On the Spectral Characterization and Scalable Mining of Network Communities

Bo Yang, Jiming Liu, and Jianfeng Feng

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 for short) is concerned with the problem of finding all such communities from a given network. A wide variety of applications can be formulated as NCMPs, ranging from social and/or biological network analysis to web mining and searching. So far, many algorithms addressing NCMPs 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 properties based on 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. Topological information regarding to the community structures hidden in networks can be inferred from their spectral signatures. Based on 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 scalably mine communities hidden in large-scale networks. The effectiveness and efficiency of the LM algorithm have been theoretically analyzed as well as experimentally validated.

Index Terms—Social network, community structure, 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 or intelligent recommender systems through analyzing and predicting the collective behaviors of users by mining their underlying community structures. Formally, this task can be formulated as a 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 webpages 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], [18], [19]. Thus, how to effectively and efficiently solve such NCMPs is of fundamental importance for both theoretical research and practical applications.

1.1 Related Work

Many methods addressing NCMPs have been developed. In view of the fact that the NCMP is an application-oriented problem, i.e., what structure should be mined will depend on specific applications, it can be stated that all the proposed methods for NCMPs have been heuristic in nature. That is to say, their methodologies would rely more or less on human intuitive observations. So far, most of the existing methods can be classified into two main categories, in terms of whether or not explicit optimization objectives are being used.

The methods with explicit optimization objectives 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 the Kernighan-Lin algorithm [20], the Q function proposed by Newman [21] and employed in several algorithms [7], [11], [15], [21], [22], [23], the energy function of a Potts model with multiple states [24], and the likelihood of a hierarchical random graph [25].

On the other hand, the methods without using explicit optimization objectives solve the NCMP based on predefined assumptions or heuristic rules. For example, the heuristic rule used in the maximum flow community (MFC) algorithm [16] is that the "flows" through intercommunity

- *B. Yang is with the College of Computer Science and Technology, Jilin University, Changchun 130012, China. E-mail: ybo@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, Warwick University, Coventry CV4 7AL, United Kingdom. E-mail: jianfeng.feng@warwick.ac.uk.*

Manuscript received 24 Apr. 2009; revised 2 July 2009; accepted 13 July 2010; published online 16 Nov. 2010.
Recommended for acceptance by S. Greco.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2009-04-0372.
Digital Object Identifier no. 10.1109/TKDE.2010.233.

links should be larger than those of intracommunity links. Similarly, the heuristic rule employed by the GN algorithm [1] is that the "edge betweenness" of intercommunity links should be larger than that of intracommunity links. The Wu-Huberman algorithm [26], Clique Percolation method [2], the Radicchi algorithm [27], and the Finding and Extracting Communities (FEC) algorithm [28] adopted different assumptions, respectively.

Besides the above mentioned two main categories, there exist some other methods for solving the NCMP. For instance, 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 Motivation

What is the nature of network communities? So far, there is no standard answer to this question. In fact, this is the reason why more and more attentions have been paid to deal with this interesting as well as challenging question. In the literature, many efforts can be found that attempt to model and detect communities by introducing predefined optimization objectives or heuristic rules relying on the human intuitive observations of certain characteristic structures or behaviors as exhibited by networks containing community structures, as discussed in the above section.

Now the question becomes whether or not it is possible to introduce a new dimension that is distinct from those addressed in the existing work, for the purpose of further exploring and describing the nature of network communities, and thereafter, to offer a new perspective on how to effectively and scalably solve the NCMP. In order to provide an answer to this question, in this paper, we present a novel model for characterizing network communities by means of introducing a stochastic process on networks and analyzing its dynamics based on the large-deviation theory. Within the framework of the proposed model, we aim to further address the following issues concerning network community structures:

- Are there any relationships between the community structures and other phenomena as demonstrated by networks, such as the stability or the metastability of stochastic processes on them?
- Are there any hidden yet significant relationships between the community structures and the characteristic features of networks, such as the eigenvalues of their adjacency matrices?
- What kinds of topological information of community structures can we infer from a given network, before actually clustering it by an algorithm?
- Can we propose a computationally scalable algorithm for mining communities from large-scale and real-world networks based on the newly proposed model?

1.3 Our Contributions

Unlike the heuristics as adopted by the existing methods, this paper attempts to explore the notion of network communities by understanding the dynamics, instead of the statics, of networks.

Previously, we have proposed a heuristic algorithm to partition signed social networks based on a proposed Markov random walk model [28]. We have found that the dynamics of such a stochastic process on a network with a well-formed community structure can exhibit local mixing behaviors [29]. Based on such an observation, we conjecture that the observable community structures are actually an external phenomenon explicitly demonstrated by, as well as implicitly resulted from, the metastability of networks.

Starting from this heuristic, in this work, we will discuss the connection between the community structure of a network and the metastable states of the Markov chain on it. Based on the hitting and exiting times of such metastable states, we will propose a new measurement, i.e., spectral signature, to characterize and analyze network communities. For a given network, without clustering it using any particular algorithms, one could infer, from its spectral signature, certain topological information as related to its community structure, such as the cohesion and separability of communities, the number of communities, and the hierarchical structure of communities. Based on the above connection and new metrics, we will present a 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 on Local Mixing properties) algorithm, to practically solve large-scale NCMPs. 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 network communities. Then, a general framework for characterizing, analyzing, and mining communities in networks is 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 applicability and limitations. Finally, Section 7 concludes the main results and contributions of this work.

2 A Stochastic Model and Its Dynamics

2.1 Basic Idea

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 global mixing state (i.e., a globally stable state), it will go through a hierarchy of local mixing states (i.e., metastable states) first, and in each of them, locally uniform transition distributions will be observed. Important topological information related to network communities can be inferred from the hitting and exiting times of local mixing states. Such times can be computed based on the large-deviation theory in the case of the potential functions corresponding to networks being explicit. However, network potential functions are all implicit and hard to find out. For this reason, we will turn to approximately estimate these times in terms of networks'

spectra based on some corollaries of the large-deviation theory. Since the estimated times are asymptotic but exact, the proposed model in terms of networks' spectra for characterizing and analyzing network community structures is heuristic.

The large-deviation theory is one of the most successful frameworks in temporary mathematics, was developed by Varadhan, Freidlin, and Wenzel. In this work, we follow the results of Freidlin and Wenzel which are actually a generalization of the classical Cramer formula in physics (a well-known result). Consider a particle moves in a potential well with a noise intensity ϵ. The Cramer formula tells us that the first exiting time is proportional to $exp(H/\epsilon)$ where H is the energy barrier. Freidlin and Wenzel generalized the result to a general nonlinear system with multipotential wells. Intuitively speaking, one potential well is one community in our understanding here. The large-deviation theorem is rigorously proved in the limit that the noise intensity ϵ goes to zero.

2.2 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

$$\begin{aligned} &P(X_t = i_t | X_0 = i_0,\ X_1 = i_1, \ldots, X_{t-1} = i_{t-1}) \\ &\quad = P(X_t = i_t | X_{t-1} = i_{t-1}). \end{aligned}$$

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 $D = diag(d_1, \ldots d_n)$, 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

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

where the stationary distribution π is generally defined as a vector satisfying $\pi P = \pi$, which is known to coincide with any row of $\lim_{t \to \infty} P^t$ for regular Markov chains.

2.3 The Dynamics and Its Spectral Transitions

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 local mixing states depending on its total number of communities K. Before the chain reaches its global mixing state, denoted as s_1, it should go through a sequence of local mixing states first along the time dimension, denoted as $\ldots, s_3, s_2, s_1$. In each of the local mixing states, vertices within the same communities have approximately identical row distributions. Correspondingly, in a local mixing state, we should observe a metastable transition matrix. That is, the random walk will stably stay in a metastable state for a period of time with a probability equal to 1, according to the large-deviation theory.

Let T_s^{ext} be the exiting time of the sth local mixing state ($1 \leq s \leq n$). From one main result of the large-deviation theory (refer to [31, Theorem 6.2]), we have

$$T_s^{ext} = e^{H_s/\epsilon}, \tag{5}$$

where H is the potential function of a given Markov chain, H_s is the sth maximum barrier of this potential, and ϵ is a predefined constant denoting the noise intensity.

In general, when P is reversible, it has a potential H. To avoid introducing the reversibility, let us consider a very simple case. With a given probability matrix P on its state space $S = \ldots, 1, 2, \ldots,$ satisfying $P(i, j) > 0, i \in S, j = i + 1, i - 1$ and 0 elsewhere, if we can find a function H such that $P(i, j) \propto \exp[(H(i) - H(j))/\epsilon]$, where ϵ is a constant and represents the noise strength, H is the potential function of P. When ϵ is small, we have $P(i, j) \sim 1$ if $H(i) > H(j)$ and $P(i, j) \sim 0$ if $H(i) < H(j)$. In other words, it almost becomes a deterministic process. The Markov process moves downhill of the energy function H. In general, for a movement in one-dimensional case, it can be described by an ordinary differential equation $dX_t = f(X_t)dt$ where f is a function. When $f = -H'$, H is the potential function and the noise version is defined by a stochastic differential equation $dY_t = f(Y_t)dt + \epsilon dB_t$ where B_t is the Brownian motion. When ϵ is small enough, Y_t is close to X_t. All the definitions above can be generalized to the high-dimensional case.

The large-deviation theory can be applied to a network with or without an explicit potential function. For a network with an implicit potential function, we can also estimate all local mixing times by 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 nonnegative real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \leq 2$. Refer to [31, Theorem 7.3], we have $\lambda_s = e^{-H_s/\epsilon}(1 + o(1))$, where the error $o(1)$ is bounded by ϵ. Recall (5), we have (also see [32, Theorem 1.3])

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

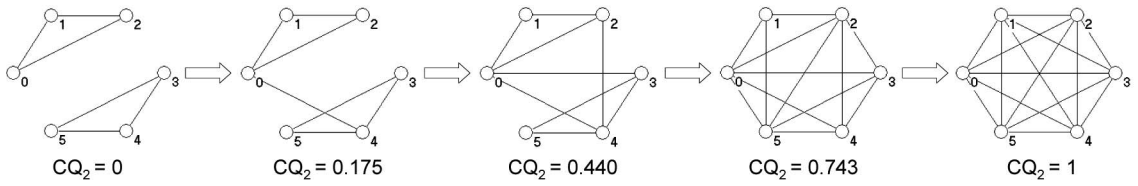


Fig. 1. An evolving process of a dynamic network with corresponding CQ_2 values.

Equation (6) says that for a given Markov chain, T_s^{ext} may fluctuate around $1/\lambda_s$ with a fixed error $o(1)$, which will be determined by the network itself (actually the spectrum of its Markov generator Q). More specifically, the largest of the values λ_i/λ_{i+1} gives the control on the error term $o(1)$. That is, there exists a constant (depend at most on the cardinality of the set of low-lying eigenvalues of Q) such that the error is bounded by the constant times the maximum over all those ratios. Reasonably, we can use the exiting time of the $(s+1)$st local mixing state to estimate the hitting time of the sth local mixing state, that is, $T_s^{hit} = T_{s+1}^{ext} \approx 1/\lambda_{s+1}$. 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 separability 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 cohesion (C_2) and the separability (S_2) of a two-community structure are defined as follows:*

$$C_2 = T_2^{hit} = 1/\lambda_3 \tag{7}$$

and

$$S_2 = T_2^{ext} - T_2^{hit} = 1/\lambda_2 - 1/\lambda_3. \tag{8}$$

A smaller C_2 means better cohesion, and larger S_2 means better separability. Note that $C_2 + S_2 = 1/\lambda_2$, which means the cohesion of each community will be implicitly improved if the separability of two communities is explicitly improved, and vice versa.

Definition 2. *The two-community quality (CQ_2) to measure how well formed a two-community structure of a given network is defined as*

$$CQ_2 = \lambda_2/\lambda_3. \tag{9}$$

Note that

$$\frac{C_2}{S_2} = \frac{1/\lambda_3}{1/\lambda_2 - 1/\lambda_3} = \frac{\lambda_2/\lambda_3}{1 - \lambda_2/\lambda_3} = \frac{CQ_2}{1 - CQ_2}.$$

It is easy to show that $0 \le CQ_2 \le 1$, and a smaller CQ_2 will result in a smaller C_2/S_2, which means better cohesion as well as better separability. Therefore, networks with small CQ_2 approaching to 0 will have well-formed two-community structures, and those with large CQ_2 approaching to 1 will have ambiguous or no community structures. In the extreme cases, the CQ_2 of an unconnected network that contains two completely separated communities is equal to 0. While the CQ_2 of a clique is equal to 1. Fig. 1 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 $\lambda_s = e^{-H_s/\epsilon}$, we have $\log CQ_2 = -(H_2 - H_3)/\epsilon = -\Delta H/\epsilon$. That means the quality of a two-community structure will be exponentially influenced by increasing or decreasing the potential barrier, or the link density, between two communities.

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 intercommunity 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 implicit network potential function. 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 wells, i.e., $\cdots < H_{K+1} \ll H_K < H_{K-1} < \cdots < H_1 = \infty$. Recall that $\lambda_s = e^{-H_s/\epsilon}$. Then, we should have a significant gap between $1/\lambda_K$ and $1/\lambda_{K+1}$.

Definition 3. *The cohesion and the separability of the K-community structure of a given network are defined as*

$$C_K = T_K^{hit} = 1/\lambda_{K+1} \tag{10}$$

and

$$S_K = T_K^{ext} - T_K^{hit} = 1/\lambda_K - 1/\lambda_{K+1}. \tag{11}$$

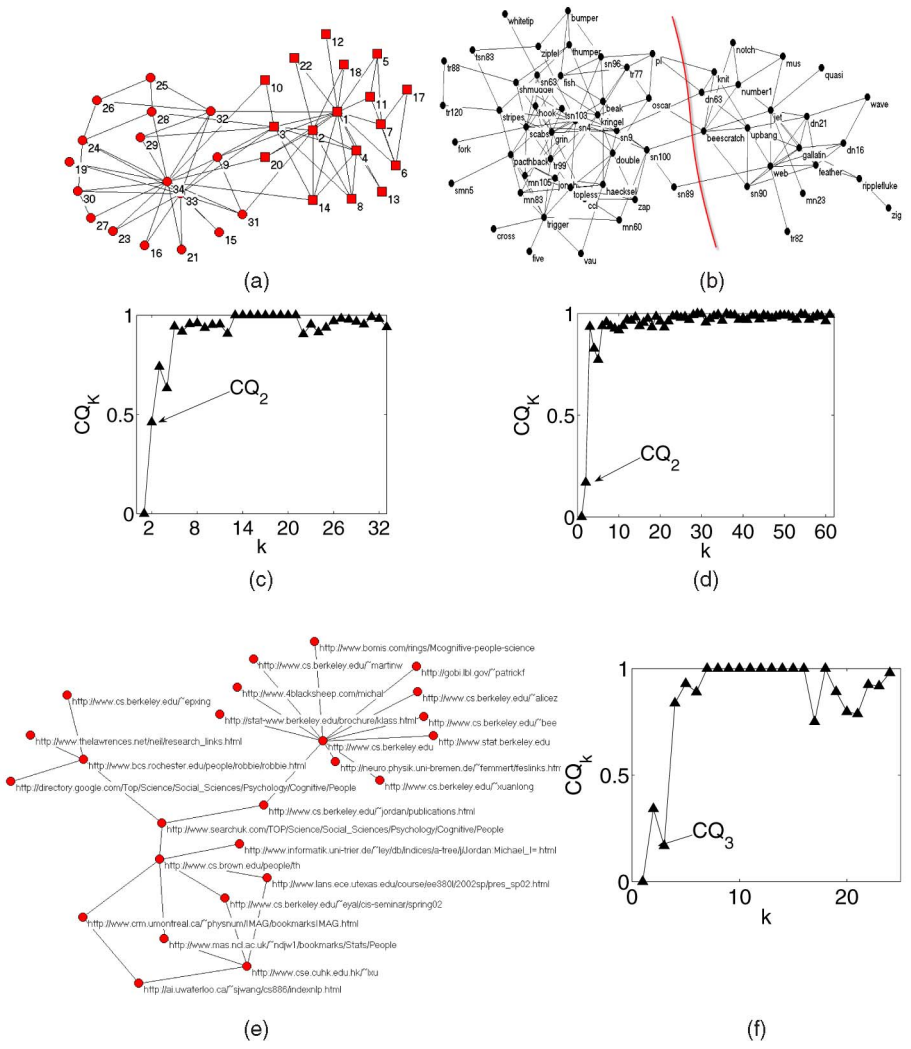


Fig. 2. (a) The karate network. (b) The dolphin network. (c)-(d) The spectral signatures of the two social networks. (e) A subgraph of the web containing 25 pages. (f) The spectral signature of the graph in (e).

Definition 4. *The K-community quality (CQ_K) to evaluate how well formed a K-community structure of a given network is defined as*

$$CQ_K = \lambda_K / \lambda_{K+1}. \tag{12}$$

Similarly, it is easy to show $0 \leq CQ_K \leq 1$, and a small CQ_K implies a better K-community structure with better cohesion as well as better separability. If a K-community structure is well formed, the barriers between K wells should be very large, and thus $\lambda_s (1 \leq s \leq K)$ should approach to zero. This indicates that we can infer the number of communities simply by counting the number of eigenvalues close to zero. While, for real networks with noises, we can use the following method to estimate the number of well-formed communities in a network.

$$K = \arg\min_k CQ_k. \tag{13}$$

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

Example 2. Figs. 2a and 2b show two simple but well-known real-world social networks, respectively. They are the karate club network [33] (containing two actual communities denoted by circles and squares, respectively) and the dolphin network [34] (containing two actual communities split by a solid line). Figs. 2c and 2d 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 two-community structure of dolphin network is much better than that of the karate network, as we intuitively observe from these two visualized networks. Fig. 2e shows a small web containing three web communities. Fig. 2f shows the spectral signature of this web, in which CQ_3 is the minimum value.

Equation (13) is robust for networks with community structures. In these cases, the spectral structures of networks will be stable against noises (a few of noisy links between communities), and thus the number of communities can be robustly estimated by finding the most significant gap of consecutive eigenvalues. On the other hand, the spectral structures of networks with inexplicit community structures

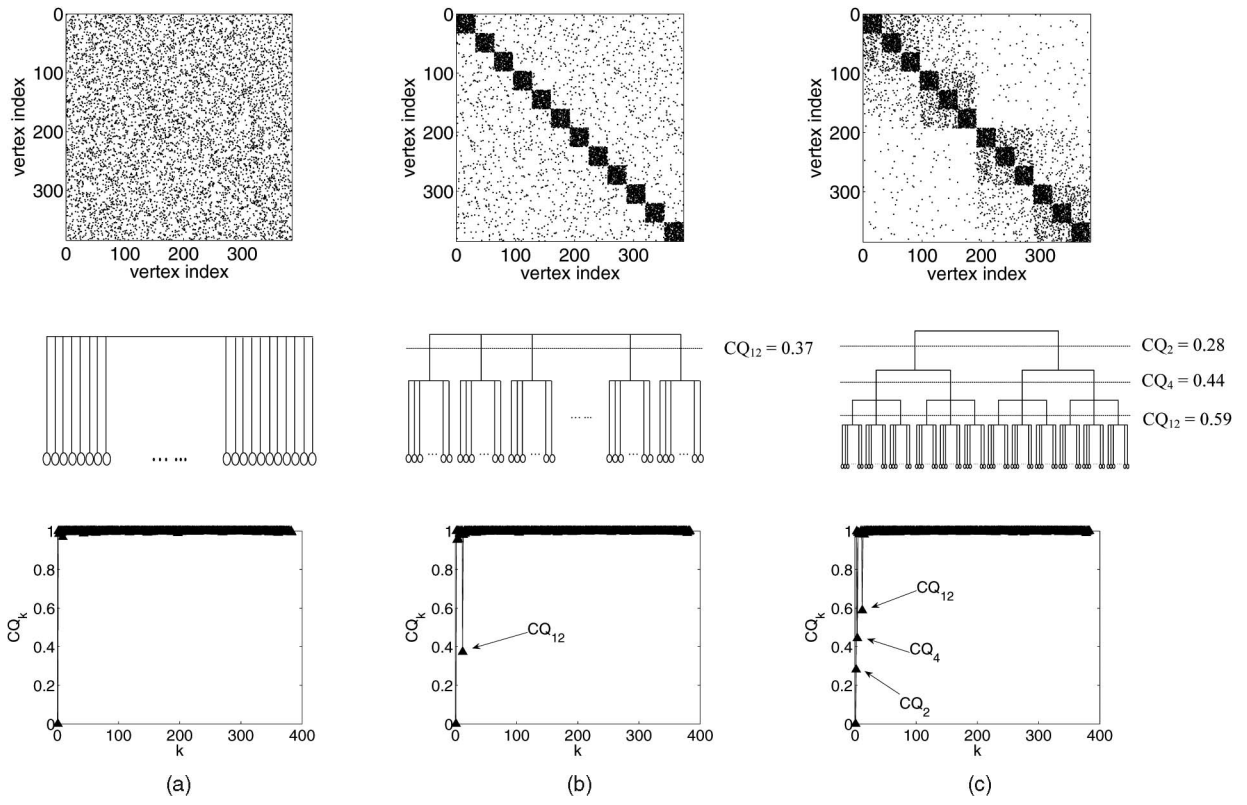


Fig. 3. (a) A network without a hierarchical structure, its dendrogram, and its spectral signature. (b) A network with a single-level community structure, its dendrogram, and its spectral signature. (c) A network with a multilevel community structure, its dendrogram, and its spectral signature.

might be sensitive to noises since their low-lying eigenvalues are far from zero and consequently there are no significant gaps between consecutive eigenvalues.

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 3. Fig. 3 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. 3a, in which all $CQ_k(k > 1)$ will be very close to 1. 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. 3b, in which CQ_K will be approach to 0 and all others (except CQ_1) will be very close to 1. While, for a network with a hierarchical community structure on multiple scales, its spectral signature will look like the one shown in Fig. 3c, in which a number of $CQ_k(k > 1)$ will be approach to 0, and again all others (except CQ_1) will be very close to 1. 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 well-formed communities. For the network shown in Fig. 3c, 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 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, one can characterize and analyze its communities by answering some questions related to its topological structures through observing and inferring its spectral signature. For example,

- Does a network have a well-formed community structure? $\Leftrightarrow$ is the minimum CQ_K close to zero?
- How many well-formed communities are hidden in a network? $\Leftrightarrow$ what is the position of the minimum CQ_K?
- Do these communities stand clear from each other? $\Leftrightarrow$ is S_K large?
- Are they close to each other, respectively? $\Leftrightarrow$ is C_K small?
- 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 community structure 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

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' communities with the above spectral signature;
5)	mine K communities from the matrix $P^{1/\lambda_{K+1}}$ if CQ_K is lower than a threshold.

system, its network will contain a unique community. While, if a system is not stable, its network will demonstrate obvious community structure, in which different levels of dendrogram correspond to different metastable states of the system. So, CQ could also be used to characterize the stability of a real social or biological system. Larger CQ means better 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 serve as an important one to characterize the topological features of networks with community structures.

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 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 multilevel 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. 3c, 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 eigenvalues/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

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

In this section, we will discuss how to infer communities from a single column distribution by addressing three 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:

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

4.3 Ordering Time Distribution (OTD)

Let $P_c^{(t)} = (p_{1,c}^{(t)}, \ldots, p_{n,c}^{(t)})^T$ be the cth column of the transition matrix P^t. It can be computed 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 partitions all vertices by the mean of one column distribution, the obtained bipartition will keep stable until the chain leaves that local mixing state. Based on this property, T_K^{hit} can be estimated as follows:

Definition 6. *Let B_t be the bipartition of $P_c^{(t)}$ by its mean. Time T_{ord} is called* ordering time *when $B_{T_{ord}} = B_{T_{ord}-1}$. Correspondingly, $P_c^{(T_{ord})}$ is called* ordering time distribution.

T_K^{hit} can be estimated by T_{ord}. For each t, (15) can be computed within $O(m)$ time for all i; the condition $B_t = B_{t-1}$ can be checked within $O(n)$ time. Totally, the time to compute an OTD is bounded by $O(T_{ord}(n+m))$. 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 $O((n+m)/\lambda_{K+1})$.

4.4 Inferring Communities from OTD

Theoretically, we can estimate the exact number of communities by finding the minimum eigen-gap according to (13). In practice, however, in order to avoid expensive eigenvalue computation, we will approximately estimate this quantity by adopting a recursive bisection strategy, together with a predefined stopping criterion. Additionally, with such a bisection strategy, we can obtain a binary-tree-like hierarchical structure of communities. In this way, communities and their hierarchy can be inferred from an

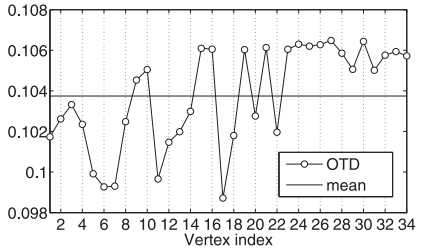


Fig. 4. The mean separates the karate network into two communities which are exactly same as reported by Zachary [33]. The vertices above the mean compose the community denoted as circles and those below the mean compose the community denoted as squares as shown in Fig. 2g.

OTD by recursively dividing it with the corresponding means. Figs. 4 and 5 show two examples.

4.5 Stopping Criterion

The stopping criterion of the recursive bisections is: the local bipartition in question will degenerate the quality of the global partition already obtained.

Q-function proposed by Newman [7] is chosen to evaluate the quality of partitions, which is defined as

$$Q = \sum_i e_{ii} - a_i^2, \tag{16}$$

where e_{ij} denotes the fraction of all weighted edges in networks that link the vertices in community i to those in community j, and $a_i = \sum_j e_{ij}$. It is expected that better partitions of a given network will be with bigger Q-values.

4.6 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.

In each bipartition, finding the node with maximum degree needs $O(n)$ time; computing an OTD needs $O((n+m)/\lambda_{K+1})$ time; dividing an OTD by its mean needs $O(n)$ time; and computing Q-value needs $O(m)$ time (in our implementation, Q-value is incrementally computed, which takes time much less than $O(m)$). So, the worst time of LM is equal to the total time required by the first bipartition multiplied by the total number of recursively callings. For finding out all K communities, exactly $2K-1$ recursive callings are required. Therefore, the worst time of LM is bounded by $O(K(n+m)/\lambda_{K+1})$. Recall that λ_{K+1} is the hitting time of the Kth metastable state, and will be decided by the cohesion of communities rather than the scale of network. The scalability of LM will be demonstrated in Section 5.

TABLE 2
The LM Algorithm

1)	select an attractor with the maximum degree;
2)	calculate an OTD regarding to this attractor;
3)	bisection this OTD by its mean;
4)	return if stopping criterion is satisfied; otherwise
5)	bipartition the network and recursively manipulate two sub-networks.

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

We have compared the accuracy of the LM with five most well-known algorithms, including: the GN algorithm [1], the FN algorithm [21], two spectral methods (the Ncut algorithm [6] and the Mincut algorithm [5]), and the GA algorithm [11] in terms of a widely used random network model, which can produce a randomly synthetic network containing four predefined communities and each of them contains 32 vertices. The average degree of vertices is 16, and the ratio of intracommunity 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. 6. Communities are considered to be correctly discovered if all vertices are clustered into four original groups.

Here, we briefly introduce the five competing algorithms. Mincut(minimum cut) and Ncut(normalized cut) are two typical spectral methods, which, respectively, aim to optimize "cut" and "normalized cut" criteria by computing the K smallest eigenvectors of Laplacian and normalized Laplacian

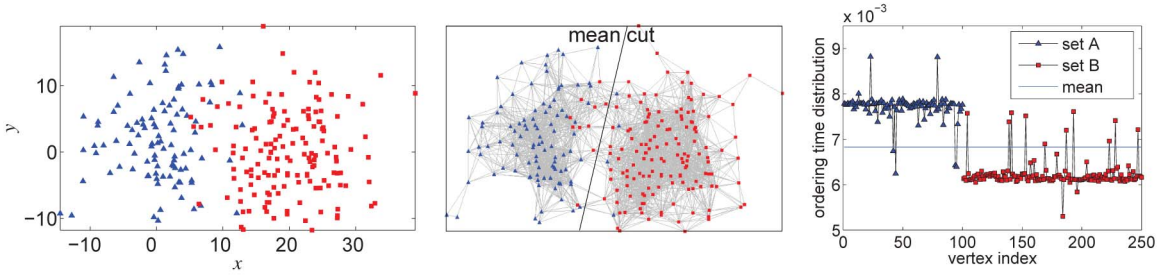


Fig. 5. Left: a point set generated by a mixed Gaussian distribution. Set A containing 100 points denoted by triangles is generated by a Gaussian distribution with mean 0 and variance 6, and set B containing 150 points denoted by squares is generated by a Gaussian distribution with mean 20 and variance 6. Middle: the network generated by adding weighted links between points according to the Gaussian kernel function $a_{ij} = exp(-\|x_i - x_j\|^2/\epsilon)$ 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.

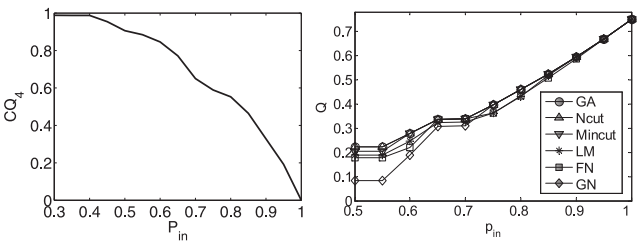


Fig. 6. Left: the CQ_4 values of networks with different P_{in}. Right: the clustering accuracy of six algorithms.

matrices. Generally, the time complexity of spectral methods is $O(n^3)$. A detailed comparison between LM and spectral methods can be found in Section 6.1. FN and GN are two optimization-based algorithms, which optimize same objective, i.e., the Q-function defined by (16). FN uses a bottom-up greedy method to obtain an approximately optimal solution, which takes $O(mn)$ time. While, GA tries to find a globally optimal partition that uses a simulated annealing (SA)-based local search method. GA runs very slowly with an exponential time. GN is a heuristic method, which repeatedly partitions a network by finding and cutting links with the biggest "edge betweenness"; its runtime is $O(nm^2)$.

Fig. 6 presents the experimental results, in which Q-function is adopted to compare the qualities of partitions obtained by different algorithms. Each point in curves was obtained by taking the average over the best 50 results of running 100 synthetic networks. In the case of $p_{in} \geq 0.65$, the Q-values obtained by five algorithms are quite close. On average of $0.5 \leq p_{in} \leq 1$, the ranking is GA, Ncut, Mincut, LM, FN, and GN. One thing should be noticed that, in this experiment, two spectral methods know the number of communities beforehand in order to determine how many eigenvectors they should compute.

5.2 Asymptotic Runtime of Algorithms

Fig. 7 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 2 GHz CPU and a 4 GB memory. The operating system was Windows XP, and the simulation software was programmed by Matlab 7.0. Since most of real-world networks are sparse, in this experiment, we use sparse synthetic networks to estimate the asymptotic runtime of respective algorithms. Given an n, a network containing $m = O(n)$ links and a predefined multiple community structure will be randomly generated. 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 Analyzing Real-World and Large-Scale Networks

We have tested the LM against some real-world networks including social, biological, and technical networks, and compared its performance in terms of efficiency and effectiveness with other algorithms. The experimental environment is same as described in Section 5.2. The results are shown in Table 3, in which the data are, respectively, karate club network [33], dolphin network [34], football association network [1], semantic network [2], scientific collaboration network [4], webs of nd.edu domain, actor coappearance network (IMDB) (above two are from www.nd.edu/ networks/resources), protein homology network, webs of Stanford and UC Berkeley, webs of google, Amazon-product network, and Pennsylvania road network (above networks are from cs-www.cs.yale.edu/homes/mmahoney/NetworkData).

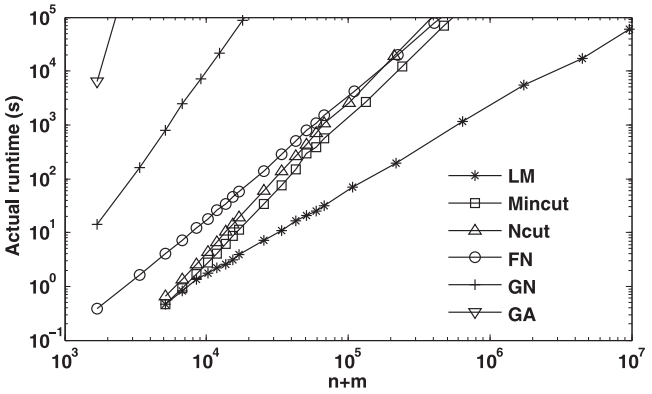


Fig. 7. The actual runtime of six algorithms with respect to network scales.

Each entry of this table is Time/Q, which, respectively, denotes actual running time and Q-value. "$-$" denotes "unavailable" due to "out of memory" or "runtime over seven days." Q-value is widely used to evaluate the quality of obtained partitions. As mentioned before, it is expected that better partitions of a given network will be with bigger Q-values. Table 3 also shows the *order time* (OT) of networks with different scales.

From Table 3, we can observe the following facts: 1) the *order time* of networks is not proportional to their scales; for very small networks such as karate club network with 34 nodes, T_{ord} is comparable with n; in the cases of large-scale networks, $T_{ord} \ll n$; 2) LM is not as efficient as spectral methods in the cases of small networks with hundreds of nodes; and it is much efficient than others in the cases of large-scale networks with more than $O(10^3)$ nodes; and 3) in terms of Q-value, GA performs best, and LM performs better than two spectral methods against real-world networks.

Fig. 8 shows the outputs of LM against four large-scale networks. The outputs of LM turn to be approximately diagonal matrices, in which communities are distributed along diagonal lines. As an example, we present the statistical information of communities of scientific collaboration network. This network codes research collaborations among 56,276 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 3,15,810 weighted edges. From the output of LM, 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. Totally, 160 communities are detected; the maximum size is 711; the minimum size is 3; and the average size is 352.

TABLE 3
Testing the Performance of Different Algorithms against Real-World Networks

Networks	Node/Link	OT	LM	Mincut	Ncut	FN	GN	GA
karate	34/78	15	0.031s/0.374	0.008s/0.234	0.021s/0.341	0.031s/0.253	0.102s/0.401	276.7s/0.420
dolphin	62/160	8	0.040s/0.498	0.010s/0.377	0.027s/0.376	0.078s/0.372	0.316s/0.471	756.1s/0.528
football	115/613	7	0.062s/0.570	0.034s/0.456	0.043s/0.459	0.125s/0.455	0.146s/0.599	1277.7s/0.603
semantic	7k/31k	68	4.28s/0.480	43.3s/0.312	51.9s/0.331	38.25s/0.382	–	–
arxiv	56k/315k	600	168.3s/0.735	5953.6s/0.314	6153.2s/0.449	3551.4s/0.595	–	–
proteins	31K/1.2M	38	465.8s/0.910	–	–	–	–	–
web-BerkStan	320k/1.5M	943	638.6s/0.826	–	–	–	–	–
web-nd	325k/1M	193	1365.2s/0.853	–	–	–	–	–
IMDB	392k/1.2M	296	1524.8s/0.682	–	–	–	–	–
AmazonProd	524k/1.5M	1395	1337s/0.8081	–	–	–	–	–
web-google	855k/4.3M	3050	4865.7s/0.941	–	–	–	–	–
road-PA	1.09M/1.54M	1824	28932s/0.971	–	–	–	–	–

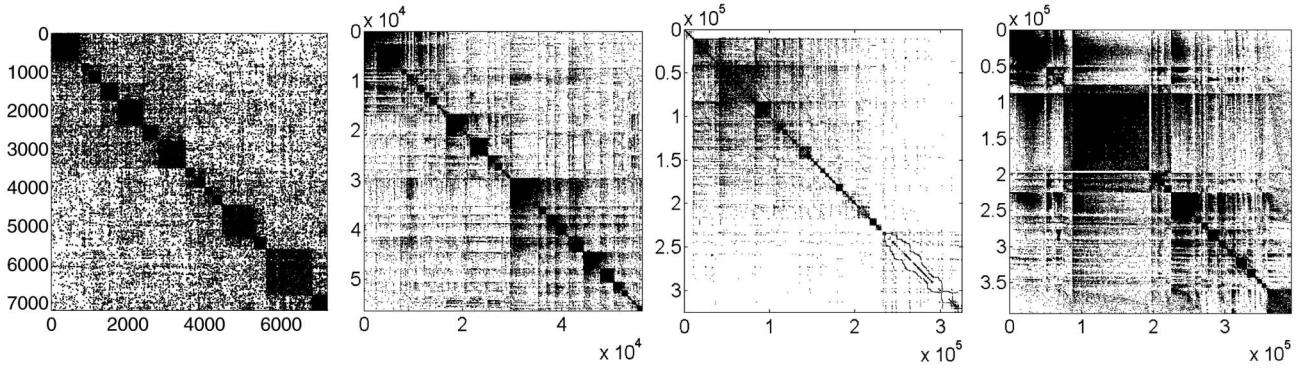


Fig. 8. The outputs of the LM algorithm against large-scale networks. From left to right: semantic network, scientific collaboration network, webs of nd.edu domain, and actor coappearance network.

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.

First, spectral graph partition methods are essentially optimization-based methods, which try to optimize different kinds of predefined "cut" criteria, such as "Minimum cut" [5], "normalized cut" [6], "ratio cut" [36], 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.

Second, 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 communities. While, this work has taken much effort to discover such hidden connections.

Finally, the rationales behind spectral methods and the LM are 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 $(K \cdot \frac{n+m}{\lambda_3-\lambda_2})$ for sparse networks by using some spectral techniques, such as the Lanczos methods [37], 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 discovers all K communities by first calculating the OTD, one column distribution of a transition matrix, within a time of $O(\frac{n+m}{\lambda_{K+1}})$, and then splits it by its means with a time of $O(n)$. As shown in Fig. 7 and Table 3, the LM is much efficient than spectral methods in practice. As to effectiveness, spectral methods perform better than LM against predefined synthetic networks as shown in Fig. 6, and LM outperforms spectral methods when dealing with real-world networks as shown in Table 3.

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 by (16) [21]. 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 networks in terms of community structures 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 community structure it has. With CQ, we can quantitatively compare different networks in terms of their community structures.

6.3 Applicability and Limitations

One should note that constructing an objective mathematical model based only on the intrinsic features of network so as to understand, characterize, and analyze network communities remains an open challenge. This is because no standard definition of network community structures exists today. In this work, we will not claim that we have completely solved this problem. Instead, we stress that the main motivation of this work is to show a connection between a community structure and a network's spectrum from the viewpoint of the dynamics of a Markov chain, which is completely different from the existing spectral methods.

It also should be noticed that the approach of characterizing networks by their spectral signatures, as shown in Table 1, is not proposed oriented toward arbitrary networks. In this work, we are particularly interested in the networks with community structures. For these kinds of networks, the metastable behaviors of their Markov chains will be demonstrated, and can be mathematically characterized and analyzed in terms of their low-lying eigenvalues.

As to the implementation issue, the efficiency of LM depends on the value of $1/\lambda_{K+1}$ of its Markov generator, which is estimated by the *ordering time* in our method. $1/\lambda_{K+1}$, the hitting time of the first metasatble state, is determined the cohesion of communities. LM will be very efficient if the hitting time is very small compared with network scales. Otherwise, it might be slow. For example, for small networks in Table 3, LM runs not as efficient as spectral methods in that *ordering time* is comparable with networks' scale.

Another point that should also be explicitly stated is that, as an efficient implementation of the framework given in Table 1, the LM algorithm is proposed by emphasizing more the computational scalability against large-scale networks than the computational accuracy. Nevertheless, the experiments so far using both synthetic and real-world networks have shown that in terms of Q-value (the most widely used evaluation index), the accuracy of the LM algorithm is still quite competitive with other well-known counterparts.

7 Conclusions

This work has uncovered the connection between network community structures 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 community structure, the cohesion and separability 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 inferring its spectral signature and one of its metastable states. Utilizing the basic properties of metastability, i.e., being locally uniform and temporarily fixed, a scalable implementation for this framework, called the LM algorithm, has been proposed, which is oriented toward large-scale networks. Its time complexity in theory has been analyzed, and its performances in practice, including effectiveness and efficiency, have been demonstrated and verified using networks of different types/scales. In the present work, the actual number of communities is estimated by using a recursive bisection strategy, together with a predefined stopping criterion. In our future work, we will address how to estimate this quantity by efficiently determining the minimum eigen-gap without explicitly computing eigenvalues.

Acknowledgments

The authors would like to express their thanks to the anonymous reviewers for their constructive comments and suggestions, and to Anton Bovier for the helpful discussion with him. This work was supported in part by the National Natural Science Foundation of China Grants (60873149, 60973088, 61133011, and 61170092) and the National High-Tech Research and Development Plan of China (2006AA10Z245 and 2006AA10A309), in part by the Open Project Program of the National Laboratory of Pattern Recognition (NLPR), the Basic Scientific Research Fund of Chinese Ministry of Education (200903177), and the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, and in part by a Hong Kong Baptist University FRG grant (06-07-II-66). Jianfeng Feng is also supported by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the FET-OPEN grant agreement BION (213219).

References

[1] M. Girvan and M.E.J. Newman, "Community Structure in Social and Biological Networks," *Proc. Nat'l Academy of Sciences USA*, vol. 9, no. 12, pp. 7821-7826, 2002.

[2] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the Overlapping Community Structures of Complex Networks in Nature and Society," *Nature*, vol. 435, no. 7043, pp. 814-818, 2005.

[3] G. Palla, A.L. Barabasi, and T. Vicsek, "Quantifying Social Group Evolution," *Nature*, vol. 446, no. 7136, pp. 664-667, 2007.

[4] M.E.J. Newman, "Coauthorship Networks and Patterns of Scientific Collaboration," *Proc. Nat'l Academy of Sciences USA*, vol. 101, no. s1, pp. 5200-5205, 2004.

[5] M. Fiedler, "Algebraic Connectivity of Graphs," *Czechoslovakian Math. J.*, vol. 23, pp. 298-305, 1973.

[6] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-904, Aug. 2000.

[7] M.E.J. Newman, "Modularity and Community Structure in Networks," *Proc. Nat'l Academy of Sciences USA*, vol. 103, no. 23, pp. 8577-8582, 2006.

[8] S. White and P. Smyth, "A Spectral Clustering Approach to Finding Communities in Graphs," *Proc. Fifth SIAM Int'l Conf. Data Mining,* 2005.

[9] M. Shiga, I. Takigawa, and H. Mamitsuka, "A Spectral Clustering Approach to Optimally Combining Numerical Vectors with a Modular Network," *Proc. 13th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 647-656, Aug. 2007.

[10] D.M. Wilkinson and B.A. Huberman, "A Method for Finding Communities of Related Genes," *Proc. Nat'l Academy of Sciences USA,* vol. 101, no. suppl 1, pp. 5241-5248, 2004.

[11] R. Guimera and L.A.N. Amaral, "Functional Cartography of Complex Metabolic Networks," *Nature,* vol. 433, no. 2, pp. 895-900, 2005.

[12] E. Ravasz, A.L. Somera, and D.A. Mongru, "Hierarchical Organization of Modularity in Metabolic Networks," *Science,* vol. 297, no. 5586, pp. 1551-1555, 2002.

[13] V. Farutin, K. Robison, E. Lightcap, V. Dancik, A. Ruttenberg, S. Letovsky, and J. Pradines, "Edge-Count Probabilities for the Identification of Local Protein Communities and Their Organization," *Proteins: Structure, Function, and Bioinformatics,* vol. 62, no. 3, pp. 800-818, 2006.

[14] B. Snel, P. Bork, and M.A. Huynen, "The Identification of Functional Modules from the Genomic Association of Genes," *Proc. Nat'l Academy of Sciences USA,* vol. 99, no. 9, pp. 5890-5895, 2002.

[15] Z. Wang and J. Zhang, "In Search of the Biological Significance of Modular Structures in Protein Networks," *PLOS Computational Biology,* vol. 3, no. 6, p. e107, 2007.

[16] G.W. Flake, S. Lawrence, C.L. Giles, and F.M. Coetzee, "Self-Organization and Identification of Web Communities," *Computer,* vol. 35, no. 3, pp. 66-70, Mar. 2002.

[17] J.M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," *J. ACM,* vol. 46, no. 5, pp. 604-632, 1999.

[18] J.P. Eckmann and E. Moses, "Curvature of Co-Links Uncovers Hidden Thematic Layers in the World Wide Web," *Proc. Nat'l Academy of Sciences USA,* vol. 99, no. 9, pp. 5825-5829, 2002.

[19] H. Zhuge, "Communities and Emerging Semantics in Semantic Link Network: Discovery and Learning," *IEEE Trans. Knowledge and Data Eng.,* vol. 21, no. 6, pp. 785-799, June 2009.

[20] B.W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," *Bell System Technical J.,* vol. 49, pp. 291-307, 1970.

[21] M.E.J. Newman, "Fast Algorithm for Detecting Community Structure in Networks," *Physical Rev. E,* vol. 69, no. 6, p. 066133, 2004.

[22] J. Duch and A. Arenas, "Community Detection in Complex Networks Using Extreme Optimization," *Physical Rev. E,* vol. 72, p. 027104, 2005.

[23] J.M. Pujol, J. Bjar, and J. Delgado, "Clustering Algorithm for Determining Community Structure in Large Networks," *Physical Rev. E,* vol. 74, p. 016107, 2006.

[24] J. Reichardt and S. Bornholdt, "Detecting Fuzzy Community Structures in Complex Networks with a Potts Model," *Physical Rev. Letters,* vol. 93, no. 21, p. 218701, 2004.

[25] A. Clauset, C. Moore, and M.E.J. Newman, "Hierarchical Structure and the Prediction of Missing Links in Networks," *Nature,* vol. 453, no. 5, pp. 98-101, 2008.

[26] F. Wu and B.A. Huberman, "Finding Communities in Linear Time: A Physics Approach," *European Physical J. B,* vol. 38, no. 2, pp. 331-338, 2004.

[27] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and Identifying Communities in Networks," *Proc. Nat'l Academy of Sciences USA,* vol. 101, no. 9, pp. 2658-2663, 2004.

[28] B. Yang, W.K. Cheung, and J. Liu, "Community Mining from Signed Social Networks," *IEEE Trans. Knowledge and Data Eng.,* vol. 19, no. 10, pp. 1333-1348, Oct. 2007.

[29] S. Albeverio, J. Feng, and M. Qian, "Role of Noise in Neural Networks," *Physical Rev. E,* vol. 52, pp. 6593-6606, 1995.

[30] J. Scott, *Social Network Analysis: A Handbook,* second ed. Sage Publications, 2000.

[31] M.I. Fredlin and A.D. Wenzell, *Random Perturbations of Dynamical Systems.* Springer-Verlag, 1984.

[32] A. Bovier, M. Eckhoff, V. Gayrard, and M. Klein, "Metastability and Low Lying Spectra in Reversible Markov Chains," *Comm. Math. Physics,* vol. 228, pp. 219-255, 2002.

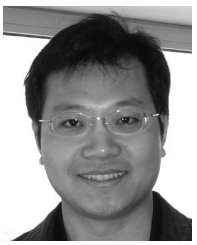
[33] W.W. Zachary, "An Information Flow Model for Conflict and Fission in Small Groups," *J. Anthropological Research,* vol. 33, pp. 452-473, 1977.

[34] D. Lusseau, "The Emergent Properties of a Dolphin Social Network," *Proc. Royal Soc. B: Biological Sciences,* vol. 270, no. Suppl 2, pp. S186-S188, 2003.

[35] S. Fortunato and M. Barthelemy, "Resolution Limit in Community Detection," *Proc. Nat'l Academy of Sciences USA,* vol. 104, no. 1, pp. 36-41, 2007.

[36] Y.C. Wei and C.K. Cheng, "Ration Cut Partitioning for Hierarchical Designs," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 10, no. 7, pp. 911-921, July 1991.

[37] G.H. Golub and C.F.V. Loan, *Matrix Computations.* Johns Hopkins Univ. Press, 1989.



Bo Yang received the BSc, MS, and PhD degrees in computer science from Jilin University in 1997, 2000, and 2003, respectively. He is a professor in the College of Computer Science and Technology at Jilin University, P.R. China. His current research interests include complex networks analysis, data mining, multiagent system, autonomy-oriented computing, and knowledge engineering.

Jiming Liu is the chair professor and the head of the Computer Science Department at Hong Kong Baptist University. He was a professor and the director of the School of Computer Science at the University of Windsor, Canada. His current research interests include autonomy-oriented computing (AOC), web intelligence (WI), and self-organizing systems and complex networks, with applications to: 1) characterizing working mechanisms that lead to emergent behavior in natural and artificial complex systems, and 2) developing self-organized solutions to large-scale, distributed computational problems. He has published more than 250 journal and conference papers, and five authored research monographs, e.g.,*Autonomy Oriented Computing: From Problem Solving to Complex Systems Modeling* (Kluwer Academic/Springer) and *Spatial Reasoning and Planning: Geometry, Mechanism, and Motion* (Springer). He has served as the editor-in-chief of *Web Intelligence and Agent Systems*, an associate editor of *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, and *Computational Intelligence*, and a member of the editorial board of several other international journals.

Jianfeng Feng is a professor in Warwick University, United Kingdom, and the director of the Centre for Computational Systems Biology Centre in Fudan University, P.R. China. His current research interests are computational biology and has published more than 150 journal papers in top tier journals. His modeling work on a "trust" hormone has attracted wide media interests and was reported in BBC News, Washington Post, Reuters, etc.