute error of NMF-based algorithms is too large for small m (the number of clusters). Thus, designing a method which can solve these two issues will be very practical.

Acknowledgements

This work was supported by NSFC-Microsoft Research Asia Joint Research Fund (Grant No. 60933009), the Ph.D Programs Foundation of Ministry of Education of China (Grant No. 200807010013) and NSFC (Grant No. 60970065). The authors thank Professor M.E.J. Newman for providing the data of karate club network and the scientist collaboration network. We thank anonymous reviewers greatly for their time and helpful comments. That makes this paper more interesting and informative.

References

[1] M. Girvan, M.E.J. Newmann, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99 (12) (2002) 7821–7826.
[2] D. Gibson, J. Kleinberg, P. Raghavan, Inferring web communities from link topology, in: ACM ICHH, 1998, pp. 225–234.
[3] H.C. Lu, X.P. Zhu, H.F. Liu, et al., The interactome as a tree-an attempt to visualize the protein–protein interaction networks in yeast, Nucleic Acids Res. 32 (2004) 4804–4811.
[4] J.M. Pujol, J. Béjar, J. Delgado, Clustering algorithms for determining community in large networks, Phys. Rev. E 74 (2006) 016107.
[5] S. White, P. Smyth, A spectral clustering approach to finding communities in graphs, in: SIAM ICDM, 2005.
[6] R.S. Wang, S.H. Zhang, Y. Wang, et al., Clustering complex networks and biological networks by nonnegative matrix factorization with various similarity measures, Neurocomputing 72 (2008) 134–141.
[7] S.H. Zhang, R.S. Wang, X.S. Zhang, Uncovering fuzzy community structure in complex networks, Phys. Rev. E 76 (2007) 046103.

[8] S.H. Zhang, R.S. Wang, X.S. Zhang, Identification of overlapping community structure in complex networks using fuzzy *c*-means clustering, Physica A 374 (2007) 483–490.
[9] M.E.J. Newmann, Detecting community structure in networks, Eur. Phys. J. B 38 (2004) 321–330.
[10] L. Danon, J. Duch, A. Diaz-Guilera, A. Arenas, Comparing community structure identification, J. Stat. Mech. Theory Exp. (2005) P09008.
[11] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791.
[12] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: ACNIPS, 2001, pp. 362–371.
[13] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2004) 026113.
[14] J. Duch, A. Arenas, Community identification using extremal optimization, Phys. Rev. E 72 (2005) 027104.
[15] M.E.J. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. USA 103 (23) (2006) 8577–8582.
[16] U. Brandes, D. Delling, M. Gaertler, et al., On modularity clustering, IEEE Trans. Knowl. Data Eng. 20 (2008) 172–188.
[17] S. Fortunato, M. Barthélemy, Resolution limit in community detection, Proc. Natl. Acad. Sci. USA 104 (36) (2007) 36–41.
[18] Z.P. Li, S.H. Zhang, R.S. Wang, et al., Quantative function for community detection, Phys. Rev. E 77 (2008) 036109.
[19] G. Palla, I. Derényi, I. Farkas, et al., Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (2005) 814–818.
[20] E. Ravasz, A.L. Somera, D.A. Mongru, et al., Hierarchical organization of modularity in metabolic networks, Science 297 (2002) 1551–1555.
[21] J. Reichardt, S. Bornholdt, Detecting fuzzy community structures in complex networks with a Potts model, Phys. Rev. Lett. 93 (2004) 218701.
[22] R.I. Kondor, J. Lafferty, Diffusion kernels on graphs and other discrete input space, in: ICML, 2002, pp. 315–322.
[23] M. Gustafsson, M. HÖrnquista, A. Lombardi, Comparison and validation of community structures in complex networks, Physica A 367 (2006) 559–576.
[24] M.P. Samanta, S. Liang, Predicting protein functions from redundancies in large-scale protein interaction networks, Proc. Natl. Acad. Sci. USA 100 (22) (2003) 12579–12583.
[25] S. Sun, Y. Zhao, Y. Jiao, et al., Faster and more accurate global protein function assignment from protein interaction networks using the MFGO algorithm, FEBS Lett. 580 (2006) 1891–1896.
[26] G. Wang, Y. Shen, E.J. Luan, A measure of centrality based on modularity matrix, Prog. Nat. Sci. 18 (2008) 1043–1047.
[27] N. Cristianini, J. Shawe-Taylor, Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods, Cambridge Univ. Press, 2000.
[28] C. Ding, X. He, H.D. Simon, On the equivalence of nonnegative matrix factorization and spectral clustering, in: SIAM ICDM, 2005, pp. 606–610.
[29] M. Rosvall, C.T. Bergstron, An information-theoretic framework for resolving community structure in complex networks, Proc. Natl. Acad. Sci. USA 104 (18) (2007) 7327–7331.
[30] W.W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. 33 (1977) 452–473.
[31] A. Lancichinetti, S. Fourtunato, F. Radicchi, Bechmark graphs for testing community detection algorithm, Phys. Rev. E 78 (2008) 046110.
[32] J.W. Berry, B. Hecdrickson, R.A. LaViolette, C.A. Phillips, Tolerating the community detection resolution limit with edge weighting, 2009. arXiv:0903.1072.
[33] L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, J. Stat. Mech. Theory Exp. (2005) P09008.
[34] M.E.J. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 (2006) 036104.
[35] T. Nepusz, A. Petróczi, L. Négyessy, et al., Fuzzy communities and the concept of bridgeness in complex networks, Phys. Rev. E 77 (2008) 016107.
[36] A. Medus, G. Acunã, C.O. Dorso, Detection of community structures in networks via global optimization, Physica A 358 (2005) 593–604.