On the Spectral Characterization and Scalable Mining of Network Communities

Bo Yang, Jiming Liu, Jianfeng Feng, and Dayou Liu

Abstract—Network communities refer to groups of vertices within which their connecting links are dense but between which they are sparse. A network community mining problem (or NCMP) is concerned with the problem of finding all such communities from a given network. A wide variety of applications can be formalized as NCMPs, ranging from social and/or biological network analysis to Web mining and searching. So far many algorithms addressing the NCMP have been developed and most of them fall into the categories of either optimization based or heuristic methods. Distinct from the existing studies, the work presented in this paper explores the notion of network communities and their intrinsic properties from the dynamics of a stochastic model naturally introduced. In the paper, a relationship between the hierarchical community structure of a network and the local mixing properties of such a stochastic model has been established with the large deviation theory. Critical topological information regarding to the community structures hidden in networks can be inferred from their spectral signatures. Based upon the above-mentioned relationship , this work proposes a general framework for characterizing, analyzing, and mining network communities. Utilizing the two basic properties of metastability, i.e., being locally uniform and temporarily fixed, an efficient implementation of the framework, called the LM algorithm, has been developed that can efficiently mine natural communities hidden in networks with scalable performances. The effectiveness and efficiency of the LM algorithm have been theoretically analyzed as well as experimentally validated.

Index Terms—Social network, community mining, modularity, Markov chain, local mixing, large deviation theory.

1 Introduction

Currently online social communities are the most popular applications provided by Web 2.0 portals, in which people with common interests join anytime anywhere to freely share information, experiences, opinions, services, and other useful resources. Techniques that can automatically discover such virtual communities will provide huge help in building and managing personalized and smart Web portals through analyzing and predicting the collective behaviors of users through mining their underlying community structures. Formally, this task can be formulated as the network community mining problem (NCMP), which aims to discover all communities from a given network. A network community generally refers to a group of vertices within which the connecting links are dense but between which they are sparse. Particularly, network communities in different contexts may be circles of a society within which people share common interests and keep more contacts, groups of proteins with similar functions, or clusters of Web pages related to common topics. Besides online social community discovering, a wide variety of applications can be represented as NCMPs, ranging from social network analysis [1], [2], [3], [4], biological network analysis [10], [11], [12], [13], [14], [15] to Web mining and Web searching [16], [17], [19], [20]. Thus, how to effectively and efficiently solve such NCMPs is of fundamental importance for both theoretical research and practical applications. This paper aims to present a novel approach to characterizing and analyzing network communities as well as a scalable algorithm for mining large-scale networks in practice.

- *B. Yang and D. Liu are with the College of Computer Science and Technology, Jilin University, Changchun, China 130012. E-mail: {ybo,dyliu}@jlu.edu.cn.*
- *J. Liu is with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong. E-mail: jiming@comp.hkbu.edu.hk.*
- *J. Feng is with the Department of Computer Science and Mathematics, Warwick University, Coventry, CV4 7AL, UK. E-mail:jianfeng.feng@warwick.ac.uk.*

Manuscript received xx xx, 2008; revised xx xx, 2008.

1.1 Related Work

So far, many methods addressing NCMPs have been developed. In terms of their basic ideas and strategies adopted, they could be classified into two main categories: (1) optimization based and (2) heuristic methods.

The optimization based methods solve an NCMP by transforming it into an optimization problem and trying to find an optimal solution for a predefined objective function, such as different kinds of cut criteria adopted by different spectral methods [5], [6], [7], [8], [9], the evaluation function used in theKernighan-Lin algorithm [21], the Q function proposed by Newman [22] and employed in several algorithms [7], [11], [15], [22], [23], [24], the energy function of a Potts model with multiple states [25], and the likelihood of a hierarchical random graph [26].

On the other hand, in the heuristic methods, there are no explicit optimization objectives. They solve the NCMP based on some intuitive assumptions or heuristic rules. For example, the heuristic rule used in the maximum flow community (MFC) algorithm [16] is that the "flows" through inter-community links should be

larger than those of intra-community links. Similarly, the heuristic rule employed by the GN algorithm [1] is that the "edge betweenness" of inter-community links should be larger than that of intra-community links. The Wu-Huberman algorithm [27], Clique Percolation method [2], the Radicchi algorithm [28], and the FEC (Finding and Extracting Communities) algorithm [29] adopted different assumptions, respectively.

Besides the above-mentioned two main categories, there exist some other methods for solving the NCMP. For example, we can cluster a network through a bottom-up means by repetitively joining pairs of current groups based their similarities, such as correlation coefficient, Euclidean distance, or Pearson correlation [30], which are defined in terms of their linkage relations.

1.2 Problems

The NCMP still remains as an open problem although considerable efforts have been taken for solving it. So far, the existing work on the NCMP focuses on how to effectively as well as efficiently partition networks into communities. However, quite a few fundamental issues concerning *network modularity* have not yet been clearly answered. For example,

- what are the deep meanings of network communities?
- why and how do they come into being hierarchically?
- are there any intrinsic relationships between the modularity and other phenomena demonstrated in networks, such as the stability or the metastability of complex systems?
- are there any hidden but significant relationships between the modularity and the characteristic features of networks, such as the eigenvalues of their adjacency matrices?

Without a clear understanding the above issues, we can only model and compute communities by subjectively introducing some optimization objectives or heuristic rules, i.e., relying on human intuitive observations of the external behaviors as demonstrated in networks containing community structures. Therefore, a basic but fundamental challenge the researchers of this community is now facing can be stated as follows: Can we build an objective mathematical model based only on the intrinsic features of networks themselves, in order to understand, characterize, analyze network modularity, and thereafter provide a new perspective on efficiently and effectively solving the NCMP?

1.3 Our contributions

Unlike all existing methods, this paper tries to explore the notion of network modularity and partially answer the above questions by means of understanding the dynamics of networks, instead of starting from a predefined optimization objective or a heuristic assumption. Previously, we have proposed a heuristic algorithm to partition signed social networks based on a proposed random walk model [29]. We have found that the dynamics of such a stochastic model can naturally reflect the intrinsic properties of networks with modularity structures and exhibit local mixing behaviors. In this work, based on the large deviation theory, we will explain this phenomenon and uncover a connection between the modularity structure of a network and its metastability and propose a new measurement, spectral signature, to characterize and analyze network modularity. For any given network, without clustering it using any particular algorithms, from its spectral signature, one can infer some critical information related to its modularity, such as the quality of its community structures, the cohesion and separatability of communities, the number of communities, and the hierarchical structure of communities. Based on the above connection and new metrics, we will present a theoretical framework for characterizing, analyzing, and mining communities. Utilizing the basic properties of local mixing, we will then propose an efficient implementation for this framework, called the LM (Network community mining based upon *L*ocal *M*ixing properties) algorithm, to solve the NCMP in practice. As to be demonstrated and discussed later, the LM is both scalable and effective.

The remainder of the paper is organized as follows: Section 2 presents a stochastic model and discusses its dynamics. Section 3 gives the concept of network spectral signature, and discusses its implications to the topological information related to modularity. Then, a general framework for characterizing, analyzing, and mining communities in networks are proposed. Section 4 proposes a scalable community mining algorithm, and analyzes its time complexity theoretically. Section 5 provides the experimental results of testing and validating the performances of the above algorithm. Section 6 discusses the distinctions of this work from others as well as its improvements and limitations. Finally, Section 7 concludes the main results and contributions of this work.

2 A Stochastic Model and its Dynamics

In this section, we will discuss the connection between the community structure of a network and the local mixing properties demonstrated by a stochastic model introduced for the network. The basic idea behind this model can be described as follows: before its dynamics reaches its globally stable state, it will go through a hierarchy of local mixing states (metastable states) first, and in each of them, locally uniform transition distributions will be observed. Based on the large deviation theory, one can estimate the hitting and exiting times of each local mixing state in terms of the spectrum of a network. Important information related to network modularity, such as the number, the shape, and the hierarchical

structure of communities, can be characterized by such times.

2.1 A stochastic model

Let $N = (V, E)$ denote a network, where V is the set of vertices and E is the set of edges (or links). Consider a stochastic process defined on N, in which an agent freely walks from one vertex to another along the links between them. After the agent arrives at one vertex, it will randomly select one of its neighbors and move there.

Let $X = \{X_t, t \geq 0\}$ denote the agent positions, and $P\{X_t = i, 1 \leq i \leq n\}$ be the probability that the agent hits the vertex i after exact t steps. For $i_t \in V$, we have $P(X_t = i_t | X_0 = i_0, X_1 = i_1, \cdots, X_{t-1} = i_{t-1}) = P(X_t = i_t | X_{t-1} = i_{t-1})$. That is, the next state of the agent is determined only by its previous state (Markov property). So, this stochastic process is a discrete *Markov chain* and its state space is V. Furthermore, X_t is $homogeneous$ because of $P(X_t = j | X_{t-1} = i) = p_{ij}$, where p_{ij} is the transition probability from vertex i to vertex j. In terms of the adjacency matrix of N, $A = (a_{ij})_{n \times n}$, p_{ij} is defined by

$$p_{ij} = \frac{a_{ij}}{\sum_j a_{ij}} \tag{1}$$

Let $D = diag(d_1, \cdots d_n)$, where $d_i = \sum_j a_{ij}$ denotes the degree of vertex i. Let P be the transition probability matrix, we have:

$$P = D^{-1} A \tag{2}$$

Let $p_{ij}^{(t)}$ be the probability of hitting vertex j after t steps starting from vertex i, we have:

$$p_{ij}^{(t)} = (P^t)_{ij} \tag{3}$$

A network is called *ergodic* if the Markov chain associated with it is ergodic. Most real networks, such as social networks, biological networks, and Web, are ergodic due to their high clustering coefficients (it means they contain triangles, and thus their corresponding Markov chains are aperiodic). Particularly, for an undirected ergodic network, it is easy to show

$$\pi_j = \lim_{t \to \infty} p_{ij}^{(t)} = \frac{d_j}{\sum_k d_k} \tag{4}$$

2.2 The dynamics

Now let us consider the dynamics of the above stochastic model. For a network with a community structure, its corresponding Markov chain should contain some metastable states depending on its total number of communities K. Before we theoretically explain the dynamic behaviors, let us first elaborate it using a simple example with a potential function.

Example 1. Given a one-dimensional potential function $H(x)$, a corresponding network can be constructed as follows: 1) sample n points from H as n nodes of the network, 2) for each pair of adjacent points, x_i and x_j, establish two directed links, $< i, j >$ and $< j, i >$, between them. According to the probability $p(x_i, x_j) \propto e^{-(H_j - H_i)/\varepsilon}$, the weight of link $< i, j >$ is defined as

$$a_{ij} = \begin{cases} 1 & \text{if } H_i > H_j \\ e^{-\frac{H_j - H_i}{\varepsilon}} & \text{if } H_i \leq H_j \end{cases} \tag{5}$$

where ε is a positive constant.

Fig. 1(a) shows an illustrative potential function, in which 20 points are sampled from four intervals $A = [6, 9], B = [11, 15], C = [27, 29]$, and $D = [31, 33]$, respectively, which contain four metastable states. Fig. 1(b) shows the network constructed by these samples, in which the weights of links within intervals are calculated according to Eq.(5), and those across intervals are calculated by $a_{ij} = e^{-\Delta H/\varepsilon}$, where ΔH is the potential difference between H_i and the barriers between intervals. Note that the weights of links across intervals are much smaller than those within intervals. Naturally, samples within each interval are considered as a network community. Hence, in this case we have four communities with a clear hierarchical structure as shown in Fig. 1(c) (see below for details).

Fig. 1(d) shows the transition probability of the network. Figs.1(e)-(h) depict the dynamics of the chain when it goes through local mixing states to global mixing states. At time T_4 (T_i denotes the time step), each community locally mixes together and we observe four local uniform distributions corresponding to four communities. At time T_3, C and D mix together. At time T_2, A with B and C with D mix together in terms of their hierarchical structure. Finally, at time T_1, a global uniform distribution is finally reached, in which all vertices have almost an identical distribution. This local mixing process corresponds to the hierarchical structure of all four communities, as shown in Fig. 1(c).

Using one of the most well developed theories in mathematics, i.e., the large-deviation theory [31], we can explain the observed dynamical properties in details.

Let H_2, H_3 and H_4 be the potential barriers as depicted in Fig. 1(a) and $H_4 < H_3 < H_2 < H_1 = \infty$. Define

$$T_s^{ext} = e^{H_s/\varepsilon}, \qquad s = 1, 2, 3, 4.$$

(see Fig. 1(i)). Assume that the Markov chain starts from community D. According to the large deviation theory, T_4^{ext} will be the first time for X_t to exit from the community D. In other words, at time $T_4 < T_4^{ext}$, the Markov chain first reaches a local mixing state, as clearly shown in Fig. 1(e). All the row distributions concentrate on the four communities A, B, C, and D, and are well locally mixed.

After T_4^{ext}, the attractor D loses its stability and starts mixing with C. During the time period $[e^{H_4/\varepsilon}, e^{H_3/\varepsilon}]$, C and D merge into one state, as shown in Fig. 1(f). Hence $T_3 \in [e^{H_4/\varepsilon}, e^{H_3/\varepsilon}]$. Similarly during the time period $[e^{H_3/\varepsilon}, e^{H_2/\varepsilon}]$, A and B have perfectly merged into one state, as clearly indicated in Fig. 1(g). Finally, after $T_1 > e^{H_2/\varepsilon}$, the Markov chain is globally and well mixed and there is only one state which has the distribution:

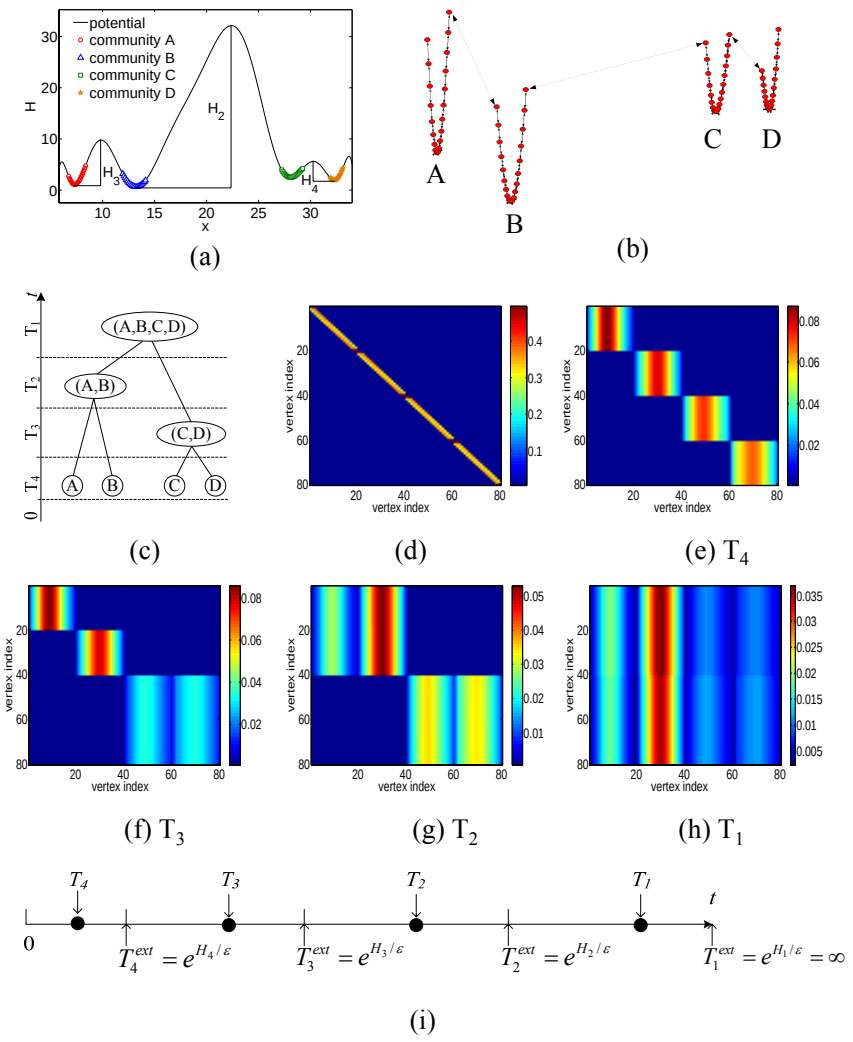

Fig. 1. A potential function and the local mixing states of a Markov chain. (a) A potential function H. (b) The corresponding network containing four communities. (c) The hierarchical structure of the network. (d) The transition matrix. (e)-(g) Three local mixing states. (h) The global mixing state. (i) The local mixing time estimated based on the large deviation theory.

the global minimum, here it is community B. At each time interval of $[0, e^{H_4/\varepsilon}]$, $(e^{H_4/\varepsilon}, e^{H_3/\varepsilon}]$, $(e^{H_3/\varepsilon}, e^{H_2/\varepsilon}]$, $(e^{H_2/\varepsilon}, e^{H_1/\varepsilon} = \infty)$ the Markov process will stay there for a relatively long time, hence the name of metastability.

Before a Markov chain reaches its global mixing state, it should go through a hierarchy of local mixing states first. In each local mixing states, vertices within the same communities have identical row distributions. Correspondingly, in each local mixing state, we should observe a metastable transition matrix, as shown in Fig. 1. That is, random walk will stably stay in a metastable state during a period of time with a probability equal to one, according to the large deviation theory.

In general, the large deviation theory could apply to a network with or without a potential function. For a general case, an action functional rather than a potential function is introduced to characterize the local mixing properties. From the theory above, we see that in terms of the dynamics of the Markov chain (local mixing properties), a clearly hierarchical and endogenous structure is exhibited. The main idea of the paper is to apply the results to develop a model to characterize, analyze, and reveal the community structure in a given network.

2.3 Spectral transitions

For a general network, we can also estimate all local mixing times using the spectrum of its Markov generator $Q = I - P$, where I is the identity matrix. For an undirected network, Q is positive semi-definite and has n non-negative real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \leq 2$. Let T_s^{hit} and T_s^{ext} be the hitting time and exiting time of the $s-th$ local mixing state ($1 \leq s \leq n$). In Example 1, we have $\lambda_s = e^{-H_s/\varepsilon} (s = 1, 2, 3, 4)$. Actually we have [31]:

$$\lambda_s = \frac{1}{T_s^{ext}}(1 + o(1)) \tag{6}$$

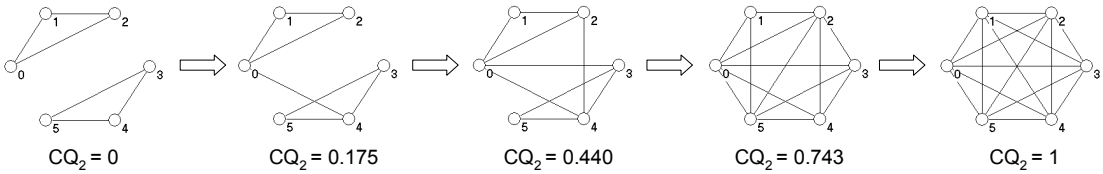

Fig. 2. An evolving process of a dynamic network with corresponding CQ_2 values.

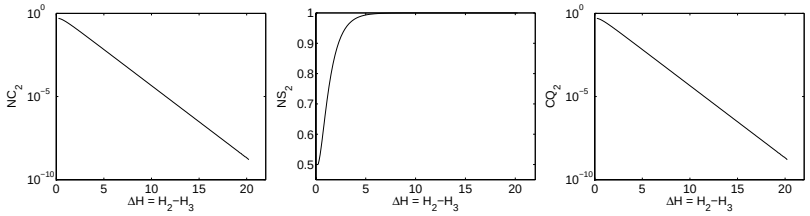

Fig. 3. Controlling NC_2, NS_2, and CQ_2 by regulating $\Delta H = H_2 - H_3$.

Reasonably, we can use the exiting time of the $(s+1)-th$ local mixing state to estimate the hitting time of the $s-th$ local mixing state, that is, $T_s^{hit} = T_{s+1}^{ext} = 1/\lambda_{s+1}$. So, $1/\lambda_1 = \infty$ is the exiting time of the global mixing state. $1/\lambda_2$ is the exiting time of the second local mixing state or the hitting time of the global mixing state, and so on.

3 Spectral Signatures of Networks

3.1 Characterizing a two-community structure

For a network containing two communities, its Markov chain will go through only one local mixing state before hitting the global mixing state. If both communities are close to each other, the chain will hit its local mixing state in a short time. Otherwise, it will be a slow process. After entering the local mixing state and if the two communities separate very well, it will take a long time to exit the state and mix together (corresponding to a metastable state). Otherwise, the chain will exit the state very rapidly and hit its global mixing state (corresponding to a unstable transition state). The cohesion and the separatability of a two-community structure can be characterized by the hitting time (estimated by $1/\lambda_3$) and the mixing time (estimated by $1/\lambda_2$-$1/\lambda_3$) of the local mixing state.

Definition 1. In terms of spectral transitions, the normalized cohesion (NC_2) and the normalized separatability (NS_2) of a two-community structure are defined as follows:

$$NC_2 = \frac{1/\lambda_3}{1/\lambda_2} = \frac{\lambda_2}{\lambda_3} \tag{7}$$

and

$$NS_2 = \frac{1/\lambda_2 - 1/\lambda_3}{1/\lambda_2} = 1 - \frac{\lambda_2}{\lambda_3} \tag{8}$$

Smaller NC_2 means better cohesion, and larger NS_2 means better separatability. Note that $NC_2 + NS_2 = 1$, which means better cohesion means better separatability at the same time and vice versa.

Definition 2. In terms of NC_2 or NS_2, the 2-community quality (CQ_2) to measure how well-formed a two-community structure of a given network is defined as:

$$CQ_2 = NC_2 = 1 - NS_2 = \lambda_2/\lambda_3 \tag{9}$$

It is easy to show that $0 \leq CQ_2 \leq 1$. Networks with small CQ_2 approaching to zero will have clear two-community structures, and those with large CQ_2 approaching to one will have ambiguous or no community structures. In the extreme cases, the CQ_2 of an unconnected network which contains two completely separated communities is equal to 0. While the CQ_2 of a clique is equal to 1. Fig.2 shows an evolving process of a dynamic network growing from a completely separated two-community structure to a clique as well as their corresponding CQ_2 values.

Recall that $\lambda_s = e^{-H_s/\varepsilon}$, we have $\log CQ_2 = -(H_2 - H_3)/\varepsilon = -\Delta H/\varepsilon$. That means the quality of a two-community structure will be exponentially improved or degenerated by increasing or decreasing the barrier between two communities. We can clearly demonstrate this by controlling the potential function shown in Example 1 (see Fig. 1).

Example 2. In this example, we assume that the network shown in Fig.1(b) contains two communities, (A B) and (C,D). The separatability of the two communities is decided by H_2. A large H_2 means a large separatability. The cohesion of each community is determined by H_3 and H_4. Smaller H_3 and H_4 imply better cohesion. So, we can regulate the two-community structure through modulating the gap between H_2 and H_3. Fig. 3 shows the experimental results.

In Fig.3, we can see that as $\Delta H = H_2 - H_3$ increases, the cohesion exponentially decreases, while the separateability increases very quickly. As a result, its two-community quality exponentially decreases and approaches to zero, which means the two-community structure exponentially becomes better as the potential gap increasing.

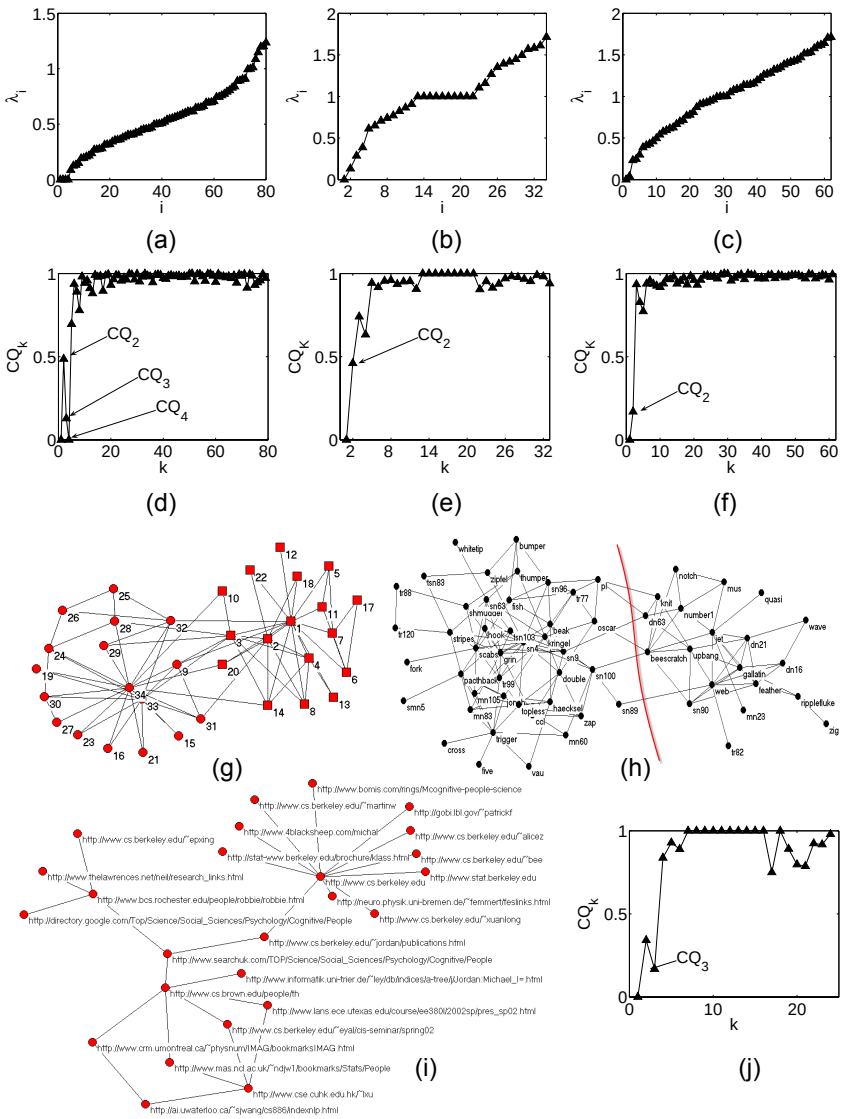

Fig. 4. (a)-(c) The spectra of the network in Fig.1 and two social networks. (d)-(f) The spectral signatures of the network in Fig.1 and two social networks. (g) The karate network. (h) The dolphin network. (i) A synthetic Web containing 25 Web pages. (j) The spectral signature of the synthetic Web.

3.2 Characterizing a K-community structure

Generally, we have n eigenvalues and intend to find the community number K. Let us enumerate the communities according to its corresponding eigenvalues. We can define the K-community quality (CQ_K) to measure how well-formed it is. For a well-formed K-community structure, each community should be cohesive, which means it is easy for the Markov chain to hit the Kth state, in which K communities local mixed, respectively. The hitting time of the Kth state should be early. On the other hand, communities should stand clear from each other, which means it is hard for the Markov chain to exit the Kth state by mixing them through only a few inter-community links. In other words, the mixing time should be long. So, there should be a big gap between $1/\lambda_K$ and $1/\lambda_{K+1}$. This point can be understood with the help of the potential function as given in Example 1. For a network with a K-community structure, in its corresponding potential function, there will be $K-1$ barriers much larger than others to separate K communities, i.e., $\cdots < H_{K+1} \ll H_K < H_{K-1} < \cdots < H_1 = \infty$. Recall that $\lambda_s = e^{-H_s/\varepsilon}$. We should have a significant gap between $1/\lambda_K$ and $1/\lambda_{K+1}$.

Definition 3. The normalized cohesion and the normalized separatability of the K-community structure of a given network are defined as:

$$NC_K = \frac{1/\lambda_{K+1}}{1/\lambda_K} = \frac{\lambda_K}{\lambda_{K+1}} \tag{10}$$

and

$$NS_K = \frac{1/\lambda_K - 1/\lambda_{K+1}}{1/\lambda_K} = 1 - \frac{\lambda_K}{\lambda_{K+1}} \tag{11}$$

Definition 4. In terms of NC_K or NS_K, the K-community quality (CQ_K) to evaluate how well-formed

a K-community structure of a given network is defined as:

$$CQ_K = NC_K = 1 - NS_K = \lambda_K/\lambda_{K+1} \quad (12)$$

It is easy to show $0 \leq CQ_K \leq 1$, and a small CQ_K implies a better community structure. If a K-community structure is well-formed, the barriers should be very large, and thus $\lambda_s(1 \leq s \leq K)$ should approach to zero. That indicates we can infer the number of communities by simply counting the number of eigenvalues close to zero. Fig.4(a) shows the spectrum of the network in Fig.1. As we see, four approximate zero eigenvalues correspond to four communities. In this case, CQ_4 is approximately equal to zero. For real networks with noises, we can use the following method to estimate the number of natural communities in a network.

$$K = \arg\min_k CQ_k \quad (13)$$

Definition 5. The *spectral signature* of a network is defined as the train of $CQ_k(1 \leq k < n)$.

Example 3. Fig.4(d) shows the spectral signature of the network shown in Fig.1, in which CQ_4 is the minimum except CQ_1 (CQ_1 is always zero). Figs.4(b)-(c) show the spectra of two well-known social networks. They are the Karate club network [32] (containing two actual communities denoted by circles and squares respectively) and the dolphin network [33] (containing two actual communities split by a solid line), respectively shown in Figs.4(g)-(h). Figs.4(e)-(f) show the spectral signatures of the two networks. In both cases, CQ_2 is the minimum value. Also, one can note that the CQ_2 value of the dolphin network is much smaller than that of the karate network, which means the modularity structure of dolphin network is much better than that of the karate network, as we intuitively observe from these two visualized networks. Fig.4(i) shows a small Web containing 3 Web communities. Fig.4(j) shows the spectral signature of this Web, in which CQ_3 is the minimum value.

3.3 Characterizing a hierarchical community structure

The information about the hierarchical community structure of a network can also be inferred from its spectral signature.

Example 4. Fig.5 gives the example networks with different community structures, respectively. For a network without a community structure, its spectral signature should look like the one shown in Fig.5(a), in which all $CQ_k(k > 1)$ will be very close to one. For a network with a single level of community structure containing exact $K(K > 1)$ communities, its spectral signature should look like the one shown in Fig.5(b), in which CQ_K will be approach to zero and all others (except CQ_1) will be very close to one. While, for a network with a hierarchical community structure on multiple scales, its spectral signature will look like the one shown in Fig.5(c), in which a number of $CQ_k(k > 1)$ will be approach to zero, and again all others (except CQ_1) will be very close to one. In the last case, the number of $CQ_k(k > 1)$ approaching to zero reveals the actual number of hierarchical levels hidden in a network, and furthermore, the significance of such levels can be quantified by their corresponding values of CQ_k. The partition of the whole network corresponding to the level with the largest CQ in its dendrogram will be the most significant in terms of finding most natural communities. For the network shown in Fig.5(c), it most likely contains two big communities, and each of which contains two moderate communities, and in turn each of which contains three small communities.

3.4 A framework for characterizing and mining communities

From the above analysis, we have uncovered the intrinsic connection between the community structure of a network and the spectrum of its corresponding Markov generator. For any given network, without clustering it using a particular algorithm, we can characterize and analyze its modularity by answering some critical questions related to its topological structures through observing and inferring its spectral signature. For examples:

1) Does a network have a well-formed community structure? $\Leftrightarrow$ Is the minimum CQ_K close to zero?
2) How many natural communities are hidden in a network? $\Leftrightarrow$ What is the position of the minimum CQ_K?
3) How do these communities stand clear from each other? $\Leftrightarrow$ How is NS_K close to one?
4) Are they close? $\Leftrightarrow$ Is the NC_K close to zero?
5) Does a network have a reasonable hierarchical community structure? $\Leftrightarrow$ Are there multiple CQ_K much smaller than others and approaching to zero?

Specifically, we can also answer the question of whether or not a network is stable. We believe that the modularity externally demonstrated by a social or biological network is essentially rooted at the stability of its corresponding social or biological system. For a very stable system, its network will contain an unique community. While, if a system is not stable, its network will demonstrate obvious modularity structure, in which different levels of dendrogram correspond to different metastable states of the system. So, CQ can also be used to characterize the stability of a real social or biological system. Larger CQ means less stability of such a system.

Therefore, like existing quantities, such as average path length, clustering coefficient, and degree distribution, a network's spectral signature could also be an important one to characterize the topological features of networks.

We have established a general framework for characterizing and analyzing communities for a given network. Now we can further extend it to a more general

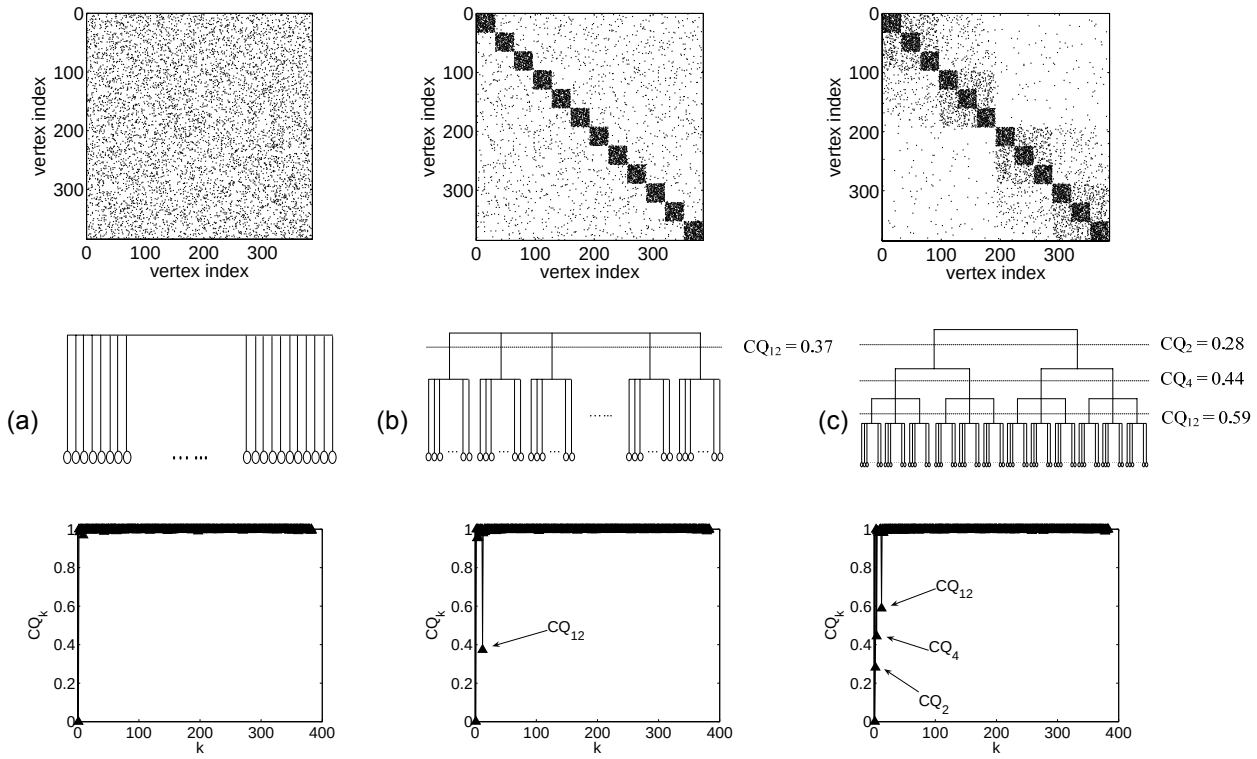

Fig. 5. (a) A network without a hierarchical structure, its dendrogram and its spectral signature. (b) A network with a single-level community structure, its dendrogram and its spectral signature. (c) A network with a multi-level community structure, its dendrogram and its spectral signature.

TABLE 1
A framework for characterizing and mining communities

1)	construct the transition matrix P of a given network;
2)	calculate the spectrum of $I - P$ ($\lambda_1 \leq \cdots \leq \lambda_n$);
3)	calculate its spectral signature ($CQ_1, \cdots, CQ_{n-1}$) and find the minimum $CQ_K (1 < K < n)$;
4)	characterize and analyze network' modularity with the above spectral signature;
5)	mine K communities from the matrix $P^{1/\lambda_{K+1}}$.

framework for mining communities with a hierarchical structure for a given network by inferring its spectral signature and one of its metastable states. Its main steps are summarized in Table 1.

For a network with a multi-level community structure, one can find all community structures on different levels by selecting the next minimum CQ_K close to zero and repeating step 5). In this way, a dendrogram, as the one shown in Fig.5(c), can be built.

Different strategies can be adopted to implement step 5) of the above framework. For example, one can define the similarity between vertices based on $P^{1/\lambda_{K+1}}$ since vertices within the same communities will have very similar row distributions. Then a similarity based clustering method can be used to find out all K communities.

In what follows, we will present an efficient implementation for the above framework to scalably mine communities, without the need of calculating eigvalues/eigenvectors and multiplying the transition matrix, which can be extremely expensive especially for processing very large-scale networks.

4 A Scalable Algorithm

4.1 The basic idea

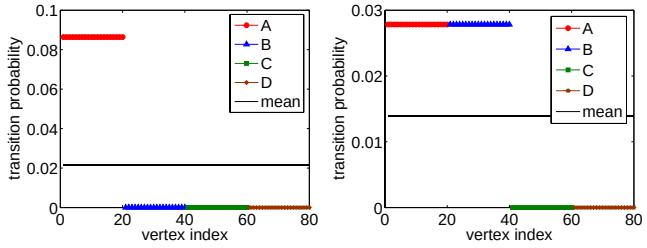

Fig. 6. Left: the 10th column distribution of the local mixing state at time T_4. Right: the 10th column distribution of the local mixing state at time T_2.

Our basic idea in implementing a scalable algorithm can be stated as follows: to cluster a network is to calculate and infer a single column distribution rather than to deal with the whole transition matrix. Each column distribution of a locally uniform transition matrix will also be locally uniform (the locally uniform property of metastability). For example, Fig.6 shows two column distributions of the transition matrices shown in Figs.1(e)

and (g). Based on the observation, we can develop an efficient implementation of the above framework to uncover all natural communities from a given network.

In this section, we will discuss how to infer communities from a single column distribution by addressing the following key questions: 1) how to select a column; 2) how to quickly calculate a column distribution at a local mixing time; and 3) how to infer communities from a single column distribution.

4.2 Column selection

For a practical application, we hope to identify all communities as well as their respective centers. From the perspective of random walk, the attractors (most stable states) of a Markov chain are likely the centers of respective communities because random walks within different communities will be attracted by them with big chance wherever they set out. So, we select the column corresponding to the most stable state of a network, which can be identified as follows:

$$c = \arg\max_i \pi_i = \arg\max_i \{\frac{d_i}{\sum_k d_k}\} = \arg\max_i \{d_i\} \quad (14)$$

4.3 Ordering time distribution (OTD)

Let $P_c^{(t)} = (p_{1,c}^{(t)}, \cdots, p_{n,c}^{(t)})^T$ be the cth column of the transition matrix P^t. It can be calculated by the following recursive equation:

$$p_{i,c}^{(t)} = \frac{1}{d_i} \sum_{\langle i,j\rangle \in E} A_{ij} \cdot p_{j,c}^{(t-1)} \quad (15)$$

Now the problem is how to estimate a suitable local mixing time t without calculating the spectrum of $I - P$. For the sake of speed, we hope to estimate the first local mixing time (denoted as T_K^{hit}), in which all communities mix together, through the metastability of local mixing states.

After a Markov chain enters its global mixing state, its transition matrix will keep fixed for ever. Correspondingly, after it goes into a local mixing state, its transition matrix will keep fixed temporarily until it exits this state (the temporarily fixed property of metastability). During that time, if one ranks all vertices according to one column distribution, the obtained sequence will keep stable until it leaves that local mixing state. Based on this property, T_K^{hit} can be estimated as follows.

Definition 6. Let S_t be the index sequence of all vertices sorted according to one of column distributions at time t. The difference between two sequential sequences is defined as $dif_t = nnz(S_t - S_{t-1})$, where $nnz(X)$ counts the number of non-zero elements of X. We have $T_K^{hit} \simeq T_{ord}$, where $dif_{T_{ord}} = 0$. T_{ord} is called *ordering time*, and correspondingly, $P_c^{(T_{ord})}$is called *ordering time distribution*(OTD).

Taking the network shown in Fig.1 as an example, The left panel of Fig. 7shows dif_t as time evolves. The middle one of Fig. 7 shows the transition matrix at T_{ord}, and the right one of Fig. 7 shows an ordering time distribution. Compared these figures with Fig.1(e) and Fig.6(a), we can see the estimated first local mixing time is quite close to the actual one.

Eq. (15) is a localized computation and can be calculated within $O(n + m)$ time for all i, where m denotes the number of links. So, the time to compute an OTD is bounded by $O(T_{ord}(n \log n + m))$ by considering additional sorting time. We have $T_{ord} \simeq T_K^{hit} = T_{K+1}^{ext} = \frac{1}{\lambda_{K+1}}$. So the time of OTD computing is bounded by $O((n \log n + m)/\lambda_{K+1})$.

In practice, we do not need to calculate a strict ordering time at which two sequential sequences are identical. Actually, we can stop iterations after dif_t is below a small threshold, which means there are only a few vertices within respective communities, which have not yet reached their final positions. While, these small fluctuations will not significantly affect computing accuracy.

4.4 Inferring a hierarchical community structure from OTD

One can infer clusters from an OTD using different strategies. This paper will adopt bipartition strategy. The most efficient way to bipartition an OTD is to divide it by its mean or by the biggest gap. For the OTD shown in Fig.7, community A can be precisely separated from others by both them. They work very well for networks with two-community structure such as a spatial point set shown in Fig.8 and a real network shown in Fig.9. While, in the cases of multiple communities, they may not find ideal hierarchy of communities. For the OTD given in Fig.7, both them can not find real hierarchy ((A,B), (C,D)). In order to find a reasonable hierarchical structure between communities, we bipartition an OTD with the aid of the concept of CQ_2.

Recall that, for a two-community structure, we have $CQ_2 = \lambda_2/\lambda_3 = e^{-(H_2-H_3)/\varepsilon} = e^{-\Delta H/\varepsilon} \propto P_\Delta$, where ΔH denotes the biggest barrier gap between two communities in its corresponding potential function, and P_Δ denotes the probability of escaping from one community to another by crossing over the barrier gap ΔH. So, it is reasonable to define a suitable P_Δ to estimate a real CQ_2 for practical use without calculating eigenvalues. In this paper, P_Δ is defined as:

$$P_\Delta^1 = \min_{V_1,V_2} \frac{1}{2}(EP(V_1, V_2) + EP(V_2, V_1)) \quad (16)$$

where (V_1, V_2) is a bipartition of $N = (V, E)$ and $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$. $EP(V_1, V_2)$ is the probability that a random walk can escape from V_1 in one step, which can be calculated by:

$$EP(V_1, V_2) = \frac{1}{|V_1|} \cdot \sum_{i\in V_1, j\in V_2} p_{ij} \quad (17)$$

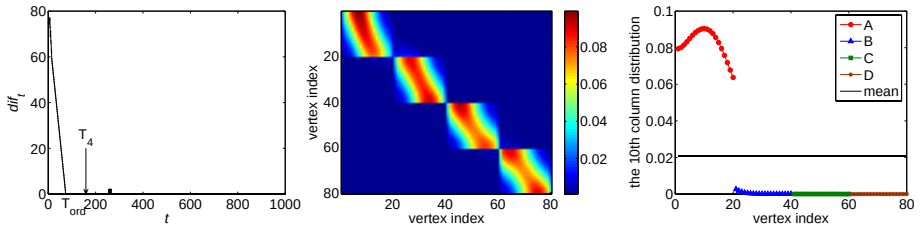

Fig. 7. Left: the trend of dif_t with time. Middle: the transition matrix at ordering time. Right: one ordering time distribution corresponding to the 10th column.

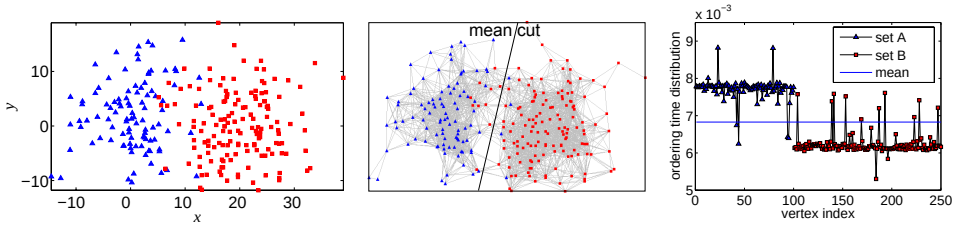

Fig. 8. Left: a point set generated by a mixed Gaussian distribution. Set A containing 100 points denoted by triangles is generated by a Gaussian distribution with mean 0 and variance 6, and set B containing 150 points denoted by squares is generated by a Gaussian distribution with mean 20 and variance 6. Middle: the network generated by adding weighted links between points according to the Gaussian similarity function $a_{ij} = exp(-\|x_i - x_j\|^2/\varepsilon)$ and removing the ones with weights close to zero. Right: the ordering time distribution in terms of a randomly selected vertex from set A. Using its mean, one can perfectly separate set A from B, as shown in the middle.

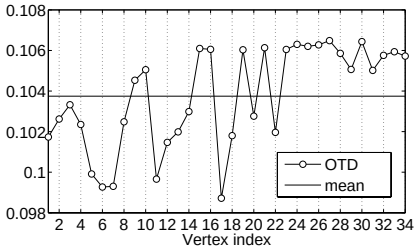

Fig. 9. The mean separates the karate network into two communities which are exactly same as reported by Wayne Zachary [32]. The vertices above the mean compose the community denoted as circles and those below the mean compose the community denoted as squares as shown in Fig.4(g).

The left panel of Fig.10 shows the relationship between a real $CQ_2 = \lambda_2/\lambda_3$ and an estimated one by P^1_Δ in terms of the potential function discussed in Example 1 by controlling the biggest gap between two communities denoted as $\Delta H = H_2 - H_3$. As the ΔH increases, it will get more and more difficult for a random walk to escape from (A,B) to (C,D) or vice versa by crossing over the barrier between them. From Eq.5 we have $P^1_\Delta \propto e^{-\Delta H/\varepsilon} = CQ_2$, as shown in Fig.10. The right panel shows a more general case, in which we regulate the two-community quality of a mixed Gaussian distribution by increasing the distance between two respective distributions. As we can see, both CQ_2 and its estimator P^1_Δ decrease with an approximately same trend as distance increasing. So, the smaller the P^1_Δ is, the better a two-community quality (CQ_2) will be.

Let vector X denote a bipartition of a network, where $X(i) = 1 \cdot I_{i\in V_1} + (-1) \cdot I_{i\in V_2}$. We have following proposition (refer to Appendix A for its proof):

Proposition 1:

$$P^1_\Delta = min_{Y^T 1=0} \frac{Y^T(I - D^{-1}A)Y}{2Y^TY} = \frac{Y^TQY}{2Y^TY} \qquad (18)$$

where $Y = (1-a)(\mathbf{1} + X) - a(\mathbf{1} - X)$ and $a = \frac{1}{n}\sum_{i=1}^n I_{\{X_i>0\}}$.

Now, to find a reasonable top-level hierarchy of a network is to find an X to minimize $\frac{Y^TQY}{Y^TY}$. One can find an optimal X by solving a Lagrange equation, which will be computationally expensive. Recall that an OTD is a locally uniform distribution in which entries corresponding to vertices in the same communities will have similar values. With the sorted OTD according to such values, we can greatly simplify this task by reducing the original search space with size 2^n into a candidate set with a size n, in which each candidate takes the following form: $X_x(i) = 1 \cdot I_{i'\le x} + (-1) \cdot I_{i'>x}$, where i is the index in the original OTD, and i' is its corresponding index in the sorted OTD and $1 \le x = |V_1| \le n$. So, an optimal x can be calculated by:

$$x^* = \arg \min_{1\le x\le n} \frac{Y_x^T(I-P)Y_x}{Y_x^TY_x} = \frac{Y_x^TQY_x}{Y_x^TY_x} \qquad (19)$$

where $Y_x = (1-x)(\mathbf{1}+X_x) - x(\mathbf{1}-X_x)$. We have proven the following proposition (see Appendix B).

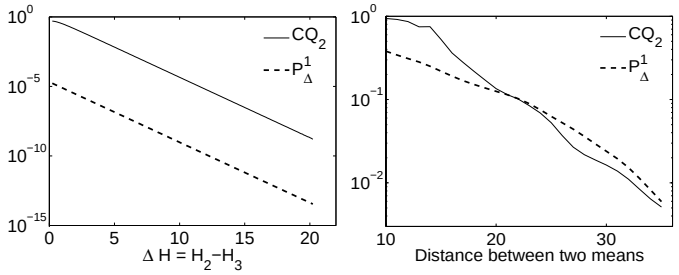

Fig. 10. The relationship between CQ_2 and P^1_Δ in terms of the potential function shown in Fig.1(a) and the mixed Gaussian distribution shown in Fig.8.

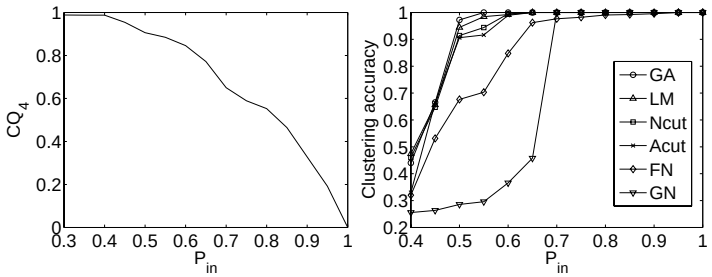

Fig. 11. Left: The CQ_4 values of networks with different P_{in}. Right: the clustering accuracy of six algorithms.

TABLE 2
The LM algorithm

1)	select an attractor with the maximum degree;
2)	calculate an OTD regarding to this attractor;
3)	calcualte an optimal bipartition in terms of P^1_Δ based on the sorted OTD;
4)	return if $P^1_\Delta > \alpha$; otherwise
5)	bipartition the network and recursively manipulate two sub-networks.

Proposition 2: x^* with P^1_Δ can be calculated within $O(n \log n + m)$ time in terms of the linked-adjacency list of P.

As discussed before, networks with large P^1_Δ will have large CQ_2, which means there is ambiguous or no community structure in it. So, we can stop the recursive bipartition process using the following criterion: $P^1_\Delta > \alpha$, where α is a predefined positive threshold, which aims to regulate the granularity of communities. Larger α will obtain more communities with fine granularity. Otherwise, fewer communities with a coarse granularity.

4.5 The LM algorithm

Table 2 summarizes the main steps of the proposed implementation of the framework in section 3.4, taking the adjacency matrix of a network as its input.

From previous discussions, the time required in the first four steps is bounded by $O((n \log n + m)/\lambda_{K+1})$. So the worst time of the LM is equal to multiplied by the total number of recursively callings by step 5. For finding out all K clusters, exactly $2K - 1$ recursive callings are required. Therefore, the time of LM is bounded by $O(K(n \log n + m)/\lambda_{K+1})$. Recall that λ_{K+1} is the hitting time of the $K-th$ metastable state, and will be decided by the cohesion of each cluster rather than the scale of network. The scalability of the LM will be demonstrated in Section 5.2.

5 EXPERIMENTS

In this section, we will test the performances of the LM algorithm. We have designed and implemented experiments oriented toward three main Objectives:

1) To evaluate the accuracy of the LM;
2) To test its actual runtime;
3) To apply it to real-world and large-scale networks.

5.1 Evaluating the accuracy of the LM

5.1.1 Using synthetic networks

We have compared the accuracy of the LM with five most well-known algorithms, Including: the GN algorithm [1], the FN algorithm [22], two spectral methods (the Ncut algorithm [6] and the Acut algorithm [5]), and the GA algorithm [11] in terms of a widely used random network model, which can produce a randomly synthetic network containing 4 predefined communities and each of them contains 32 vertices. The average degree of vertices is 16, and the ratio of intra-community links is denoted as P_{in}. As P_{in} decreases, the community structures of such synthetic networks become more and more ambiguous, and correspondingly, their CQ_4 values climb from 0 to 1, as shown in the left panel of Fig.11. Communities are considered to be correctly discovered if all vertices are clustered into four original groups.

Fig. 11 presents the experimental results, in which y-axis denotes the fraction of vertices correctly clustered by respective algorithms, and each point in curves is obtained by testing them against 200 synthetic networks sampled from above model. As we can see, all algorithms work very well when P_{in} is more than than 0.7 ($CQ_4 < 0.65$) with clustering accuracy more than 95%. Compared with other five algorithms, the LM performs quite well and its accuracy is only slightly worse than that of the GA in the case of $0.5 \leq P_{in} \leq 0.6$.

5.1.2 Using benchmark networks

We have tested the LM against some widely used benchmark networks, including karate club network [32], dolphin network [33], and football association network [1]. The LM obtained quite good results for all cases. Please refer to Appendix C for details.

5.2 Scalable runtime of the LM

In theory, the time complexity of the LM is bounded by $O(K(n \log n + m)/\lambda_{K+1})$. As discussed before, λ_{K+1} is decided by the cohesion of communities and is not sensitive to the scales of networks. In practice, sometimes

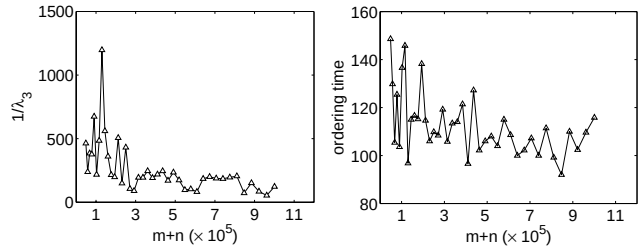

Fig. 12. $1/\lambda_3$ and ordering time decay as networks expand.

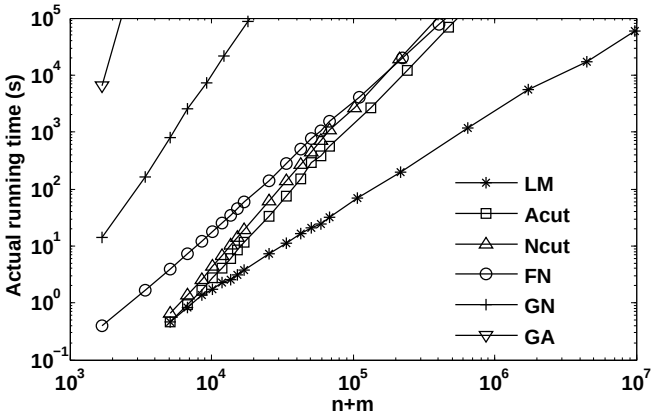

Fig. 13. The actual runtime of six algorithms with respect to network scales.

λ_{K+1} will decay as networks expand. The reason lies that communities will become much denser when more vertices keep joining in them, and consequently the cohesion of communities keeps improving. Fig.12 shows this interesting property using the mixed Gaussian distribution model shown in Fig.8. In this example, we fix the mean and variance of each distribution, while keep increasing the number of points. We can observe that $1/\lambda_3$ as well as its estimator, the corresponding ordering time, decays during this expanding course.

Fig.13 shows the actual runtime of six algorithms with respect to different network scales. In this experiment, all algorithms are run on a workstation with a 2GHz CPU and a 2GB memory. The operating system was Windows XP, and the simulation software was programmed by Matlab 7.0. Given an n, a sparse network with $m = O(n)$ and a predefined multiple community structure will be randomly generated for testing the speed of different algorithms. As we see, 1) the LM is much faster than others when the scale of network is larger than 10^5; and 2) the runtime of the LM is scalable to the scales of networks, and it can efficiently manipulate a very large network with size of $n+m = O(10^7)$ within $O(10^4)$ seconds.

5.3 Experiments with two large-scale networks

5.3.1 A semantic network

A semantic network [2] contains 7207 phrases and 31784 edges, as shown in Fig. 14(a). The weights of edges are calculated in terms of phrase co-occurrences. For visualization purpose, the LM outputs a transformed adjacency matrix (in which the vertices within the same communities will be arranged together) with a hierarchical community structure. The output matrix is shown in Fig. 14(b), and its corresponding hierarchical structure is given in Fig. 14(c). The distribution of community sizes is shown in Fig. 14(d). Totally, 573 communities are detected by setting $\alpha = 0.5$, the maximum size is 139, the minimum size is 2, and the average size is 12.57. One can see an approximate power-law phenomenon: that is, most communities are small and only a few are big. From them, we have selected four interesting communities listed as follows:

Community 1 = {Violin, Instrument, Cello, Band, Tuba, Clarinet, Orchestra, Trumpet, Trombone, Oboe, Woodwind, Symphony, Flute, Bass, Viola, Fiddle};

Community 2 = {Opera, Alto, Chorus, Voice, Psalm, Hymn, Choir, Singer, Song, Tenor, Sing, Soprano, Fat lady};

Community 3 = {Ovation, Sitting, Low, Descent, Up, Step, Ascend, Elevator, Ascent, Staircase, Stairwell, Climb, Steps, Ladder, Stairs, Wake, Stairway, Rise, Escalator, Stair, Down, Standing, Resting};

Community 4 = {Nails, Hammer, Carpenter, Screw, Screwdriver, Tool, Pliers, Wrench, Sickle, Mechanic, Phillips}.

5.3.2 A scientific collaboration network

Fig.15(a) shows a scientific collaboration network [4], which codes the research collaborations among 56276 physicists in terms of their coauthored papers posted on the Physics E-print Archive at arxiv.org. The weights of edges between physicist are proportional to the numbers of papers coauthored by them. Totally, this network contains 315810 weighted edges. The output matrix with a hierarchical structure is given in Figs.15(b) and (c). From the transformed matrix of Figs.15(b), one can observe a quite strong community structure, or a group-oriented collaboration pattern, among these physicists, in which three biggest research communities are self-organized regarding to three main research fields: condensed matter, high-energy physics (including theory, phenomenology and nuclear), and astrophysics. The distribution of community sizes is shown in Fig. 15(d). Totally, 843 communities are detected by setting $\alpha = 0.5$, the maximum size is 199, the minimum size is 2, and the average size is 67.

6 Discussions

6.1 A comparison with spectral methods

In the literature, a large body of works have dedicated their efforts to partitioning a graph by calculating the eigenvectors of its Laplacian matrix, normalized Laplacian matrix or other variants. However, the work of this paper is completely distinct from the existing spectral graph partition methods in three main aspects.

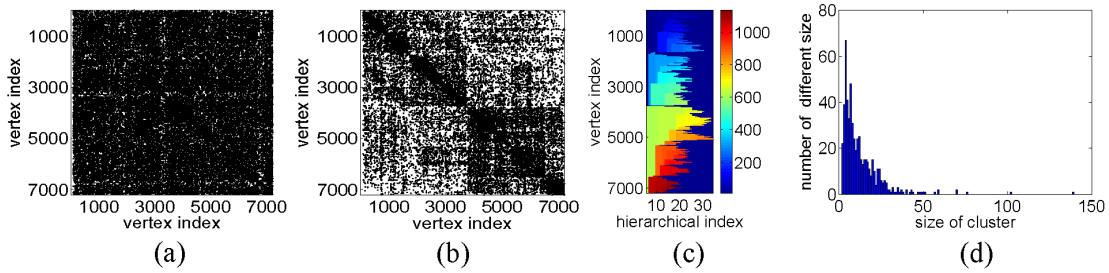

Fig. 14. Results of applying the LM to a semantic network. (a) The initial adjacency matrix of a semantic network. (b)-(c) The output of the LM, a transformed adjacency matrix plus a hierarchical community structure. (d) The statistics of all identified communities.

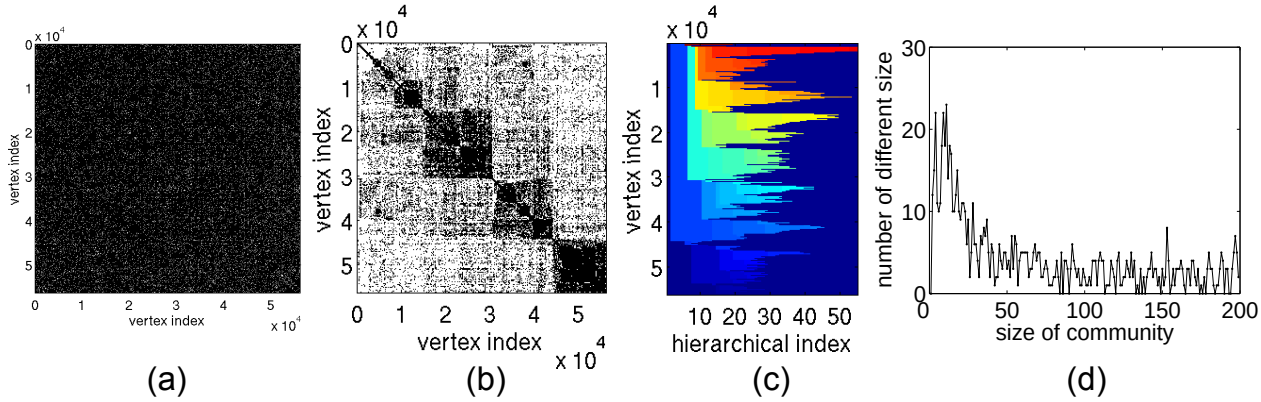

Fig. 15. The output of the LM algorithm against a scientific collaboration network of 56,276 physicists.

Firstly, spectral graph partition methods are essentially optimization based methods, which try to optimize different kinds of predefined 'cut' criteria, such as 'average cut [5]', 'normalized cut [6]', 'ratio cut [34]', and others with specific constraints like [8], [9]. Based on the matrix theory, spectral methods transform those tasks into kinds of constraint quadratic optimization problems, and their approximately optimal solutions can be estimated by the second smallest eigenvector of different versions of the Laplacian matrices. While, in this paper, we try to uncover the intrinsic connections between network community structures and networks' spectral properties by inferring the dynamics of a stochastic process based on the local mixing theory and the large deviation theory.

Secondly, most of the existing spectral methods utilize the connections between networks'eigenvectors and their optimal partitions, but rarely discuss or make clear the deep meaning of networks' eigenvalues with the implications to network modularity. While, this work has taken much effort to discover such hidden connections.

Finally, the rationales behind spectral methods and the LM is completely distinct from the viewpoint of practical calculation. Spectral methods partition graphs into K communities by first calculating the smallest K eigenvectors of the Laplacian matrices, which generally takes $O(n^3)$ time or $(K \cdot \frac{n+m}{\lambda_3-\lambda_2})$ by using some spectral techniques, such as the Lanczos methods [35], and then clustering n K-dimension vectors into K clusters with the K-means method, which will cost $O(InK^2)$ time, where I denotes the number of iterations required by K-means to converge. On the other hand, the LM discover all K communities by first calculating the OTD, one column distribution of a transition matrix, within a time of $O(\frac{n \log n+m}{\lambda_{K+1}})$, and then split it by its means with a time of $O(n)$ or, more precisely, by optimizing a predefined P_Δ^1 within a time of $O(n+m)$. As shown in Fig.13, the LM is much efficient than spectral methods in practice.

6.2 A comparison with Newman's modularity

Also, we should emphasize the distinctions between two quantities, the CQ proposed in this paper and the Q proposed by Newman [7]. The Q is a widely used criterion for evaluating a specific partition scheme of a network, which is defined as "the fraction of edges that fall within communities, minus the expected value of the same quantity if edges fall at random without regard for the community structure" [22]. Different partitions will get different Q values for the same network, and larger ones mean better partitions in terms of community structure. On the other hand, the CQ tries to characterize and evaluate the modularity of networks themselves based on networks' spectra, rather than a specific network partition based on a predefined function. Therefore, a network has exactly only one CQ value regardless of how many partition schemes it would have, and the smaller the CQ the better the network modularity. With

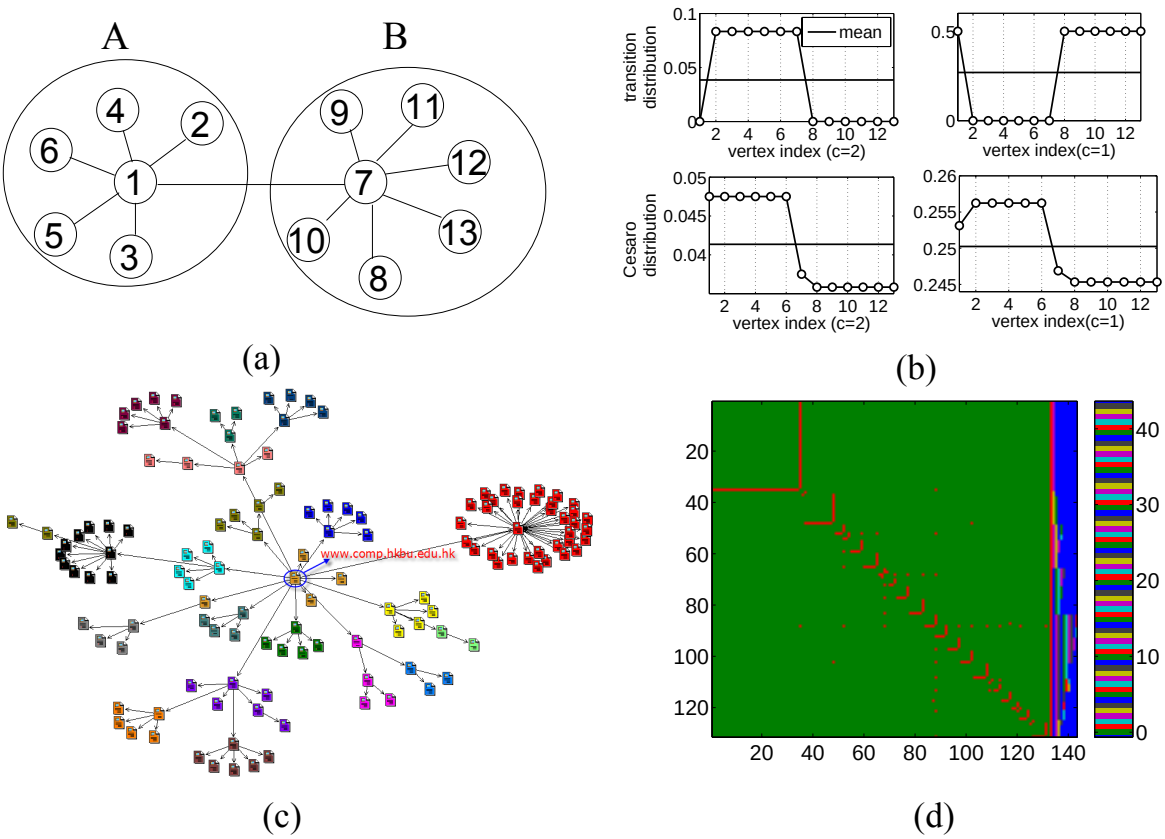

Fig. 16. (a) A simple hub-style network, in which community A(1,2,3,4,5,6) and community B(7,8,9,10,11,12,13) are, respectively, formed around two hubs, vertices 1 and 7. (b) Two types of transition distributions of the hub-style network with selecting different attractors. (c) A hub-style Web-page network containing 131 pages, which is obtained by a Web crawler called websphinx with parameters: "root = www.comp.hkbu.edu.hk", "hops = depth first", and "depth = 3". Web pages are colored according to their respective cluster memberships. (d) The output matrix of the LM with calculating C-OTD, in which the rightmost gray degree bar denotes the hierarchical community structure.

CQ, we can quantitatively compare the modularity of different types of complex networks.

6.3 Improvement and limitation of the LM

In order to scalably mining communities from large-scale networks, we have developed the LM algorithm, which actually is an approximate implementation of the framework proposed in Section 3.4. Through experiments, the LM demonstrates its good performance in both effectiveness and efficiency. However, here we still need to point out its main improvement and limitation worth being noted.

6.3.1 An improvement

The current version of the LM is only suitable for clustering strongly-ergodic networks, whose Markov chains are irreducible and aperiodic. Most real networks are aperiodic because they contain many triangle-like loops due to their high clustering coefficient. But in some cases, we are also interested in finding the community structure from weakly-ergodic networks, whose Markov chains are irreducible and periodic, such as the hub-style network shown in Fig.16(a). Actually, such hub-style networks widely exist and are the basic building blocks of some huge scale-free networks, such as World Wide Web as shown in Fig.16(c). The LM can be easily improved to be suited for weakly-ergodic networks.

Let $\overline{\Pi} = (\overline{\pi}_1, \cdots, \overline{\pi}_n)$ be the $Cesàro$ average distribution of a Markov chain, where $\overline{\pi}_j = \lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t} p_{ij}^{(k)}$. It is has been provn that $\overline{\Pi}$ exists if the chain is weakly ergodic. Thus, instead of the OTD by Eq.15, we calculate the C-OTD by:

$$\overline{p_{i,c}^{(t)}} = \frac{1}{t} \sum_{k=1}^{t} p_{i,c}^{(k)} \tag{20}$$

Fig.16(b) shows two types of distribution of the hub-style network shown in Fig.16(a). The top trace shows two original transition distributions corresponding two different attractors. In both cases, two abnormal communities (2,3,4,5,6,7) and (1,8,9,10,11,12,13) are detected, in which both hubs are misclassified due to the non-existence of the limit distribution in such a weakly-ergodic network. The bottom trace shows two $Cesàro$ average distributions corresponding to two different

attractors. In this case, community A(1,2,3,4,5,6) and community B(7,8,9,10,11,12,13) are correctly detected.

The community structure of the weakly-ergodic network in Fig.16(c) was discovered by the LM with calculating C-OTD, as shown in Fig.16(d), in which total 23 communities are detected with $\alpha = 0.5$. Based on the output matrix, the original Web-page network was colored so that different gray degree denotes different cluster, as shown in Fig.16(c). We have also tested the improved LM algorithm against all networks as discussed in this paper and the obtained results are identical to or quite similar as those obtained by the original LM.

6.3.2 A limitation

The LM is not parameter-free. As discussed in Section 3, we can reasonably estimate community number by spectral signatures. While, in order to avoid computing eigenvalues, a very expensive calculation especially for very large matrices, the LM adopts a very economical way, a predefined criterion $P_\Delta^1 > \alpha$, to stop recursive partitions. The parameter α is used to regulate the granularity of communities. Generally speaking, larger α will result in more communities with a fine granularity. Otherwise, fewer communities with a coarse granularity. While, we have not yet found a theoretical guide to optimally set parameters for different real-world networks.

7 Conclusions

This work has uncovered the connection between network modularity and network's spectrum properties, and proposed the concept of network's spectral signature. One can infer a lot of important information related to community structure from a network's spectral signature, such as the quality of modularity, the cohesion and separatability of communities, the number of communities, and the hierarchical structure of communities. Based on the concept of spectral signature, this work has presented a theoretical framework for characterizing, analyzing, and mining communities of a given network by means of inferring its spectral signature and one of its metastable states. Utilizing the basic properties of metastability, i.e., locally uniform and temporarily fixed, a scalable implementation for this framework, called LM algorithm, has been proposed that is oriented toward practical applications. Its time complexity in theory has been analyzed and its performances in practice including effectiveness and efficiency have been demonstrated and verified using different types/scales of networks.

Appendix A
Proof for Proposition 1

Let $X(i) = 1 \cdot I_{\{i \in V_1\}} + (-1) \cdot I_{\{i \in V_2\}}$ and $R = EP(V_1, V_2) + EP(V_2, V_1) = \frac{1}{|V_1|} \sum_{i \in V_1, j \in V_2} p_{ij} + \frac{1}{|V_2|} \sum_{i \in V_2, j \in V_1} p_{ij}$, we have:

$R = \frac{\sum_{x_i>0, x_j<0} \frac{-A_{ij}}{d_i} x_i x_j}{\sum_{x_i>0} x_i} + \frac{\sum_{x_i<0, x_j>0} \frac{-A_{ij}}{d_i} x_i x_j}{\sum_{x_i<0} -x_i}$

$= \frac{(\mathbf{1}+X)^T(I-P)(\mathbf{1}+X)}{(\mathbf{1}+X)^T(\mathbf{1}+X)} + \frac{(\mathbf{1}-X)^T(I-P)(\mathbf{1}+X)}{(\mathbf{1}-X)^T(\mathbf{1}-X)}$

Let $k = \frac{1}{n} \sum_{i=1}^n I_{\{x_i>0\}}$, we have

$(\mathbf{1}-X)^T(\mathbf{1}-X) = \frac{1-k}{k}(\mathbf{1}+X)^T(\mathbf{1}+X)$ and

$R = \frac{(\mathbf{1}+X)^T(I-P)(\mathbf{1}+X)}{(\mathbf{1}+X)^T(\mathbf{1}+X)} + \frac{(\mathbf{1}-X)^T(I-P)(\mathbf{1}+X)}{\frac{1-k}{k}(\mathbf{1}+X)^T(\mathbf{1}+X)}$

$= \frac{X^T(I-P)X + (1-2k)\mathbf{1}^T(I-P)X + (1-2k)X^T(I-P)\mathbf{1}}{(1-k)(\mathbf{1}+X)^T(\mathbf{1}+X)}$

Due to $\mathbf{1}^T(I-P)\mathbf{1} = 0$, we have

$R = \frac{(X+\mathbf{1}-2k\cdot\mathbf{1})^T(I-P)(X+\mathbf{1}-2k\cdot\mathbf{1})}{(1-k)(\mathbf{1}+X)^T(\mathbf{1}+X)}$

$= \frac{((1-k)(\mathbf{1}+X)-k(\mathbf{1}-X))^T(I-P)((1-k))-k(\mathbf{1}-X))}{((1-k)(\mathbf{1}+X)-k(\mathbf{1}-X))^T((1-k)(\mathbf{1}+X)-k(\mathbf{1}-X))}$

Let $Y = (1-k)(\mathbf{1}+X) - k(\mathbf{1}-X)$, we have $R = \frac{Y^T(I-P)Y}{Y^TY}$.

Furthermore, we have

$Y^T\mathbf{1} = ((1-k)(1+x_1) - k(1-x_1), \cdots, (1-k)(1+x_n) - k(1-x_n)) \cdot \mathbf{1}$

$= 2(1-k)\sum_{i=1}^n I_{\{x_i>0\}} - 2k\sum_{i=1}^n I_{\{x_i<0\}}$

$= 2((1-k)\sum_{i=1}^n I_{\{x_i>0\}} - k(n - \sum_{i=1}^n I_{\{x_i>0\}}))$

$= 2(\sum_{i=1}^n I_{\{x_i>0\}} - kn)$

$= 2(\sum_{i=1}^n I_{\{x_i>0\}} - \frac{\sum_{i=1}^n I_{\{x_i>0\}}}{n} \cdot n$

$= 0$

So, we have

$P_\Delta^1 = min_{V_1,V_2} \frac{1}{2} R = min_{Y^T\mathbf{1}=0} \frac{Y^T(I-P)Y}{2Y^TY} = min_{Y^T\mathbf{1}=0} \frac{Y^TQY}{2Y^TY}$.

Appendix B
Proof for Proposition 2

The routine for calculating x^* and P_Δ^1 is given as follows:

```
[P_Δ^1, x*] = PCQ2(P,OTD)
n = length(P);
[i,j,v] = find(P);
m = length(v);
% transform P according to the sorted OTD
[OTD, IX] = sort(OTD);
for r = 1:n
    map(IX(r)) = r;
end
for r = 1:m
    I(r)=map(i(r)); J(r)=map(j(r)); V(r)=v(r);
end
% top-down scanning
S1 = zeros(n,1); T1 = zeros(n,1);
for r = 1:m
    if I(r)<=J(r)
      S1(I(r)) = S1(I(r)) + V(r);
    else
      S1(J(r)) = S1(J(r)) + V(r);
    end
end
for r = 2:n
    S1(r) = S1(r-1)+S1(r); T1(r) = S1(r)/r;
end
T1 = T1(1:n-1);
% bottom-up scanning
S2 = zeros(n,1); T2 = zeros(n,1);
for r = m:-1:1
    if J(r)<=I(r)
```

```
        S2(J(r)) = S2(J(r)) + V(r);
    else
        S2(I(r)) = S2(I(r)) + V(r);
    end
end
for r = n-1:-1:1
    S2(r) = S2(r+1)+S2(r); T2(r) = S2(r)/(n-r+1);
end
T2 = T2(2:n);
% computing the P_Δ^1
for x =1:n-1
    PD(x) = 0.5*(2-T1(x)-T2(x));
end
[P_Δ^1, x*] = min(PD);
```

P is stored as a sparse matrix. n=length(P) is the number of vertices, and m=length(v) is the number of edges. It is easy to see the time of above codes is bounded by the $O(n \log n + m)$.

APPENDIX C
EXAMPLES OF SOCIAL NETWORK MINING

Example C.1 Fig.17 presents the output of the LM against the karate club network [32], in which three communities are detected with $\alpha = 0.5$. The top-level hierarchy detected by the LM is exactly identical with the actual division of the karate club. In addition, the LM predicts a potential division of the group led by its administrator.

Example C.2 Fig.18 shows the output of the LM against the dolphin network [33], in which five communities are detected with $\alpha = 0.5$. The top level hierarchy is identical with the actual division except the vertex "sn89", which respectively has one link with two different groups. In addition, the LM also predicts some further divisions of the two biggest groups possibly happened in the future.

Example C.3 Fig.19 shows the output of the LM against football association network [1] with $\alpha = 0.5$. The string on the bottom of each column denotes the team no., team name and association no.. For example, the string "115:Hawaii:11" denotes that the name of team 115 is Hawaii and it is one team of association 11. The tree at the bottom of the matrix denotes the hierarchy of communities. Its 12 leaves respectively correspond to 12 football associations. Compared with the real organization of total 12 associations, 8 teams are misclassified due to their more inter-association matches.

ACKNOWLEDGMENTS

Jiming Liu is the corresponding author. The work was funded by the National Natural Science Foundation of China grants 60503016,60573073, the National High-Tech Research and Development Plan of China under Grant 2006AA10Z245, and the Major State Basic Research Development Program of China (973 Program) 2003CB317001.

REFERENCES

[1] M.Girvan, M.E.J.Newman.Community Structure in Social and Biological Networks. *Proc Natl Acad Sci USA*, 2002, 9(12):7821-7826.
[2] G.Palla, I.Derenyi, I.Farkas, T.Vicsek. Uncovering the overlapping community structures of complex networks in nature and society. *Nature*, 2005, 435(7043): 814-818.
[3] G.Palla, A.L.Barab?si, T.Vicsek. Quantifying Social Group Evolution. *Nature*, 2007, 446(7136), 664-667.
[4] M.E.J.Newman. Coauthorship networks and patterns of scientific collaboration. *Proc Natl Acad Sci USA*, 2004, 101(s1):5200-5205.
[5] M.Fiedler. Algebraic Connectivity of Graphs. *Czechoslovakian Math J*, 1973, 23:298-305.
[6] J.Shi, J.Malik. Normalized Cuts and Image Segmentation. *IEEE Tans. On Pattern Analysis and Machine Intelligent*, 2000, 22(8):888-904.
[7] M.E.J.Newman. Modularity and community structure in networks. *Proc Natl Acad Sci USA*, 2006, 103(23):8577-8582.
[8] S.White, P.Smyth. A spectral clustering approach to finding communities in graphs. In Proceedings of the 5th SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, Philadelphia, 2005.
[9] M.Shiga, I.Takigawa, H.Mamitsuka. A spectral clustering approach to optimally combining numerical vectors with a modular network. In Proceedings of the 13th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining ,San Jose, California, United States, August 12-15, 2007. ACM Press, New York, NY, 647-656.
[10] D.M.Wilkinson, B.A.Huberman. A method for finding communities of related genes. *Proc Natl Acad Sci USA*, 2004, 101(suppl 1):5241-5248.
[11] R.Guimera, L.A.N.Amaral. Functional cartography of complex metabolic networks. *Nature*, 2005, 433(2): 895-900.
[12] E.Ravasz, A.L.Somera, D.A.Mongru. Hierarchical organization of modularity in metabolic networks. *Science*, 2002,297(5586): 1551-1555.
[13] V.Farutin, K.Robison, E.Lightcap, V.Dancik, A.Ruttenberg, S.Letovsky, J.Pradines. Edge-Count Probabilities for the Identification of Local Protein Communities and Their Organization. *Proteins: Structure, Function, and Bioinformatics*, 2006,62(3):800-818.
[14] B.Snel, P.Bork, M.A. Huynen. The identification of functional modules from the genomic association of genes. *Proc Natl Acad Sci USA*, 2002, 99(9):5890-5895.
[15] Z.Wang, J.Zhang. In Serach of the Biological Significance of Modular Structures in Protein Networks. *PLOS Computational Biology*, 2007, 3(6):e107.
[16] G.W.Flake, S.Lawrence, C.L.Giles, F.M.Coetzee. Self-Organization and Identification of Web Communities. *IEEE Computer*, 2002, 35(3):66-71.
[17] J.M.Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of ACM*, 1999, 46(5):604-632.
[18] M.I.Fredlin, A.D.Wenzell. *Random perturbations of dyanmical systems*, Springer-Verlag, New York, 1984.
[19] J.P.Eckmann, E.Moses. Curvature of co-links uncovers hidden thematic layers in the World Wide Web. *Proc Natl Acad Sci USA*, Vol.99, No.9, 2002, pp.5825-5829.
[20] H.Ino, M.Kudo, A.Nakamura. Partitioning of Web graphs by community topology. In Proceedings of the 14th international Conference on World Wide Web, Chiba, Japan, May 10-14, 2005, ACM Press, New York, NY, 661-669.
[21] B.W.Kernighan, S.Lin. An Efficient Heuristic Procedure for Partitioning Graphs. *Bell System Technical*, 1970, 49:291-307.
[22] M.E.J.Newman. Fast Algorithm for Detecting Community Structure in Networks. *Physical Rev. E*, 2004, 69(6):066133.
[23] J.Duch, A.Arenas. Community detection in complex networks using extreme optimization. *Physical Review E*, 2005, 72:027104.
[24] J.M. Pujol, J. B?jar, J. Delgado. Clustering algorithm for determining community structure in large networks. *Phys. Rev. E*, 2006,74:016107.
[25] J.Reichardt, S.Bornholdt. Detecting fuzzy community structures in complex networks with a potts model. *Phys. Rev.Let.*, 2004, 93(19):218701.
[26] A.Clauset,C.Moore,M.E.J Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 2008, 453(5):98-101.
[27] F.Wu, B.A.Huberman. Finding Communities in Linear Time: A Physics Approach. *European Physical J. B*, 2004, 38(2):331-338.

[28] F.Radicchi, C.Castellano, F.Cecconi, V. Loreto, D.Parisi. Defining and Identifying Communities in Networks. *Proc Natl Acad Sci USA*, 2004,101(9): 2658-2663.
[29] B.Yang, W.K.Cheung, J. Liu. Community Mining from signed Social Networks. *IEEE Transactions on Knowledge and Data Engineering*, 2007,19(10):1333-1348.
[30] J.Scott. *Social Network Analysis: A Handbook*, 2nd ed, Sage Publications, London, 2000.
[31] M.I.Fredlin, A.D.Wenzell. *Random perturbations of dyanmical systems*, Springer-Verlag, New York, 1984.
[32] W.W.Zachary. An Information Flow Model for Conflict and Fission in Small Groups. *J. Anthropological Research*, 1977, 33:452-473.
[33] D.Lusseau. The Emergent Properties of a Dolphin Social Network. *Proc Biol Sci*, 2003, 270(Suppl 2):S186-8.
[34] Y.C. Wei and C.K. Cheng. Ration cut partitioning for hierarchical designs, *IEEE Trans. Computer-Aided Design*, 1991, 10(7):911-921.
[35] G.H.Golub, C.F.V.Loan. *Matrix Computations*, Johns Hopkins Univ. Press, Baltimore, Md., 1989.

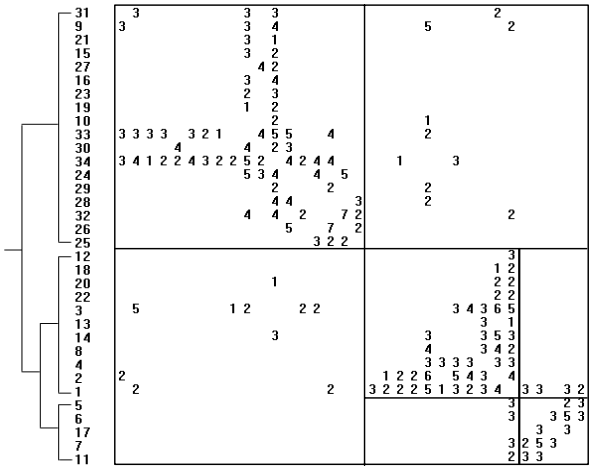

Fig. 17. The output of the LM algorithm against the karate club network.

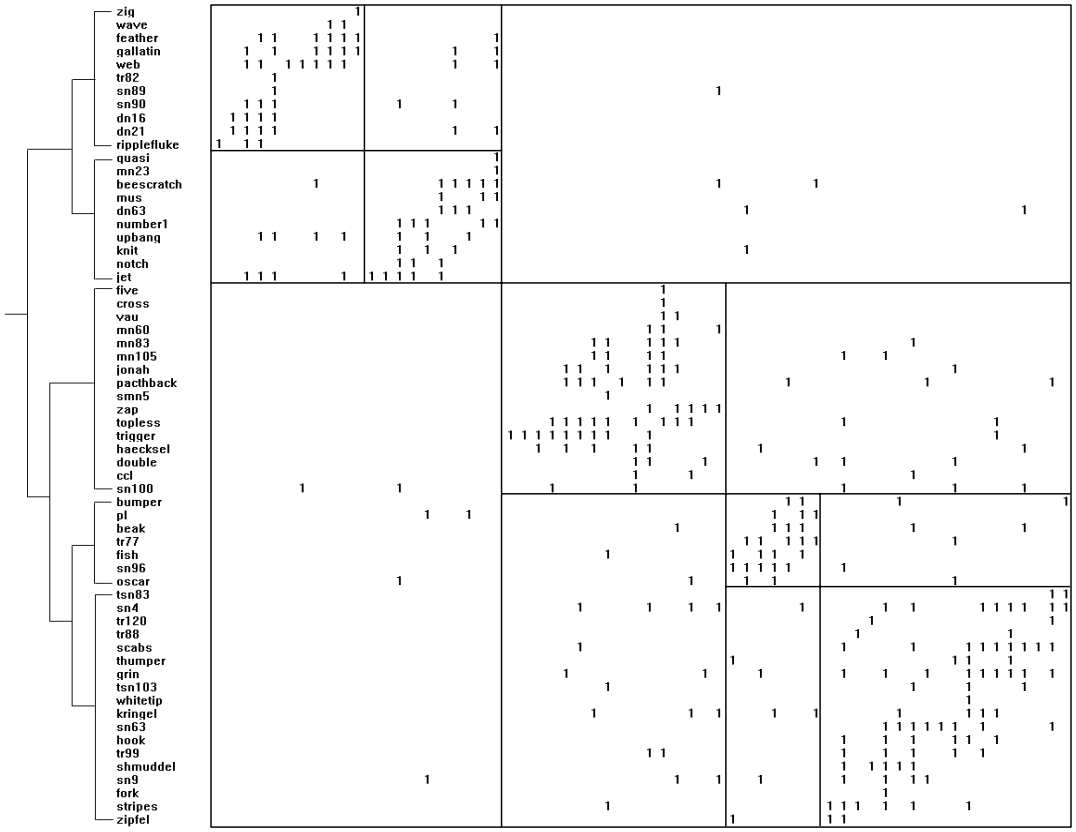

Fig. 18. The output of the LM algorithm against the dolphin network.

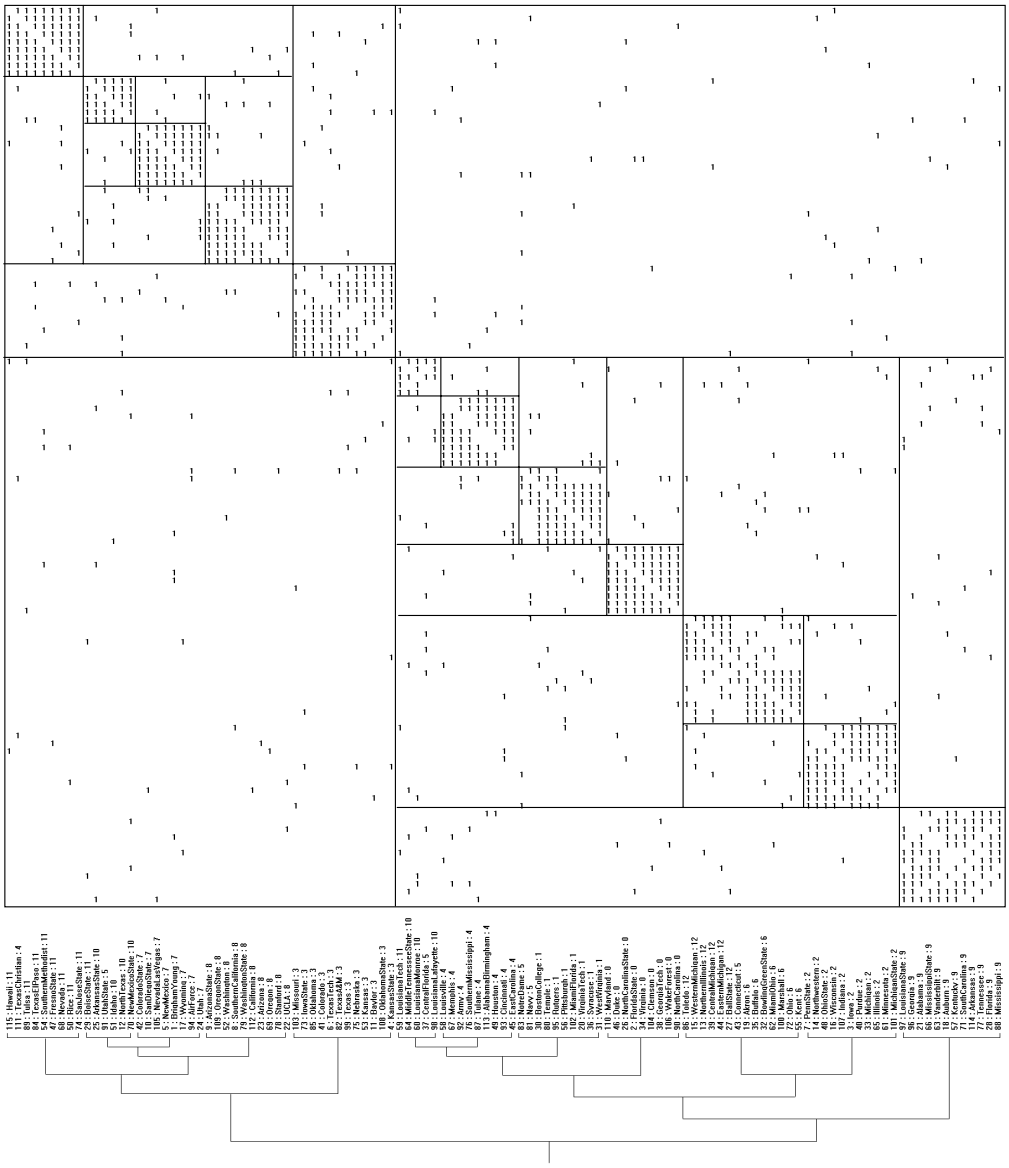

Fig. 19. The output of the LM algorithm against the network of US college football association in 2000 season.