Probabilistic framework for network partition

Tiejun Li,[1,*] Jian Liu,[1,†] and Weinan E[1,2,‡]

[1] *LMAM and School of Mathematical Sciences, Peking University, Beijing 100871, People's Republic of China*
[2] *Department of Mathematics, Princeton University, Princeton, New Jersey 08544, USA*

(Received 17 December 2008; published 7 August 2009)

Given a large and complex network, we would like to find the partition of this network into a small number of clusters. This question has been addressed in many different ways. In a previous paper, we proposed a deterministic framework for an optimal partition of a network as well as the associated algorithms. In this paper, we extend this framework to a probabilistic setting, in which each node has a certain probability of belonging to a certain cluster. Two classes of numerical algorithms for such a probabilistic network partition are presented and tested. Application to three representative examples is discussed.

DOI: 10.1103/PhysRevE.80.026106 PACS number(s): 89.75.Hc, 89.75.Fb

I. INTRODUCTION

In recent years, the problem of partitioning complex networks into a small number of clusters has attracted a great deal of attention, many different strategies have been proposed [1–11]. Of particular interest among these strategies is the concept of modularity proposed by Newman and coworkers [1–4] as well as the different algorithms introduced for maximizing modularity, such as the greedy algorithm, spectral bisection method, simulated annealing, etc [2,4,8]. The community structure of complex networks has also been studied from the viewpoint of statistical learning. Algorithms such as the hierarchical clustering analysis, parameter estimate of mixture models, and k-means type methods are some of the examples of methods introduced from this viewpoint [3,5–7].

All these algorithms produce a partition of the network and, for any given network, different algorithms may produce different partition. It is natural to ask whether we can quantify the quality of these partitions. This means introducing quantitative measures for the quality of different partitions. One example is the modularity concept discussed earlier. Another example of such a measure was introduced in [6], in the spirit of the optimal prediction theory proposed by Chorin *et al.* [12,13]. The basic idea is to associate the network with a Markov chain [14], introduce a metric on the space of Markov chains (or stochastic matrices) on the network, and then optimally reduce the given Markov chain under this metric. The final minimization problem is solved by an algorithm analogous to the traditional k-means algorithm used in clustering [15]. This approach also bears some similarity to the modified normalized cut (MNCut) algorithms in image segmentation [16,17] and the diffusion maps in data mining [18].

The current paper extends the work in [6] to a probabilistic setting. Instead of assigning nodes to specific clusters, we say that each node has a certain probability of belonging to a certain cluster. We introduce a free energy in the space of probabilistic distributions on the clusters. At zero temperature, this free energy reduces to the functional proposed in [6]. We also develop algorithms for partitioning the network in this setting. This extension is quite natural and useful, particularly, for networks whose community structure is not that pronounced. It is also similar in spirit to "soft clustering" [7] and the fuzzy c-means (FCM) algorithm in data mining [19,20]. As we will see later, soft clustering usually contains more detailed information.

We will present two classes of algorithms: one based on iterating alternatively between the two Euler-Lagrange equations obtained from minimizing the free energy and the other based on the steepest-decent dynamics for the free energy. These algorithms are tested on three examples: the Zachary's karate club network, a sample network generated from a Gaussian mixture model, and the *ad hoc* network model with 1280 nodes. Our numerical results suggest that the alternating iteration algorithm is usually more efficient and accurate. But as an iterative method for a nonlinear problem, convergence is not guaranteed. In this case, the steepest-descent method may provide a reasonable alternative.

The rest of the paper is organized as follows. In Sec. II, we first briefly review the framework in [6] and then introduce our fuzzy network partition formulation. In Sec. III, we introduce the two algorithms alternating iteration algorithm with projections (AIP) and exponentially transformed steepest-descent (ETSD). In Sec. IV, we apply the two algorithms to three examples mentioned before and compare the numerical results and performance of the algorithms. All details of the derivation are left in the Appendix.

II. FRAMEWORK FOR PROBABLISTIC PARTITION OF NETWORKS

We will start with a brief review of the framework of optimal network partition proposed in [6]. Let $G(S,E)$ be a network with n nodes, where S is the set of nodes, $E=\{e(x,y)\}_{x,y\in S}$ is the weight matrix and $e(x,y)$ is the weight for the edge connecting the nodes x and y. A simple example of the weight matrix is given by the adjacency matrix $e(x,y)=0$ or 1, depending whether x and y are connected. We can relate this network to a discrete-time Markov chain with stochastic matrix $P=[p(x,y)]$ whose entries are given by

[*] tieli@pku.edu.cn
[†] dugujian@pku.edu.cn
[‡] weinan@math.princeton.edu

$$p(x,y) = \frac{e(x,y)}{d(x)}, \quad d(x) = \sum_{z\in S} e(x,z), \tag{1}$$

where $d(x)$ is the degree of the node x [14,21]. This corresponds to the isotropic random walk on the network and this chain has stationary distribution

$$\mu(x) = \frac{d(x)}{\Sigma_{z\in S} d(z)} \tag{2}$$

and it satisfies the detailed balance condition.

For a given partition of S as $S=\cup_{k=1}^{N} S_k$ with $S_k \cap S_l = \varnothing$ if $k \neq l$, let $\hat{p}_{kl}$ be the coarse-grained transition probability from S_k to S_l on the state space $\mathbb{S}=\{S_1,\ldots,S_N\}$. This matrix can be naturally lifted to the space of stochastic matrices on the original state space S via

$$\tilde{p}(x,y) = \sum_{k,l=1}^{N} 1_{S_k}(x)\hat{p}_{kl}\mu_l(y), \tag{3}$$

where $\mathbf{1}_{S_k}(x)=1$ if $x\in S_k$ and $\mathbf{1}_{S_k}(x)=0$, otherwise, and

$$\mu_k(x) = \frac{\mu(x)\mathbf{1}_{S_k}(x)}{\hat{\mu}_k}, \quad \hat{\mu}_k = \sum_{z\in S_k} \mu(z). \tag{4}$$

Similar ideas to compress and lift the size of stochastic matrices while preserving their "stochastic matrix" properties are also proposed in [18,22,23], etc.

[6] introduces a metric (Hilbert-Schmidt norm in mathematical language) in the space of stochastic matrices. Let $p_1=[p_1(x,y)]$ and $p_2=[p_2(x,y)]$ be two stochastic matrices. Define

$$\|p_1 - p_2\|_\mu^2 = \sum_{x,y\in S} \frac{\mu(x)}{\mu(y)} |p_1(x,y) - p_2(x,y)|^2. \tag{5}$$

The optimal partition is found by minimizing $\|\tilde{p}-p\|_\mu$.

In the formulation given above, after the partition, every node belongs to one and only one cluster. This is often too restrictive since in many cases, there are nodes on the network that share commonalities with more than one cluster. In the graphical representations, nodes at the boundary between different clusters are typically of this type. In social networks, when one wants to divide the people into different groups according to their mutual social contacts, some of them will have nonzero probability belonging to different clusters. They play the role of the intermediates. In molecular dynamics, when one aims to divide the trajectory into different domains which subordinate to different metastable states, the transitional nodes will stay in the middle and play the role of bottlenecks. This motivates the extension of the optimal partition theory to a probabilistic setting.

The main idea is to replace the indicator function $\mathbf{1}_{S_k}(x)$ in Eq. (3) by a general probability functions $\rho_k(x)$, where $\rho_k(x)$ is the probability that the node x belongs to the kth community. Naturally, we require

$$\rho_k(x) \geq 0, \quad \sum_{k=1}^{N} \rho_k(x) = 1, \tag{6}$$

for all $x\in S$. As before, we define the transition-probability matrix of the induced Markov chain as

$$\tilde{p}(x,y) = \sum_{k,l=1}^{N} \rho_k(x)\hat{p}_{kl}\mu_l(y), \quad x,y \in S, \tag{7}$$

where

$$\mu_k(x) = \frac{\rho_k(x)\mu(x)}{\hat{\mu}_k}, \quad \hat{\mu}_k = \sum_{z\in S} \rho_k(z)\mu(z). \tag{8}$$

The idea of lifting the size of stochastic matrices is similar as the hard clustering case and it expresses the perspective that the node x transits to y through different channels from community k to community l with their corresponding belonging probability and stay there in equilibrium state. It is not difficult to check that $\tilde{p}(x,y)$ is indeed a transition-probability matrix and satisfies the detailed balance condition with respect to μ if $\hat{p}_{kl}$ satisfies the detailed balance condition with respect to $\hat{\mu}$.

Given the number of the communities N, we optimally reduce the Markov chain from the network dynamics by considering the following minimization problem:

$$\min_{\rho_k(x),\hat{p}_{kl}} J = \|p - \tilde{p}\|_\mu^2 = \sum_{x,y\in S} \frac{\mu(x)}{\mu(y)} |p(x,y) - \tilde{p}(x,y)|^2 \tag{9}$$

subject to the constraints [Eq. (6)] and

$$\hat{p}_{kl} \geq 0, \quad \sum_{l=1}^{N} \hat{p}_{kl} = 1. \tag{10}$$

The minimization problem (9) can be understood as the infinite "temperature" version of the following problem:

$$\min_{\rho_k(x),\hat{p}_{kl}} \left[J + \frac{1}{T} \sum_x \sum_k \rho_k(x) \ln \rho_k(x) \right], \tag{11}$$

where the non-negative parameter T plays the role of temperature. When $T=0$, the last term in Eq. (11) becomes a hard constraint, namely, $\rho_k(x)$ is either 0 or 1. When $T=\infty$, we recover Eq. (9). Note a peculiar feature of the functional (11): $1/T$ rather than T appears in front of the entropy term.

To minimize the objective function J in Eq. (9), we define

$$\hat{p}_{kl}^* = \sum_{x,y\in S} \mu_k(x)p(x,y)\rho_l(y) = \frac{1}{\hat{\mu}_k} \sum_{x,y\in S} \mu(x)\rho_k(x)p(x,y)\rho_l(y), \tag{12}$$

which is motivated by the hard clustering case. Then $\hat{p}_{kl}^*$ is indeed a stochastic matrix since $\Sigma_{z\in S}\mu_k(z)=1$ for all k. Furthermore, it is easy to see that $\hat{p}_{kl}^*$ satisfies the detailed balance condition with respect to $\hat{\mu}$.

The optimization of J with constraints $\Sigma_{k=1}^{N}\rho_k(x)=1$ corresponds to find the critical points of Eq. (9). We can derive the Euler-Lagrange equations as

$$(\mathit{\Gamma}_{\hat{\mu}}^{-1} \cdot \hat{\hat{\mu}}) \cdot \hat{p} \cdot (\mathit{\Gamma}_{\hat{\mu}}^{-1} \cdot \hat{\hat{\mu}}) = \hat{p}^*, \tag{13a}$$

$$\rho = I_{\hat{\mu}} \hat{p}^{-1} \hat{\hat{\mu}}^{-1} \rho p^T, \tag{13b}$$

where $\rho=[\rho_k(x)]$ is an $N \times n$ matrix, $I_{\hat{\mu}}$ and $\hat{\hat{\mu}}$ are two $N \times N$ matrices with entries

$$\hat{\hat{\mu}}_{kl} = \sum_{z \in S} \mu(z) \rho_k(z) \rho_l(z) = (\rho \cdot I_\mu \cdot \rho^T)_{kl}, \tag{14}$$

and

$$(I_{\hat{\mu}})_{kl} = \hat{\mu}_k \delta_{kl}, \quad k,l = 1, \ldots, N, \tag{15}$$

respectively. Here $I_\mu(x,y)=\mu(x)\delta(x,y)$, where $\delta(x,y)$ and δ_{kl} are both Kronecker delta symbols.

All of the derivation details for Eqs. (13a) and (13b) are left in the Appendix. They give the necessary condition that the miminizer should satisfy.

III. ALGORITHMS

A. Algorithm based on the Euler-Lagrange equations

A strategy suggested immediately by the Euler-Lagrange equations (13) is to iterate alternatively between the equations for $\hat{p}$ and ρ. To ensure realizability, i.e., the nonnegativity and normalization conditions for $\hat{p}$ and ρ, we add a projection step after each iteration, i.e., we change the optimality conditions (13) to

$$\hat{p} = \mathcal{P}(\hat{\hat{\mu}}^{-1} \cdot I_{\hat{\mu}} \cdot \hat{p}^* \cdot \hat{\hat{\mu}}^{-1} \cdot I_{\hat{\mu}}), \tag{16a}$$

$$\rho = \mathcal{P}(I_{\hat{\mu}} \hat{p}^{-1} \hat{\hat{\mu}}^{-1} \rho p^T). \tag{16b}$$

Here $\mathcal{P}$ is a projection operator which maps a real vector into a vector with non-negative normalized (sum is one) components. This leads to the following.

Algorithm 1: AIP.

Step 1: set up the initial state $\rho^{(0)}$ as the indicator matrix for each node in the network with the k-means algorithm in [6], $n=0$.

Step2: perform the following simple iteration until $\|\rho^{(n+1)} - \rho^{(n)}\| \le E_{\text{tol}}$:

$$\hat{p}^{(n+1)} = \mathcal{P}[(\hat{\hat{\mu}}^{-1} \cdot I_{\hat{\mu}} \cdot \hat{p}^* \cdot \hat{\hat{\mu}}^{-1} \cdot I_{\hat{\mu}})^{(n)}], \tag{17a}$$

$$\rho^{(n+1)} = \mathcal{P}[(I_{\hat{\mu}} \hat{p}^{-1} \hat{\hat{\mu}}^{-1} \rho p^T)^{(n)}]. \tag{17b}$$

Here E_{tol} is a prescribed tolerance.

Step 3: the final $\rho^{(n)}$ gives the fuzzy partition for each node.

Two choices of the projection operator $\mathcal{P}$ are used in our computation. The numerical results seem to be insensitive to the choice. Let $\boldsymbol{u}=(u_1,u_2,\ldots,u_n) \in \mathbb{R}^n$, and $u_i<0$ when $i \in \Lambda$.

(i) Choice 1: direct projection to the boundary.

When $i \in \Lambda$, we set $\mathcal{P}u_i=0$; otherwise, we set $\mathcal{P}u_i = u_i / \Sigma_{j \notin \Lambda} u_j$.

(ii) Choice 2: iterative projection.

First project $\boldsymbol{u}$ to the hyperplane $\Sigma_{i=1}^n u_i=1$. Then check each component of the projected $\boldsymbol{u}$. If $u_{i_0}<0$, we take $\mathcal{P}u_{i_0}=0$ and project it again to a reduced hyperplane $\Sigma_{i \neq i_0} u_i=1$. Repeat the projection procedure to lower and lower dimensional hyperplane until no component is negative.

We have found that the convergence rate depends on the structure of the network. For a complex network with well-clustered community structure, the convergence is usually fast. But for a very diffusive network, convergence may be very slow.

Now let us estimate the computational cost in each iteration. In the iteration step for $\hat{p}$, all of the matrices are on the order of $N \times N$.

(a) The cost in the step for $\hat{p}$. It is easy to see that the computation of $\hat{\mu}$ costs $O(Nn)$, and the computation of $\hat{\hat{\mu}}$ costs $O(N^2 n)$. The computation for $\hat{p}^*$ costs $O(N^2 \mathcal{E})$, where $\mathcal{E}$ represents the number of edges, which is usually assumed to be $O(n)$ in realistic networks. The cost of computing $\hat{\hat{\mu}}^{-1}$ is $O(N^3)$. Therefore, the total cost in in the step of computing $\hat{p}$ is $O[N^2(\mathcal{E}+n)]$.

(b) The cost in the step for ρ. The cost for ρp^T is $O(N\mathcal{E})$, for $I_{\hat{\mu}} \hat{p}^{-1} \hat{\hat{\mu}}^{-1}$ is $O(N^3)$. So the cost for ρ is also $O(N^2 n + N\mathcal{E})$.

B. Variants of the steepest-descent method

Another obvious choice is to minimize the objective function using the steepest-descent method. Then the gradient flow of Eq. (9) is given by

$$\frac{d\hat{p}}{dt} = -\frac{\partial J}{\partial \hat{p}}(\hat{p},\rho), \tag{18a}$$

$$\frac{d\rho}{dt} = -\frac{\partial J}{\partial \rho}(\hat{p},\rho). \tag{18b}$$

Constraints must be enforced to guarantee realizability. There are two natural strategies for enforcing realizability. The first is similar to the procedure used in AIP, namely, to apply projection after each step. In the steepest-descent setting, this is

$$\hat{p}^{(n+1)} = \mathcal{P}\left[\hat{p}^{(n)} - \alpha \frac{\partial J}{\partial \hat{p}}(\hat{p}^{(n)},\rho^{(n)})\right], \tag{19a}$$

$$\rho^{(n+1)} = \mathcal{P}\left[\rho^{(n)} - \alpha \frac{\partial J}{\partial \rho}(\hat{p}^{(n)},\rho^{(n)})\right], \tag{19b}$$

where $\alpha>0$ is a time stepsize. Another strategy is to use simple transforms of the type

$$\hat{p}_{kl} = \frac{e^{Y_{kl}}}{\sum_{m=1}^{N} e^{Y_{km}}}, \quad \rho_k(x) = \frac{e^{Z_k(x)}}{\sum_{m=1}^{N} e^{Z_m(x)}}, \tag{20}$$

where $\{Y_{kl}\}, \{Z_k(x)\} \in \mathbb{R}$ are the generalized coordinates for $\hat{p}_{kl}$ and $\rho_k(x)$, respectively.

To obtain an implementable scheme, let us define the auxiliary matrices

$$M_1 = \hat{p} I_{\hat{\mu}}^{-1} \hat{\hat{\mu}} I_{\hat{\mu}}^{-1} \hat{p}^T \rho - \hat{p} I_{\hat{\mu}}^{-1} \rho p^T, \tag{21a}$$

$$M_2 = \Gamma_{\hat{\mu}}^{-1}\hat{p}^T\hat{\hat{\mu}}\hat{p}\Gamma_{\hat{\mu}}^{-1}\rho - \Gamma_{\hat{\mu}}^{-1}\hat{p}^T\rho p^T. \tag{21b}$$

Then the Euler-Lagrange equations for the minimization problem (9) with the transformation (20) can be given by straightforward calculations,

$$\frac{\partial J}{\partial Y} = 2[(\hat{\hat{\mu}}\hat{p}\Gamma_{\hat{\mu}}^{-1}\hat{\hat{\mu}}\Gamma_{\hat{\mu}}^{-1}) * \hat{p} - (\hat{p}^*)^T * \hat{p} - \mathrm{diag}\{\hat{\hat{\mu}}\hat{p}\Gamma_{\hat{\mu}}^{-1}\hat{\hat{\mu}}\Gamma_{\hat{\mu}}^{-1}\cdot\hat{p}^T\}\cdot\hat{p} + \mathrm{diag}\{(\hat{p}^*)^T\cdot\hat{p}^T\}\cdot\hat{p}], \tag{22a}$$

$$\begin{aligned}\frac{\partial J}{\partial Z} = {} & 2[(M_1+M_2)*\rho - \rho\cdot\mathrm{diag}\{\rho^T\cdot(M_1+M_2)\} \\ & - \mathrm{diag}\{\Gamma_{\hat{\mu}}^{-2}\hat{\hat{\mu}}\Gamma_{\hat{\mu}}^{-1}\hat{p}^T\hat{\hat{\mu}}\hat{p}\}\cdot\rho + \mathrm{diag}\{\hat{p}^*\hat{p}\Gamma_{\hat{\mu}}^{-1}\}\cdot\rho \\ & + \rho\cdot\mathrm{diag}_{vm}\{1_{1\times N}\cdot[(\hat{\hat{\mu}}\hat{p}\Gamma_{\hat{\mu}}^{-1}\hat{\hat{\mu}}\Gamma_{\hat{\mu}}^{-2})*\hat{p}]\cdot\rho\} \\ & - \rho\cdot\mathrm{diag}_{vm}\{1_{1\times N}\cdot[(\hat{p}^*)^T*\hat{p}]\cdot\Gamma_{\hat{\mu}}^{-1}\rho\}]I_\mu,\end{aligned} \tag{22b}$$

where $*$ denotes the element-by-element multiplication of matrices, diag$\{A\}$ is the diagonal part of the matrix A, and diag$_{vm}\{u\}$ is a diagonal matrix formed using the components of the vector u. This leads to the following version of the steepest-descent algorithm.

Algorithm 2: ETSD.

Step 0: get $\hat{p}^*$ and ρ as the indicator matrix obtained from the k-means algorithm in [6].

Step 1: set up the initial value of the matrix $Y_{kl}^{(0)}=\ln\hat{p}_{kl}^*$, take $Z_k^{(0)}(x)=0$ if $\rho_k(x)=1$ and $Z_k^{(0)}(x)=-5$ if $\rho_k(x)=0$ for simplicity [$\exp(-5)\approx 0.006\,7$].

Step 2: update Y and Z with the steepest-descent algorithm,

$$Y^{(n+1)} = Y^{(n)} - \alpha\frac{\partial J}{\partial Y}(Y^{(n)},Z^{(n)}), \tag{23a}$$

$$Z^{(n+1)} = Z^{(n)} - \alpha\frac{\partial J}{\partial Z}(Y^{(n)},Z^{(n)}), \tag{23b}$$

where α is the stepsize for Y and Z.

Step 3: repeat step 2 until $|J^{(n+1)}-J^{(n)}|\le E_{\mathrm{tol}}$. The final $\rho^{(n+1)}$ gives the fuzzy partition probability for each node.

Here taking $Z_k^{(0)}(x)=-5$ when $\rho_k(x)=0$ in the initial step is one of the many reasonable choices. It does not affect the final result. We can estimate the computational cost in each iteration as follows.

(i) The cost in the step for Y. Similar to the AIP algorithm, the cost for computing $\hat{\hat{\mu}}$ is $O(N^2n)$, for $\hat{p}^*$ is $O(N^2\mathcal{E})$. The others are dominated by these two. Thus, the cost for computing $\partial J/\partial Y$ is $O[N^2(\mathcal{E}+n)]$. So the total computational cost in one step for Y is $O[N^2(\mathcal{E}+n)]$ for multiplications and $O(N^2)$ for exponential operations.

(ii) The cost in the step for Z. The cost for computing $\partial J/\partial Z$ is also $O[N^2(\mathcal{E}+n)]$ since $\hat{p}^*$ is involved in the equations. So the total computational cost in one step for Z is $O[N^2(\mathcal{E}+n)]$ for multiplications and $O(Nn)$ for exponential operations.

Note that the computational cost in each iteration step is of the same order as the AIP algorithm except the exponential operations.

IV. NUMERICAL EXAMPLES

We will test these algorithms for three examples: the karate club network, sample network generated from Gaussian mixture model, and the *ad hoc* network with 1280 nodes. We will compare the convergence rate and numerical results for the two algorithms proposed above.

A. Karate club network

This network was constructed by Zachary after he observed social interactions between members of a karate club at an American university [24]. Soon after, a dispute arose between the clubs administrator and main teacher and the club split into two smaller clubs. It has been used in several papers to test the algorithms for finding community structure in networks [3,6].

There are 34 nodes in karate club network (see Figs. 2 and 3), where each node represents one member in the club. In Zachary's original partition, each node belongs to only one subclub after splitting. We label it as black or white color in the figures to show its attribute in the graph representation. From the viewpoint of the soft clustering, the attribute of each node is no longer an indicator function but rather a discrete probability distribution. In our following notations, the association probability ρ_K and ρ_W means the probability of each node belonging to black or white colored group, respectively.

The convergence rate. The convergence history for both methods AIP and ETSD are shown in Fig. 1. We set the tolerance $E_{\mathrm{tol}}=10^{-6}$ in both algorithms. It is used to control $\|\rho^{(n+1)}-\rho^{(n)}\|$ in AIP and $|J^{(n+1)}-J^{(n)}|$ in ETSD. We simply choose $\alpha=20$ in the computations since numerically we observed that larger values of α cause blow up. For the AIP algorithm, the number of iterations needed is 47 with $J_{\min}=4.039\,030$, which is smaller than the result $J_{\min}=4.179\,811$ using the k-means algorithm [6]. For ETSD, the number of iterations needed is 631 with $J_{\min}=4.039\,674$. To further improve the accuracy of ETSD, we use smaller and smaller values for E_{tol}, the results are shown in Table I. We observe that even with $E_{\mathrm{tol}}=10^{-9}$ and after 1944 iteration steps, the resulting $J_{\min}$ is still not good enough compared with the result by AIP. Our explanation is as follows. At first let us remind that the direct iteration of the Euler-Lagrange equations (13a) and (13b) gives negative components, which means that we may have zero components for the final ρ when constrained to the convex domain $\Sigma_{k=1}^N\rho_k(x)=1$, say $\rho_{k_0}(x_0)=0$. These zero components are achieved by the projection step in AIP. But in ETSD, we take the exponential transformation, which implies that the corresponding component $Z_{k_0}(x_0)=-\infty$. To reach this limit, we should have long enough iteration steps. In practical computations, the steepest-descent method drives the component $Z_{k_0}(x_0)$ to a negative number, but it will be stopped after some marching steps with the stopping criterion $|J^{(n+1)}-J^{(n)}|<E_{\mathrm{tol}}$. This

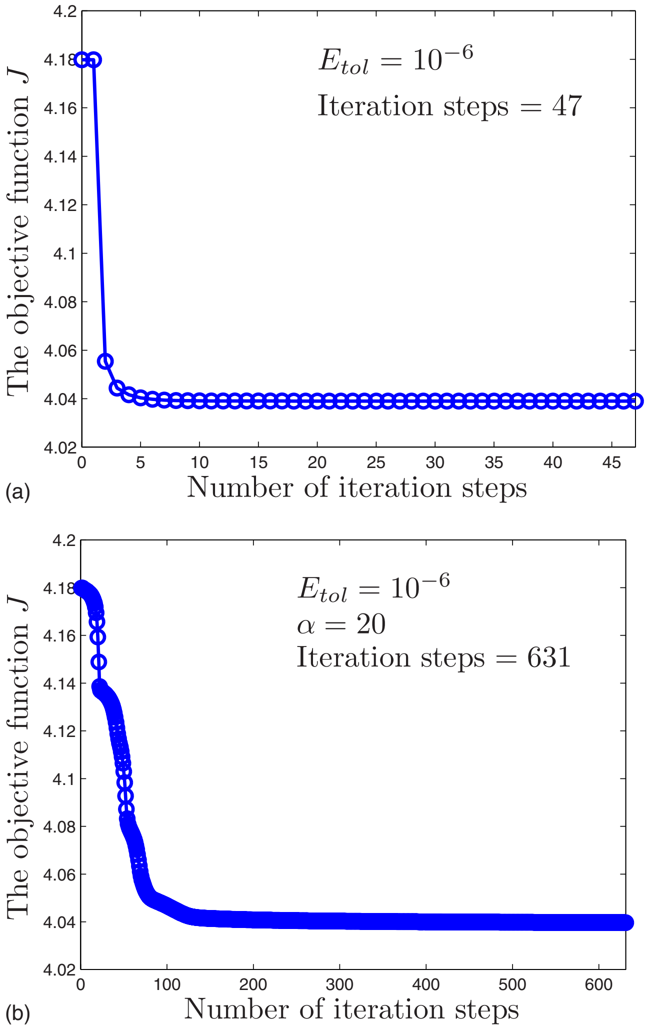

FIG. 1. (Color online) Shown above is the convergence history of the objective function J. The panels (a) and (b) show the results of AIP and ETSD, respectively. The number of iterations needed is 47 for AIP when the $E_{\text{tol}}=10^{-6}$ and 631 for ETSD when the $E_{\text{tol}}=10^{-6}$ and $\alpha=20$.

stopping may introduce noticeable error for $\rho_k(x)$. To achieve better accuracy, we should set the tolerance E_{tol} smaller and smaller and run more and more iterations, but it may cause numerical efficiency problem.

The association probability ρ. The final clustering results are presented in Table II, where ρ_K and ρ_W are the probabilities of belonging to the black or white colored group shown in Fig. 2, respectively. Comparing ρ_K or ρ_W between AIP and ETSD, we find that almost all the errors are less than 10^{-2}, but the association probability—or the soft clustering probability ρ—is quite different from the 0-1 distributions obtained in the k-means algorithm.

TABLE I. The value of the objective function with different tolerances in ETSD. Here $\alpha=20.0$.

E_{tol}	IterStep[a]	J_{min}
10^{-5}	183	4.040980
10^{-6}	631	4.039674
10^{-7}	1861	4.039190
10^{-8}	1901	4.039187
10^{-9}	1944	4.039188

[a] The number of iteration steps.

Now let us compare the association probability ρ_K, ρ_W obtained by AIP with the original partition result obtained by Zachary. In [24], Zachary gave the partition $S_W=\{1:8,11:14,17,18,20,22\}$ and $S_K=\{9,10,15,16,19,21,23:34\}$. If we classify the nodes according to the majority rule, i.e., if $\rho_K(x) > \rho_W(x)$ then we set $x \in S_K$, otherwise, we set $x \in S_W$, we obtain the same partition as Zachary's (see Fig. 2). But in fact we have more detailed information at least geometrically. From Table II, we find $\rho_W=1$ for nodes {5:7, 11:13, 17:18, 22}, which lie at the boundary of the white colored group, and $\rho_K=1$ for nodes {15:16, 19, 21, 23:27, 30, 33}, which mostly lie at the boundary of the black colored group (except node 33, which lies at the center of the black colored group). The others belong to the black and white colored groups with nonzero probability but they fit the intuition from Fig. 2. The nodes {3, 9, 10, 14, 20, 31} have more diffusive probability and they play the role of transition nodes between the black and white colored groups. In particular, node 3 is the most diffusive one. We can visualize the data ρ more transparently with the grayscale plot for each node shown in Fig. 3.

From this result, one would naturally speculate that the members in the middle are somewhat closely associated with both clusters and would be the people who would have a hard time deciding which group to join when the club splits into two, though at this point, we have no additional data to substantiate this.

B. Sample network generated from the Gaussian mixture model

Our second example is a sample network generated from a Gaussian mixture model. This model is related the concept of "random geometric graph" proposed by Penrose [25], except that we take Gaussian mixture here instead of uniform distribution in [25].

First we generate n sample points $\{\mathbf{x}_i\}$ in two-dimensional Euclidean space subject to a 3-Gaussian mixture distribution,

$$\sum_{i=1}^{3} q_i N(\boldsymbol{u}_i, \Sigma_i), \tag{24}$$

where $\{q_i\}$ are weights that satisfy $0 < q_i < 1$, $\Sigma_{i=1}^{3} q_i = 1$. $\boldsymbol{u}_i$ and Σ_i are the mean positions and covariance matrices for each component, respectively. Next, we generate the network with a thresholding strategy. That is, if $|\boldsymbol{x}_i - \boldsymbol{x}_j| \leq \text{dist}$, we assign an edge between the ith and jth nodes; otherwise, they are not connected. With this strategy, the connectivity of the network is induced by a metric. We are interested in the connection between our network clustering and the traditional clustering in the metric space.

We take $n=40$ and generate sample points with the means

$$\boldsymbol{u}_1 = (1.0, 4.0)^T, \boldsymbol{u}_2 = (2.3, 5.3)^T, \boldsymbol{u}_3 = (0.5, 5.8)^T, \tag{25}$$

and the covariance matrices

TABLE II. The association probability of each node. ρ_K and ρ_W are the probabilities of belonging to the black or white colored groups in Fig. 2, respectively.

Nodes		1	2	3	4	5	6	7	8	9	10	11	12
AIP	ρ_K	0.0427	0.0821	0.4314	0.0015	0	0	0	0.0111	0.6619	0.7430	0	0
	ρ_W	0.9573	0.9179	0.5686	0.9985	1.0000	1.0000	1.0000	0.9889	0.3381	0.2570	1.0000	1.0000
ETSD	ρ_K	0.0485	0.0898	0.4412	0.0046	0.0010	0.0007	0.0007	0.0087	0.6718	0.7564	0.0010	0.0027
	ρ_W	0.9515	0.9102	0.5588	0.9954	0.9990	0.9993	0.9993	0.9913	0.3282	0.2436	0.9990	0.9973

Nodes		13	14	15	16	17	18	19	20	21	22	23	24
AIP	ρ_K	0	0.2262	1.0000	1.0000	0	0	1.0000	0.3012	1.0000	0	1.0000	1.0000
	ρ_W	1.0000	0.7738	0	0	1.0000	1.0000	0	0.6988	0	1.0000	0	0
ETSD	ρ_K	0.0014	0.2359	0.9984	0.9984	0.0012	0.0019	0.9984	0.3114	0.9984	0.0019	0.9984	0.9993
	ρ_W	0.9986	0.7641	0.0016	0.0016	0.9988	0.9981	0.0016	0.6886	0.0016	0.9981	0.0016	0.0007

Nodes		25	26	27	28	29	30	31	32	33	34
AIP	ρ_K	1.0000	1.0000	1.0000	0.9496	0.8344	1.0000	0.7210	0.8956	1.0000	0.9475
	ρ_W	0	0	0	0.0504	0.1656	0	0.2790	0.1044	0	0.0525
ETSD	ρ_K	0.9987	0.9988	0.9984	0.9570	0.8473	0.9992	0.7305	0.9026	0.9982	0.9550
	ρ_W	0.0013	0.0012	0.0016	0.0430	0.1527	0.0008	0.2695	0.0974	0.0018	0.0450

$$\Sigma_1 = \Sigma_2 = \Sigma_3 = \begin{pmatrix} 0.15 & 0 \\ 0 & 0.15 \end{pmatrix}. \tag{26}$$

Here we pick nodes 1:10 in group 1, nodes 11:25 in group 2, and nodes 26:40 in group 3 for simplicity. With this choice, approximately $q_1 = 10/40$, $q_2 = q_3 = 15/40$. We take dist=1.0 in this example. The sample points are shown in Fig. 4 and the corresponding network is shown in Fig. 5.

To evaluate our result obtained by the algorithms proposed above, we first define *a priori* soft clustering probability $\rho_i^{\text{priori}}(\boldsymbol{x})$ for any $\boldsymbol{x}$ as

$$\rho_i^{\text{priori}}(\boldsymbol{x}) = \frac{q_i p_i(\boldsymbol{x})}{\sum\limits_{i=1}^{N} q_i p_i(\boldsymbol{x})},$$

where $p_i(\boldsymbol{x})$ is the Gaussian probability density function with mean $\boldsymbol{u}_i$ and covariance Σ_i. Notice that this priori probability is independent of the topology of the network, which can be only considered as a reasonable reference value but not an exact object.

It will be instructive to compare our result with those obtained from fuzzy c-means algorithm [19,20] since the metric is known in this case. We also apply it to classify the samples. The main idea of the traditional fuzzy c-means algorithm is to minimize the objective function

$$J_{\text{FCM}} = \sum_{j=1}^{N} \sum_{i=1}^{n} \rho_j(\boldsymbol{x}_i)^b \|\boldsymbol{x}_i - \boldsymbol{m}_j\|^2, b \geq 1, \tag{27}$$

where $\boldsymbol{x}_i$ are samples and $\boldsymbol{m}_j$ are cluster centers. We choose $b=2$ in our computation. $\rho_j(\boldsymbol{x}_i)$ denotes the probability that $\boldsymbol{x}_i$ belongs to cluster j, which satisfies the condition

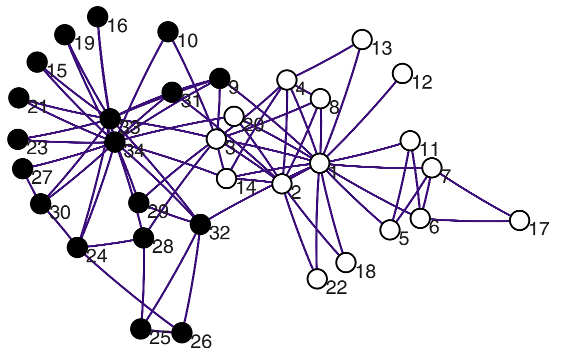

FIG. 2. (Color online) The partition obtained using the majority rule, i.e., if $\rho_K(x) > \rho_W(x)$ then we set $x \in S_K$; otherwise, we set $x \in S_W$. The result is the same as Zachary's.

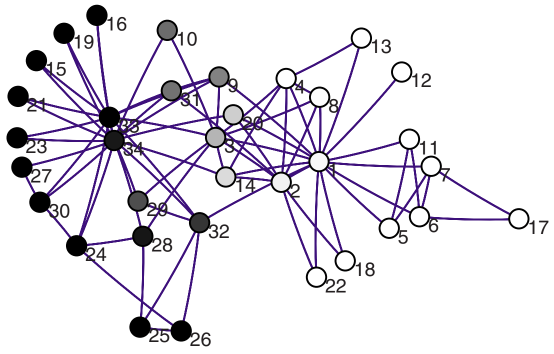

FIG. 3. (Color online) The grayscale plot of ρ_K and ρ_W for each node in karate club network. The darker the color, the larger the value of ρ_K. The transition nodes or intermediate nodes are clearly shown.

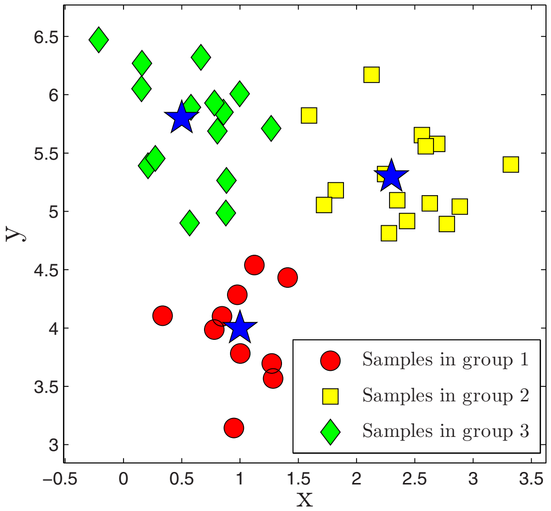

FIG. 4. (Color online) 40 sample points generated from the given 3-Gaussian mixture model. The star symbols represent the centers of each Gaussian component. The diamonds, circles, and squares represent the sample points in the three different components, respectively.

$$0 \le \rho_j(\boldsymbol{x}_i), \sum_{j=1}^{N} \rho_j(\boldsymbol{x}_i) = 1, \quad i = 1,2, \ldots ,n. \tag{28}$$

We can derive the Euler-Lagrange equations for this objective function with respect to $\boldsymbol{m}$ and ρ and iterate until the fixed points are found. We refer the readers to [19,20] for more details.

In Table III, we compare the needed iteration steps, the minimum value of the objective function $J_{\min}$, and the mean and maximal L^{∞} error of ρ compared with the traditional fuzzy c-means algorithm and the priori probabilities for AIP and ETSD. The intermediate association probabilities ρ are listed in Table IV. Comparing AIP and ETSD, we can say AIP is more efficient. The maximal deviation of ρ between these two algorithms is less than 0.03. Comparing our methods with the traditional FCM, the mean deviation of ρ is less than 0.083, but the maximal deviation is about 0.22. Comparing with the priori probabilities, the mean deviation of ρ is less than 0.063, which is smaller, but the maximal deviation is about 0.40, which is larger. A detailed inspection shows that the nodes with large deviations are all located in the transition region and the largest deviation occurs for node 20. From the above comparisons, our methods achieve reasonable results that fits the intuition from the network topology visualization.

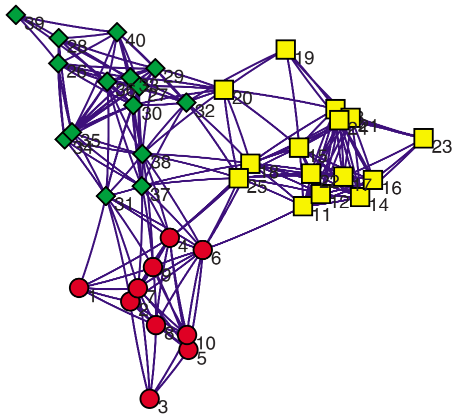

FIG. 5. (Color online) The network generated from the sample points in Fig. 4 with the parameter dist=1.0.

The weights $\{\rho_k(x)\}$ are shown in Fig. 6. This is done as follows. Assume that the vectorial representations for the colors red, yellow, and green in the visualization tool are $\boldsymbol{v}_R$, $\boldsymbol{v}_Y$, and $\boldsymbol{v}_G$, respectively. Then the color vector for the node x is given by the weighted average $\rho_R(x)\boldsymbol{v}_R+\rho_Y(x)\boldsymbol{v}_Y+\rho_G(x)\boldsymbol{v}_G$. This shows more clearly the transition between different communities. In particular, the nodes {4, 6, 9, 11, 18:20, 25, 31:32, 37:38} show clearly transitional behavior. If we further partition by the majority rule, namely, cluster the nodes according to their maximum weight, AIP and ETSD give almost the same result except for node 31 (the figures are not shown here). From Table IV, we see that node 31 has almost equal weight of belonging to the green or red clusters.

Next we take n=400, where nodes 1:100 are in group 1, nodes 101:250 in group 2, and nodes 251:400 in group 3. This means approximately q_1=100/400, q_2=q_3=150/400. The other model parameters are chosen as

$$\boldsymbol{u}_1 = (1.0,4.0)^T, \boldsymbol{u}_2 = (2.5,5.5)^T, \boldsymbol{u}_3 = (0.5,6.0)^T, \tag{29a}$$

$$\Sigma_1 = \Sigma_2 = \Sigma_3 = \begin{pmatrix} 0.15 & 0 \\ 0 & 0.15 \end{pmatrix}. \tag{29b}$$

Here we take dist=0.8. Then we generate the network and perform clustering using the methods proposed here. The

TABLE III. The number of iterations, the value of the objective function $J_{\min}$, and the mean and maximum L^{∞} error of ρ for AIP and ETSD compared with the traditional fuzzy c-means algorithm and the priori probability for the sample network with 40 nodes generated from the 3-Gaussian mixture model.

	Iterstep	$J_{\min}$	E_{ρ}^{m} [a]	E_{ρ}^{∞} [b]	$\bar{E}_{\rho}^{m}$ [c]	$\bar{E}_{\rho}^{\infty}$ [d]
AIP	27	1.1554	0.0810	0.2143	0.0628	0.3984
ETSD	859	1.1557	0.0821	0.2130	0.0628	0.4015

[a] The mean L^{∞} error: $\frac{1}{n}\Sigma_{i=1}^{n}\|\rho(\boldsymbol{x}_i)-\rho^{\mathrm{FCM}}(\boldsymbol{x}_i)\|_{\infty}$.

[b] The maximal L^{∞} error: $\max_i\|\rho(\boldsymbol{x}_i)-\rho^{\mathrm{FCM}}(\boldsymbol{x}_i)\|_{\infty}$.

[c] The mean L^{∞} error: $\frac{1}{n}\Sigma_{i=1}^{n}\|\rho(\boldsymbol{x}_i)-\rho^{\mathrm{priori}}(\boldsymbol{x}_i)\|_{\infty}$.

[d] The maximal L^{∞} error: $\max_i\|\rho(\boldsymbol{x}_i)-\rho^{\mathrm{priori}}(\boldsymbol{x}_i)\|_{\infty}$.

TABLE IV. The nodes that have intermediate weights of belonging to different clusters in the 3-Gaussian mixture model. ρ_G, ρ_R, and ρ_W are the weights of belonging to green, red, or yellow clusters, respectively. The other nodes have weights 0 or 1. Nodes 1:3,5,7,8,10 have weight $\rho_R=1$. Nodes 12:14,16,17,21:24 have weight $\rho_Y=1$. Nodes 26,28,29,33,35,36,39,40 have weight $\rho_G=1$. The tolerance $E_{\text{tol}}=10^{-6}$ for both methods. The stepsize of ETSD is $\alpha=26.0$.

Nodes		4	6	9	11	15	18	19	20	25	27	30	31	32	34	37	38
AIP	ρ_G	0.0944	0.0987	0.0757	0	0.0160	0.1509	0.1811	0.5965	0.2222	0.9980	0.9980	0.4994	0.7941	0.9981	0.6152	0.6943
	ρ_R	0.8247	0.7392	0.9243	0.0417	0	0.1275	0	0	0.1805	0.0020	0.0020	0.5006	0.0084	0.0019	0.3098	0.2114
	ρ_Y	0.0809	0.1621	0	0.9583	0.9840	0.7216	0.8189	0.4035	0.5973	0	0	0	0.1975	0	0.0750	0.0943
ETSD	ρ_G	0.1124	0.1169	0.0916	0.0026	0.0054	0.1646	0.1764	0.5977	0.2386	0.9977	0.9977	0.5147	0.8022	0.9981	0.6351	0.7138
	ρ_R	0.7985	0.7122	0.9051	0.0301	0.0015	0.1069	0.0019	0.0019	0.1563	0.0011	0.0011	0.4833	0.0032	0.0012	0.2845	0.1873
	ρ_Y	0.0891	0.1709	0.0033	0.9673	0.9931	0.7285	0.8217	0.4004	0.6051	0.0012	0.0012	0.0020	0.1945	0.0007	0.0804	0.0989

numerical results are shown in Table V and in Fig. 7. The partition obtained by the majority rule gives the same results for AIP and ETSD in this sample.

C. *Ad hoc* networks with 1280 nodes

Our third example is the *ad hoc* network with 1280 nodes. The *ad hoc* network is a benchmark problem used in many papers [2,3,6,8]. It has a known community structure and is constructed as follows. Suppose we choose $n=1280$ nodes, split them into four communities with 320 nodes each. Assume that pairs of nodes belonging to the same communities are linked with probability p_{in} and pairs belonging to different communities with probability p_{out}. These values are chosen so that the average node degree d is fixed at $d=160$. In other words, p_{in} and p_{out} are related as

$$319p_{\text{in}} + 960p_{\text{out}} = 160. \tag{30}$$

We will denote $S_1=\{1:320\}$, $S_2=\{321:640\}$, $S_3=\{641:960\}$, and $S_4=\{961:1280\}$. To test on a more diffusive network, we take $z_{\text{out}}=960p_{\text{out}}=80$. The numerical results are shown in Table VI and in Fig. 8. In Table VI, we compare $\rho_k(x)$ with an interesting quantity, the degree fraction $\tilde{\rho}_k(x)$ which is defined as

$$\tilde{\rho}_k(x) = \frac{E_k(x)}{d(x)}, \quad k = 1, \ldots, 4, \quad x \in S, \tag{31}$$

where $E_k(x)$ is the number of nodes that are connected with x and lie in group S_k. Thus, we have $\Sigma_{k=1}^{4} E_k(x)=d(x)$. With this definition, $\tilde{\rho}_k(x)$ means the fraction of the edges connected with the node x in the kth community. Note that this is not the same as the clustering probability, even though it is an interesting quantity to be compared with. We found that the deviation between these two is about 0.2. Let us also remark that the iteration number for ETSD is less than that for AIP in this example though the final accuracy is not better.

In Fig. 8 we plot the probability distribution function (pdf) of ρ_k and $\tilde{\rho}_k$ $(k=1,2,3,4)$. We observe that the shapes of the pdf for ρ_k or $\tilde{\rho}_k$ are almost the same. Note that all the ρ_k's have a lower peak centered at about 0.7, which corresponds to the nodes in this community, and a higher peak centered at about 0.1, which corresponds to the other nodes outside of this community. The case for $\tilde{\rho}_k$ is similar but with the lower peak centered at about 0.5 and the higher peak centered at about 0.5/3. We note here that the center 0.5 exactly corresponds to the choice of the parameters $z_{\text{out}}/d=0.5$. If we partition the network using the majority rule, we obtain the four-group partition exactly for this model. This also verifies the accuracy of our algorithms; but our algorithm gives more detailed information for each node.

D. Determination of the number of communities

So far, we have assumed that the number of communities N was given. In many applications, this number is not known beforehand and needed to be determined. Suppose we have an optimal number of partitions N_0 for a fixed network. Naively, we may expect that when we artificially choose the number of communities bigger, say $N>N_0$, the fuzzy clustering weights will tend to a common limit $\rho(x)$ for each node x. That is, the components of $\rho(x)$ corresponding to the ghost communities will be zero. However, this is not true for the current model. Suppose we have already obtained ρ and $\hat{p}$ when $N=N_0$, now we choose a larger N and make the following extensions. We extend the value of $\rho_k(x)$ to zero in the new added communities and $\hat{p}$ by an $N-N_0$-dimensional identity matrix. With this extended ρ, $\hat{p}$, and community number N, the objective function value J will be equal to the value when $N=N_0$ if we ignore the singularity of $\hat{\mu}$ ($\hat{\mu}_k=0$ in

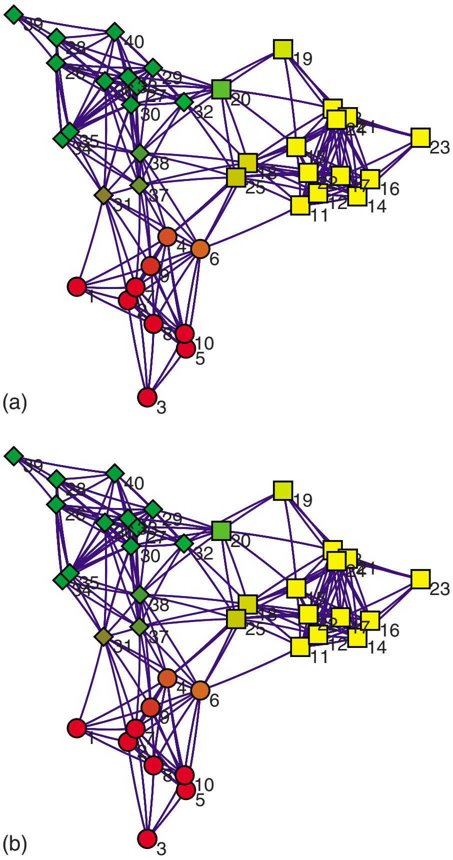

FIG. 6. (Color online) The visualization of the weights $\{\rho_k(x)\}$. The color vector for each node is given by the weighted average $\rho_R \boldsymbol{v}_R + \rho_Y \boldsymbol{v}_Y + \rho_G \boldsymbol{v}_G$, where $\boldsymbol{v}_R, \boldsymbol{v}_Y, \boldsymbol{v}_G$ are the vector representations for the colors red, yellow, and green, respectively. The panels (a) and (b) show the results by using AIP and ETSD, respectively. The nodes {4, 6, 9, 11, 18:20, 25, 31:32, 37:38} have observable transition colors, and they play the role of transition nodes in the network.

the ghost communities). This can be easily seen from Eqs. (7) and (8). The minimization results by k-means and AIP algorithm are shown in Fig. 9. As the prescribed number of communities is increased, the minimized objective function value J is also decreased (the minimized J is obtained by using initial values from the k-means algorithm). This is similar to the case in the k-means algorithm [6]. In fact even for nodes in Euclidean space, one cannot simply use the fuzzy c-means algorithm to do model selection [26].

V. CONCLUSIONS

We have presented a probabilistic framework for network partition, which can be considered as a natural extension of either the fuzzy c-means algorithm in statistics to network partitioning, or the deterministic framework for optimal network partition presented in [6]. Two algorithms, the AIP and the ETSD, are proposed and successfully applied to three representative examples. Our numerical results show that they produce similar results, but the AIP algorithm has better efficiency and accuracy.

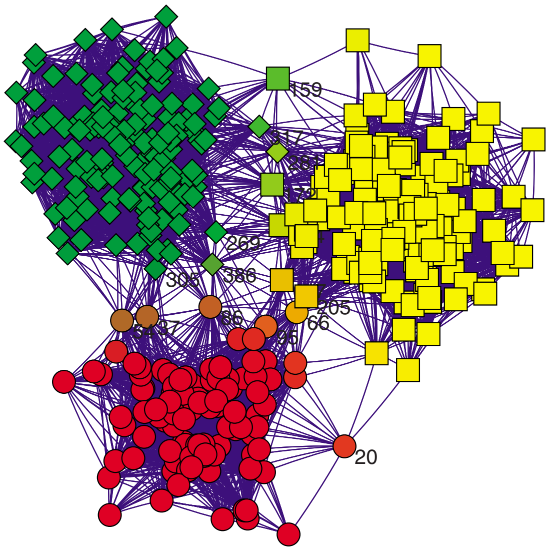

FIG. 7. (Color online) The visualization of the weights $\{\rho_k(x)\}$ obtained with AIP for 400 nodes from 3-Gaussian mixture model. The color vector for each node is given by the weighted average as in Fig. 6. The nodes {20, 37, 54, 66, 86, 95, 104, 147, 159, 172, 205, 269, 281, 305, 317, 386} have more diffusive weights than the others, which show transition colors in the figure.

The probabilistic framework outlined here is a much more mature way of discussing network partition. More importantly, it has more predictive power than the old way of doing network partition. One could imagine, for example, using the algorithms discussed here on the voting recording of the U.S. senators, and predict who is most likely to switch parties.

One open question is how to determine the number of clusters to begin with. This is a generic issue in network partition. We are investigating a number of strategies, but clearly help is needed on this issue.

ACKNOWLEDGMENTS

We are very grateful to Eric Vanden-Eijnden for many discussions. This is part of the joint project with Eric and his comments have always been very helpful. The work of T.L.

TABLE V. The number of iterations, the value of the objective function $J_{\min}$, and the mean and maximum L^∞ error of ρ compared with the traditional FCM and the priori probabilities for the sample network with 400 nodes generated from the 3-Gaussian mixture model.

	Iterstep	$J_{\min}$	E_ρ^m	E_ρ^∞	$\bar{E}_\rho^m$	$\bar{E}_\rho^\infty$
AIP	16	1.7942	0.1037	0.3837	0.0116	0.2243
ETSD	104	1.7962	0.1014	0.4045	0.0126	0.3193

TABLE VI. The results obtained for the *ad hoc* network with 1280 nodes and $z_{\mathrm{out}}=80$. The tolerance is $E_{\mathrm{tol}}=10^{-6}$ in both methods. The differences between ρ and $\tilde{\rho}$ [the fraction of the edges defined in Eq. (31)] are shown in the last two columns.

	Iterstep	$J_{\min}$	E_ρ^m	E_ρ^∞
AIP	907	6.603824	0.182283	0.269223
ETSD	494	6.604187	0.182256	0.266623

and J.L. is supported by the Natural Science Foundation of China under Grant No. 10871010 and the National Basic Research Program of China under Grant No. 2005CB321704. The work of W.E is partially supported by Army Research Office under Grant No. W911NF-07-1-0637 and the Natural Science Foundation of China under Grant No. 10871010.

APPENDIX: DERIVATION OF EQS. (13)

To derive the Euler-Lagrange equations of problem (9), we first take the variation in J with respect to $\hat{p}_{kl}$. We have

$$\frac{\partial J}{\partial \hat{p}_{kl}} = -2\sum_{x,y\in S}\mu(x)\mu(y)\left[\sum_{m,n=1}^{N}\rho_m(x)\rho_n(y)\left(\frac{p(x,y)}{\mu(y)} - \frac{\hat{p}_{mn}}{\hat{\mu}_n}\right)\right]\cdot\left[\sum_{s,t=1}^{N}\rho_s(x)\rho_t(y)\frac{1}{\hat{\mu}_t}\delta_{ks}\delta_{lt}\right]=0.$$

After suitable manipulations, we obtain

$$\sum_{x,y\in S}\sum_{m,n=1}^{N}\mu(x)\mu(y)\rho_m(x)\rho_n(y)\frac{\hat{p}_{mn}}{\hat{\mu}_n}\rho_k(x)\rho_l(y) = \sum_{x,y\in S}\mu(x)p(x,y)\rho_k(x)\rho_l(y) = \hat{\mu}_k\hat{p}^*_{kl}. \tag{A1}$$

Representing the above result with matrix form gives Eq. (13a).

We can prove that $\hat{p}$ is a stochastic matrix from Eq. (13a). To do this, we should note that

$$\hat{\mu}\cdot\mathbf{1}_{N\times 1}=\hat{\mu}, \quad I_{\hat{\mu}}\cdot\mathbf{1}_{N\times 1}=\hat{\mu},$$

where $\mathbf{1}_{N\times 1}$ means the N by 1 vector with all entries equal to 1. Now it is straightforward to obtain the following:

$$\hat{p}\cdot\mathbf{1}_{N\times 1}=\hat{\mu}^{-1}I_{\hat{\mu}}\hat{p}^*\hat{\mu}^{-1}I_{\hat{\mu}}\cdot\mathbf{1}_{N\times 1}=\mathbf{1}_{N\times 1},$$

if $\hat{\mu}$ is invertible.

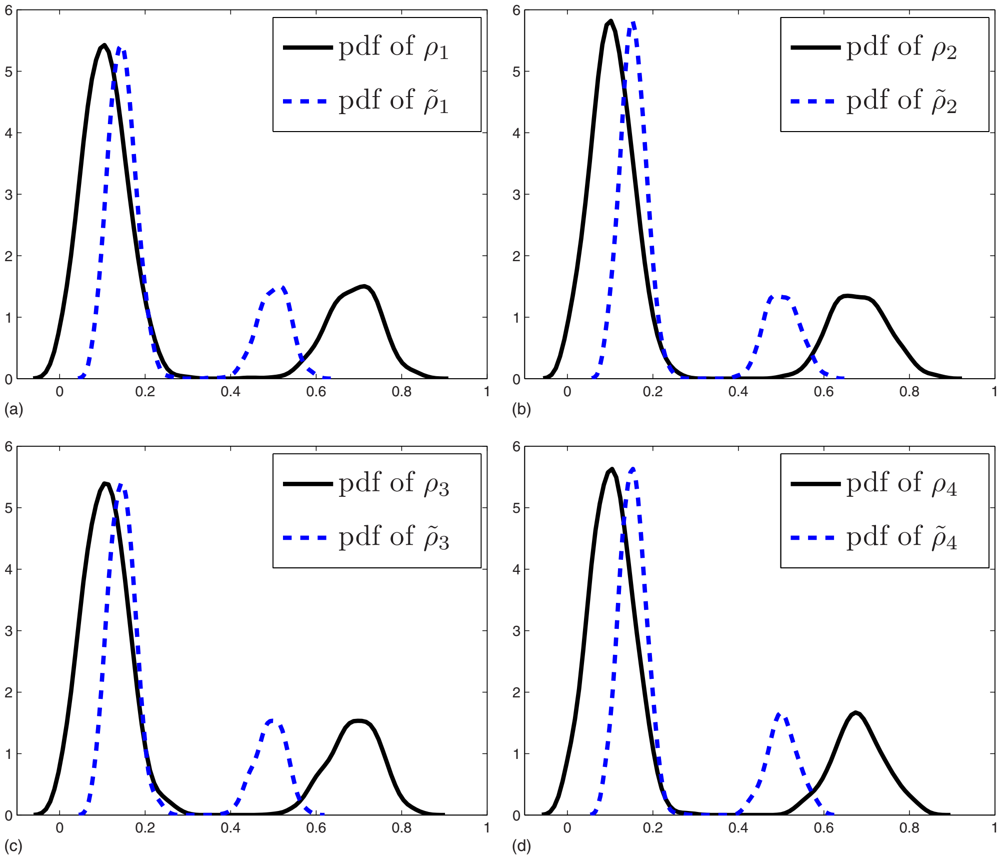

FIG. 8. (Color online) The panels (a)–(d) show the pdf of ρ_k and $\tilde{\rho}_k$ ($k=1,2,3,4$) for the *ad hoc* network with 1280 nodes. The solid and dashed lines represent the pdf of ρ_k and $\tilde{\rho}_k$, respectively. In each figure, the lower peak corresponds to the nodes in this community and the higher peak corresponds to the other nodes outside of the community.

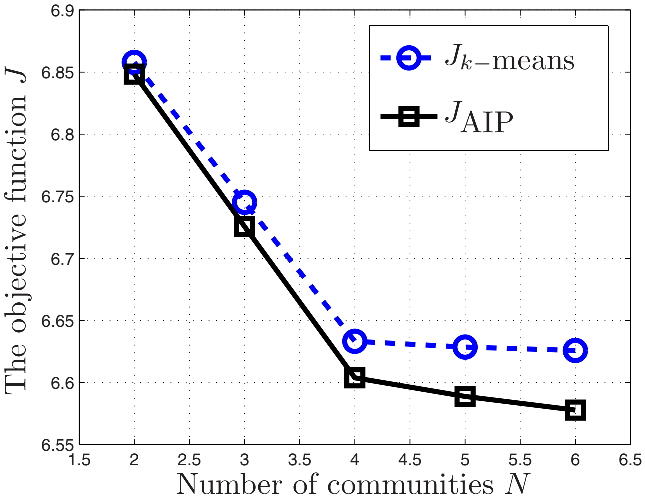

FIG. 9. (Color online) The minimized objective function J versus the number of communities. The dashed line and circles correspond to the result using the k-means algorithm. The solid line and squares correspond to the result using AIP. One can see that the minimum objective function is decreased with the increasing number of communities, and the final values of J obtained by AIP are less than those by k means.

We can also prove that $\hat{p}$ satisfies the detailed balance condition with respect to $\hat{\mu}$,

$$\hat{p}\cdot\Gamma_{\hat{\mu}}^{-1}=\Gamma_{\hat{\mu}}^{-1}\hat{p}^{T}, \tag{A2}$$

under the condition that $\hat{p}^{*}$ satisfies the detailed balance condition with respect to $\hat{\mu}$.

Now we take the variation in J with respect to $\rho_r(z)$ under the normalization condition $\Sigma_{m=1}^{N}\rho_m(x)=1$. We define the extended objective function with Lagrange multipliers $\lambda(x)$

$$\tilde{J}=J+\sum_{x\in S}\lambda(x)\left[\sum_{m=1}^{N}\rho_m(x)-1\right]. \tag{A3}$$

The variation in $\tilde{J}$ with respect to $\rho_r(z)$ gives

$$\begin{aligned}&\sum_{y\in S}\sum_{k,l=1}^{N}\sum_{n=1}^{N}\mu(y)\rho_k(z)\rho_l(y)\rho_n(y)\left(\frac{p(z,y)}{\mu(y)}-\frac{\hat{p}_{kl}}{\hat{\mu}_l}\right)\\&\quad\times\left(\frac{p(z,y)}{\mu(y)}-\frac{\hat{p}_{rn}}{\hat{\mu}_n}\right)+\sum_{y\in S}\sum_{k,l=1}^{N}\sum_{n=1}^{N}\mu(y)\rho_k(z)\rho_l(y)\rho_n(y)\\&\quad\times\left(\frac{p(y,z)}{\mu(z)}-\frac{\hat{p}_{lk}}{\hat{\mu}_k}\right)\left(\frac{p(y,z)}{\mu(z)}-\frac{\hat{p}_{nr}}{\hat{\mu}_r}\right)\\&\quad+\sum_{x,y\in S}\sum_{k,l=1}^{N}\sum_{n=1}^{N}\mu(x)\mu(y)\rho_k(x)\rho_l(y)\rho_n(x)\rho_r(y)\\&\quad\times\left(\frac{p(x,y)}{\mu(y)}-\frac{\hat{p}_{kl}}{\hat{\mu}_l}\right)\frac{\hat{p}_{nr}}{\hat{\mu}_r^2}=-\frac{\lambda(z)}{2\mu(z)}.\end{aligned} \tag{A4}$$

We simply denote the above formula as

$$P_1+P_2+P_3=-\frac{\lambda(z)}{2\mu(z)}.$$

We have

$$P_3=\sum_{n=1}^{N}\hat{p}_{nr}^{*}\hat{\mu}_n\frac{\hat{p}_{nr}}{\hat{\mu}_r^2}-\sum_{k,l=1}^{N}\sum_{n=1}^{N}\hat{\hat{\mu}}_{nk}\hat{\hat{\mu}}_{lr}\frac{\hat{p}_{kl}}{\hat{\mu}_l}\frac{\hat{p}_{nr}}{\hat{\mu}_r^2}. \tag{A5}$$

With the derived Eq. (13a), we actually have $P_3=0$!. Furthermore, we have

$$\begin{aligned}P_1=\mathbf{1}_{N\times1}\cdot\mathrm{diag}_{mv}\{&p^2\cdot\Gamma_{\mu}^{-1}-p\cdot\rho^T\cdot\Gamma_{\hat{\mu}}^{-1}\cdot\hat{p}^T\cdot\rho\}\\&-\hat{p}\cdot\Gamma_{\hat{\mu}}^{-1}\cdot\rho\cdot p^T+\hat{p}\cdot\Gamma_{\hat{\mu}}^{-1}\cdot\hat{\hat{\mu}}\cdot\Gamma_{\hat{\mu}}^{-1}\cdot\hat{p}^T\cdot\rho.\end{aligned} \tag{A6}$$

$$\begin{aligned}P_2=\mathbf{1}_{N\times1}\cdot\mathrm{diag}_{mv}\{&p^2\cdot\Gamma_{\mu}^{-1}-p\cdot\rho^T\cdot\hat{p}\cdot\Gamma_{\hat{\mu}}^{-1}\cdot\rho\}\\&-\Gamma_{\hat{\mu}}^{-1}\cdot\hat{p}^T\cdot\rho\cdot p^T+\Gamma_{\hat{\mu}}^{-1}\cdot\hat{p}^T\cdot\hat{\hat{\mu}}\cdot\hat{p}\cdot\Gamma_{\hat{\mu}}^{-1}\cdot\rho.\end{aligned} \tag{A7}$$

Here the symbol $\mathrm{diag}_{mv}\{A\}$ is the matrix-to-vector operator, which extracts the diagonals of the matrix A. With condition (A2), we obtain

$$\begin{aligned}P_1=P_2=\mathbf{1}_{N\times1}\cdot\mathrm{diag}_{mv}\{&p^2\cdot\Gamma_{\mu}^{-1}-p\cdot\rho^T\cdot\hat{p}\cdot\Gamma_{\hat{\mu}}^{-1}\cdot\rho\}\\&-\hat{p}\cdot\Gamma_{\hat{\mu}}^{-1}\cdot\rho\cdot p^T+\hat{p}\cdot\Gamma_{\hat{\mu}}^{-1}\cdot\hat{\hat{\mu}}\cdot\hat{p}\cdot\Gamma_{\hat{\mu}}^{-1}\cdot\rho.\end{aligned}$$

After suitable manipulations, we obtain

$$\begin{aligned}\rho=-\hat{\mu}\cdot\Bigg[&\mathrm{diag}_{mv}\{p^2\cdot\Gamma_{\mu}^{-1}-p\cdot\rho^T\cdot\hat{p}\cdot\Gamma_{\hat{\mu}}^{-1}\cdot\rho\}\\&+\frac{1}{2}\mathrm{diag}_{mv}\{I_{\lambda}\cdot\Gamma_{\mu}^{-1}\}\Bigg]+I_{\hat{\mu}}\hat{p}^{-1}\hat{\hat{\mu}}^{-1}\rho p^T.\end{aligned} \tag{A8}$$

With the normalization condition of ρ, we set the Lagrange multiplier

$$\lambda(z)=\mu(z)\sum_{y\in S}\sum_{k,l=1}^{N}\rho_k(z)\rho_l(y)p(z,y)\frac{\hat{p}_{kl}}{\hat{\mu}_l}-\sum_{y\in S}p(z,y)p(y,z). \tag{A9}$$

Substituting Eq. (A9) into Eq. (A8), we obtain the equation for ρ, finally,

$$\rho=I_{\hat{\mu}}\hat{p}^{-1}\hat{\hat{\mu}}^{-1}\rho p^T. \tag{A10}$$

[1] M. Girvan and M. Newman, Proc. Natl. Acad. Sci. U.S.A. **99**, 7821 (2002).

[2] M. E. J. Newman, Phys. Rev. E **69**, 066133 (2004).

[3] M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).

[4] M. Newman, Proc. Natl. Acad. Sci. U.S.A. **103**, 8577 (2006).

[5] M. Newman and E. Leicht, Proc. Natl. Acad. Sci. U.S.A. **104**, 9564 (2007).

[6] W. E, T. Li, and E. Vanden-Eijnden, Proc. Natl. Acad. Sci. U.S.A. **105**, 7907 (2008).

[7] J. M. Hofman and C. H. Wiggins, Phys. Rev. Lett. **100**, 258701 (2008).

[8] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, J. Stat. Mech.: Theory Exp. (2005) P09008.

[9] H. Zhou, Phys. Rev. E **67**, 061901 (2003).

[10] F. Wu and B. Huberman, Eur. Phys. J. B **38**, 331 (2004).

[11] F. Radicchi, C. Castellano, F. Cecconi, and V. Loreto, Proc. Natl. Acad. Sci. U.S.A. **101**, 2658 (2004).

[12] A. Chorin, A. Kast, and R. Kupferman, Commun. Pure Appl. Math. **52**, 1231 (1999).

[13] A. Chorin, Multiscale Model. Simul. **1**, 105 (2003).

[14] L. Lovász, in *Combinatorics* edited by D. Miklós, V. T. Sós, T. Szönyi (Janos Bolyai Mathematical Society, Budapest, 1993), Vol. 2, p. 1.

[15] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, New York, 2001).

[16] J. Shi and J. Malik, IEEE Trans. Pattern Anal. Mach. Intell. **22**, 888 (2000).

[17] M. Meilă and J. Shi, *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics* (Kaufmann, San Francisco, 2001), pp. 92–97.

[18] S. Lafon and A. Lee, IEEE Trans. Pattern Anal. Mach. Intell. **28**, 1393 (2006).

[19] J. Dunn, Cybern. Syst. **3**, 32 (1973).

[20] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms* (Plenum Press, New York, 1981).

[21] F. Chung, *Spectral Graph Theory* (American Mathematical Society, Rhode Island, 1997).

[22] D. Gfeller and P. De Los Rios, Phys. Rev. Lett. **99**, 038701 (2007).

[23] D. Gfeller and P. De Los Rios (unpublished).

[24] W. Zachary, J. Anthropol. Res. **33**, 452 (1977).

[25] M. Penrose, *Random Geometric Graphs* (Oxford University Press, Oxford, 2003).

[26] X. Xie and G. Beni, IEEE Trans. Pattern Anal. Mach. Intell. **13**, 841 (1991).