Pattern Recognition 43 (2010) 1334–1345

Contents lists available at ScienceDirect

Pattern Recognition

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

Detecting the fuzzy clusters of complex networks

Jian Liu

LMAM and School of Mathematical Sciences, Peking University, Beijing 100871, China

ARTICLE INFO

Article history:
Received 24 March 2009
Received in revised form
22 September 2009
Accepted 9 November 2009

Keywords:
Fuzzy clusters
Optimal prediction
k-Means
Fuzzy *c*-means
Steepest descent
Conjugate gradient
Projection

ABSTRACT

To find the best partition of a large and complex network into a small number of clusters has been addressed in many different ways. However, the probabilistic setting in which each node has a certain probability of belonging to a certain cluster has been scarcely discussed. In this paper, a fuzzy partitioning formulation, which is extended from a deterministic framework for network partition based on the optimal prediction of a random walker Markovian dynamics, is derived to solve this problem. The algorithms are constructed to minimize the objective function under this framework. It is demonstrated by the simulation experiments that our algorithms can efficiently determine the probabilities with which a node belongs to different clusters during the learning process. Moreover, they are successfully applied to two real-world networks, including the social interactions between members of a karate club and the relationships of some books on American politics bought from Amazon.com.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

In recent years an explosive growth of interest and activity on the structure and dynamics of complex networks [1–3] has appeared. This is partly due to the influx of new ideas, particularly ideas from statistical mechanics, to the subject, and partly due to the emergence of interesting and challenging new examples of complex networks such as the internet and wireless communication networks. Network models have also become popular tools in social science, economics, the design of transportation and communication systems, banking systems, powergrid, etc., due to our increased capability of analyzing the models. On a related but different front, recent advances in computer vision and data mining have also relied heavily on the idea of viewing a data set or an image as a graph or a network, in order to extract information about the important features of the images or more generally, the data sets [4,5].

To give a coarse definition about the study of complex networks from the viewpoints of applied mathematics, it is about the research of dynamical systems on graphs. The graph structure may be fixed, or time-varying; the dynamical system may be deterministic, or stochastic. Since these networks are typically very complex, it is of great interest to see whether they can be reduced to much simpler systems. Such issues have been addressed before. In particular, much effort has gone into partitioning the network into a small number of clusters [4–14]. And in a broader aspect, it is also closely related to the model reduction theory of differential equations [15]. These proposals in the literature are constructed from different viewing angles, and their numerical performance applied to a benchmark model—the ad hoc network with 128 nodes and known community structures—are summarized in [16].

In a previous paper [12], a *k*-means approach is proposed to partition the networks based on optimal prediction theory proposed by Chorin and coworkers [17,18]. The basic idea is to associate the network with the random walker Markovian dynamics [19], then introduce a metric on the space of Markov chains (stochastic matrices), and optimally reduce the chain under this metric. The final minimization problem is solved by an analogy to the traditional *k*-means algorithm [20,21] in clustering analysis. This approach is motivated by the diffusion maps [11] and MNCut algorithms in imaging science [4].

The current paper is along the lines of extending the *k*-means type clustering techniques to the partitioning of networks. In statistical literature, a widely used generalization of *k*-means algorithm is the fuzzy *c*-means (FCM) algorithm [22,23]. In this framework, each node has a certain probability of belonging to a certain cluster, instead of assigning nodes to specific clusters, which is called fuzzy clustering in some papers [14]. This idea is quite valuable since usually it is not well separated for most of networks and the extending fuzzy partitioning framework seems extremely meaningful [13]. For the nodes lying in the transition domain between different clusters, the fuzzy partition will be more acceptable. To obtain the hard clustering result, one only needs to threshold the weights. But the fuzzy clustering presents more detailed information than the hard one, and it gives more reasonable explanations in some cases.

E-mail address: dugujian@pku.edu.cn

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.11.007

We constructed two algorithms—the steepest descent method with projection (SDP) and the reduced conjugate gradient method with projection (CGP)—from minimizing the objective function under the generalized framework in this paper. According to two choices of projection operators P1, P2, we obtain the formulations—SDP1, SDP2, CGP1, CGP2—which have been applied to two artificial networks, including the ad hoc network and the sample network generated from Gaussian mixture model, as well as two real-world networks, including the karate club network and the political books network. The proposed algorithms are easy to be implemented with reasonable computational effort and the final results do make sense in the considered models. It is demonstrated by these experiments that the algorithms can always perform successfully during the learning process and lead to a good clustering result.

The rest of the paper is organized as follows. In Section 2, the hard partitioning framework based on the optimal prediction in [12] is briefly introduced, and the corresponding fuzzy partitioning formulation is derived. The algorithms, SDP and CGP, are described in detail in Section 3. Several simulation and practical experiments are conducted in Section 4 to demonstrate the efficiency of the proposed algorithms. The numerical results and performance are typically compared. Finally, we conclude the paper in Section 5. All details of the derivation are left in the Appendix.

2. Framework for fuzzy clustering of networks

In [12], a new strategy for reducing the random walker Markovian dynamics based on optimal prediction theory [17,18] is proposed. Let $G(S,E)$ be a network with N nodes and M edges, where S is the nodes set, $E=\{e(x,y)\}_{x,y\in S}$ 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 with entries $p(x,y)$ given by

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

where $d(x)$ is the degree of the node x [11,19,24]. This Markov chain has stationary distribution

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

and it satisfies the detailed balance condition

$$\mu(x)p(x,y)=\mu(y)p(y,x). \tag{3}$$

The basic idea in [12] is to introduce a metric for the stochastic matrix $p(x,y)$

$$\|p\|_{\mu}^2=\sum_{x,y\in S}\frac{\mu(x)}{\mu(y)}|p(x,y)|^2 \tag{4}$$

and find the reduced Markov chain $\tilde{p}$ by minimizing the distance $\|\tilde{p}-p\|_{\mu}$. For a given partition of S as $S=\bigcup_{k=1}^{K}S_k$ with $S_k\cap S_l=\emptyset$ if $k\neq l$, let $\hat{p}_{kl}$ be the coarse grained transition probability from S_k to S_l on the state space $\mathbb{S}=\{S_1,\dots,S_K\}$ which naturally satisfies

$$\hat{p}_{kl}\geq 0 \quad\text{and}\quad \sum_{l=1}^{K}\hat{p}_{kl}=1. \tag{5}$$

This matrix can be naturally lifted to the space of stochastic matrices on the original state space S via

$$\tilde{p}(x,y)=\sum_{k,l=1}^{K}\mathbf{1}_{S_k}(x)\hat{p}_{kl}\mu_l(y), \tag{6}$$

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

$$\mu_k(x)=\frac{\mu(x)\mathbf{1}_{S_k}(x)}{\hat{\mu}_k},\quad \hat{\mu}_k=\sum_{z\in S_k}\mu(z). \tag{7}$$

Based upon this formulation, we can find the optimal $\hat{p}_{kl}$ for any fixed partition. With this optimal form $\hat{p}_{kl}$, we further search for the best partition $\{S_1,\dots,S_K\}$ with the given number of clusters K by minimizing the optimal prediction error. This is the theoretical basis for constructing the k-means algorithm for network partition in [12].

In the above formulation of hard clustering, each node belongs to only one cluster after the partition. This is often too restrictive for the reason that nodes at the boundary among clusters share commonalities with more than one cluster and play a role of transition in many diffusive networks. This motivates the extension of the optimal partition theory to a probabilistic setting [13]. Here we use the terminology hard clustering since we take indicator function $\mathbf{1}_{S_k}(x)$ in Eq. (6) when the node x belongs to the k-th cluster. Now it is extended to the fuzzy clustering concept where each node may belong to different clusters with nonzero probabilities at the same time. We denote such probability function as $\rho_k(x)$ to represent the probability which the node x belongs to the k-th cluster with. Naturally we need the assumption that

$$\rho_k(x)\geq 0 \quad\text{and}\quad \sum_{k=1}^{K}\rho_k(x)=1 \tag{8}$$

for all $x\in S$.

Similar as before, we define the transition probability matrix of the induced Markov chain as

$$\tilde{p}(x,y)=\sum_{k,l=1}^{K}\rho_k(x)\hat{p}_{kl}\mu_l(y),\quad x,y\in S, \tag{9}$$

where

$$\mu_k(x)=\frac{\rho_k(x)\mu(x)}{\hat{\mu}_k}\quad\text{and}\quad \hat{\mu}_k=\sum_{z\in S}\rho_k(z)\mu(z). \tag{10}$$

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 cluster S_k to cluster S_l with their corresponding belonging probability and stay there in equilibrium state. It is not difficult to show that $\tilde{p}(x,y)$ is indeed a transition probability matrix and satisfies the detailed balance condition with respect to μ

$$\mu(x)\tilde{p}(x,y)=\mu(y)\tilde{p}(y,x) \tag{11}$$

if $\hat{p}_{kl}$ satisfies the detailed balance condition with respect to $\hat{\mu}$, that is

$$\hat{\mu}_k\hat{p}_{kl}=\hat{\mu}_l\hat{p}_{lk}. \tag{12}$$

Given the number of the clusters K, we optimally reduce the random walker dynamics by considering the following minimization problem:

$$\min_{\rho_k(x),\ \hat{p}_{kl}} J=\|p-\tilde{p}\|_{\mu}^2 \tag{13}$$

where

$$J=\sum_{x,y\in S}\frac{\mu(x)}{\mu(y)}|p(x,y)-\tilde{p}(x,y)|^2=\sum_{x,y\in S}\mu(x)\mu(y)\cdot\left(\sum_{m,n=1}^{K}\rho_m(x)\rho_n(y)\frac{\hat{p}_{mn}}{\hat{\mu}_n}-\frac{p(x,y)}{\mu(y)}\right)^2, \tag{14}$$

subject to the constraints Eqs. (5) and (8).

To minimize the objective function J in Eq. (14), we define

$$\hat{p}^*_{kl} = \sum_{x,y \in S} \mu_k(x)p(x,y)\rho_l(y) = \frac{1}{\hat{\mu}_k} \sum_{x,y \in S} \mu(x)\rho_k(x)p(x,y)\rho_l(y) \tag{15}$$

which has the similar form as $\hat{p}_{kl}$ in hard clustering. Then $\hat{p}^*_{kl}$ is indeed a stochastic matrix because $\sum_{z \in S} \mu_k(z) = 1$ for all k. Furthermore we have that $\hat{p}^*_{kl}$ satisfies the detailed balance condition with respect to $\hat{\mu}$, that is

$$\hat{\mu}_k \hat{p}^*_{kl} = \hat{\mu}_l \hat{p}^*_{lk}. \tag{16}$$

With the above background, we have the following basic result.

Lemma 1. *The expressions of $\partial J/\partial \hat{p}$ and $\partial J/\partial \rho$ according to Eq.* (14) *are given by*

$$\frac{\partial J}{\partial \hat{p}} = 2(\hat{\hat{\mu}}\hat{p}I_{\hat{\mu}}^{-1}\hat{\hat{\mu}}I_{\hat{\mu}}^{-1} - I_{\hat{\mu}}\hat{p}^* I_{\hat{\mu}}^{-1}), \tag{17a}$$

$$\begin{aligned}\frac{\partial J}{\partial \rho} = 2[&(\hat{p}I_{\hat{\mu}}^{-1}\hat{\hat{\mu}}I_{\hat{\mu}}^{-1}\hat{p}^T + I_{\hat{\mu}}^{-1}\hat{p}^T\hat{\hat{\mu}}\hat{p}I_{\hat{\mu}}^{-1}) \cdot \rho I_\mu - (\hat{p}I_{\hat{\mu}}^{-1} + I_{\hat{\mu}}^{-1}\hat{p}^T) \\ &\cdot \rho p^T I_\mu - ((I_{\hat{\mu}}^{-2}\hat{p}^T) * (\hat{\hat{\mu}}I_{\hat{\mu}}^{-1}\hat{p}^T\hat{\hat{\mu}})) \\ &\cdot \mathbf{1}_{K\times 1}\mu^T + ((I_{\hat{\mu}}^{-2}\hat{p}^T) * (\hat{p}^*)^T) \cdot \hat{\mu}\mu^T],\end{aligned} \tag{17b}$$

where $\rho = (\rho_k(x))_{k=1,\ldots,K,x \in S}$ is a $K \times N$ matrix and $\hat{\hat{\mu}}$ is a $K \times K$ matrix with entries

$$\hat{\hat{\mu}}_{kl} = \sum_{z \in S} \mu(z)\rho_k(z)\rho_l(z) = (\rho \cdot I_\mu \cdot \rho^T)_{kl}. \tag{18}$$

The diagonal matrices I_μ, $I_{\hat{\mu}}$ are $N \times N$ and $K \times K$, respectively, with entries

$$I_\mu(x,y) = \mu(x)\delta(x,y), \quad x,y \in S, \tag{19a}$$

$$(I_{\hat{\mu}})_{kl} = \hat{\mu}_k \delta_{kl}, \quad k,l = 1,\ldots,K, \tag{19b}$$

where $\delta(x,y)$ and δ_{kl} are both Kronecker delta symbols.

The proof of Lemma 1 can be found in Appendix A. Eqs. (17a) and (17b) give the partial derivatives of the objective function in Eq. (14), which are the critical points of constructing gradient methods.

3. Algorithms

An obvious choice to solve the constrained optimization Eq. (13) is the steepest descent method [21]. However, the components of $\hat{p}$ and ρ may become negative and nonnormalized during the descent procedure for our problems, which makes the probabilistic interpretation useless. To ensure the nonnegativity and normalization conditions for $\hat{p}$ and ρ, we add a projection step after each renewal. This leads to the following Steepest Descent method with Projections (SDP).

Algorithm 1 (*SDP*).

(1) Set up the initial state $\rho^{(0)}$ as the indicator matrix for each node in the network with the k-means algorithm in [12], $n = 0$.

(2) Perform the following iteration until $\|\rho^{(n+1)} - \rho^{(n)}\| \le TOL$:

$$\hat{p}^{(n+1)} = \mathcal{P}\left(\hat{p}^{(n)} - \alpha\frac{\partial J}{\partial \hat{p}}(\hat{p}^{(n)}, \rho^{(n)})\right), \tag{20a}$$

$$\rho^{(n+1)} = \mathcal{P}\left(\rho^{(n)} - \alpha\frac{\partial J}{\partial \rho}(\hat{p}^{(n)}, \rho^{(n)})\right). \tag{20b}$$

Here $\mathcal{P}$ is some type of projection operator which maps a real vector into a probability vector, $\alpha > 0$ is the learning rate and TOL is a prescribed tolerance.

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

Two choices of the projection operator $\mathcal{P}$ are used in our computations, but the final results seem to be insensitive to them. Now suppose $\mathbf{v} = (v_1, v_2, \ldots, v_N) \in \mathbb{R}^N$, and $v_i < 0$ when $i \in \Lambda$, we make projection as any of the following two choices:

P1: Direct projection to the boundary.
When $i \in \Lambda$, we set $\mathcal{P}v_i = 0$; otherwise we set $\mathcal{P}v_i = v_i / \sum_{j \notin \Lambda} v_j$.

P2: Iterative projections.
At first make projection to the hyperplane $\sum_{i=1}^N v_i = 1$, then check each component of the projected $\mathbf{v}$. If $v_{i_0} < 0$, we take $\mathcal{P}v_{i_0} = 0$ and make projection again to a dimension reduced hyperplane $\sum_{i \ne i_0} v_i = 1$. Repeat the projection procedure to a lower and lower dimensional hyperplane until no component is negative.

The application of projection operators for SD can simply solve the original constrained optimization Eq. (13), while to solve it exactly may involve much more complexity. Note that with this SDP method, we cannot guarantee that $\hat{p}$ and ρ are the exact minimizer of the original problem Eq. (13) with nonnegative and normalization constraints, but we take them as an approximate minimizer. The numerical results show that this strategy works fine for many examples.

The learning rate α was usually chosen that started from a reasonable initial value and then reduced to zero with the iteration number n in such a way that $0 \le \alpha(n) \le 1$, and

$$\lim_{n\to\infty} \alpha(n) = 0, \qquad \sum_{n=1}^{\infty} \alpha(n) = \infty. \tag{21}$$

The typical example of such case is $\alpha(n) = \alpha_0/n$, where α_0 is a positive constant [25]. Another choice is to fix the learning rate α as a positive constant [26,27], which we utilize here, since the initial partition is good enough that the objective function Eq. (14) descends much more slowly when the learning rate becomes smaller, while larger values of α cause blow up.

The number of the iteration steps is difficult to be estimated, which may depend on the structure of the network itself, the choice of the initial values, etc. It usually converges fast for well-clustered networks and may converge slowly for diffusive networks.

Now let us estimate the computational cost in each iteration. In the iteration step for $\hat{p}$, all of the matrices are of order $K \times K$ and full according to Eq. (17a). It is easy to find that the computation of $\hat{\mu}$ costs $O(KN)$, and the computation of $\hat{\hat{\mu}}$ costs $O(K^2N)$. Note that the stochastic matrix p is sparse with M entries, so the computation for $\hat{p}^*$ costs $O(K^2M)$, where M represents the number of edges, which is usually assumed $O(N)$ in realistic networks. So finally, we obtain the cost in the step for $\hat{p}$ is $O(K^2(M+N))$. The cost for ρ is also $O(K^2(M+N))$ according to Eq. (17b), since $\hat{p}^*$ is involved in the equations.

Another choice is to minimize the objective function using a simplified formulation of traditional conjugate gradient method, which is frequently used in machine learning [28]. It can be also regard as the above steepest descent method with a nonzero momentum term, which leads to the following reduced Conjugate Gradient method with Projections (CGP).

Algorithm 2 (*CGP*).

(1) Set up the initial state $\rho^{(0)}$ as the indicator matrix for each node in the network with the k-means algorithm in [12], $n = 0$.

(2) Perform the following iteration until $\|\rho^{(n+1)}-\rho^{(n)}\| \le TOL$:

$$\hat{p}^{(n+1)} = \mathcal{P}\left(\hat{p}^{(n)} - \alpha\frac{\partial J}{\partial \hat{p}}(\hat{p}^{(n)},\rho^{(n)}) + \beta(\hat{p}^{(n)}-\hat{p}^{(n-1)})\right), \tag{22a}$$

$$\rho^{(n+1)} = \mathcal{P}\left(\rho^{(n)} - \alpha\frac{\partial J}{\partial \rho}(\hat{p}^{(n)},\rho^{(n)}) + \beta(\rho^{(n)}-\rho^{(n-1)})\right), \tag{22b}$$

Here $\mathcal{P}$ is some type of projection operator which maps a real vector into a probability vector, $\alpha,\beta>0$ is the learning rates and *TOL* is a prescribed tolerance.

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

We again note that this is just a reduced form of conjugate gradient method, and it is demonstrated by simulation experiments that such method performs more efficiently than SD, just like the superiority traditional conjugate gradient has. The learning rates α and β are still chosen as constants by experience due to the same reason mentioned above. The computational cost in each iteration of CGP is the same as SDP, which is also $O(K^2(M+N))$ for both $\hat{p}$ and ρ. Associating the two projections described above with SDP and CGP, respectively, we refer to the derived algorithms: SDP1, SDP2, CGP1, CGP2, as the fuzzy partitioning algorithms for networks.

Table 1
The iterations, the value of the objective function $J_{\min}$ and the mean and maximum L^{∞}- error of ρ defined in Eq. (25) for algorithms.

	Iterations	$J_{\min}$	e_ρ^m	e_ρ^∞
SDP1&CGP1	110&55	5.8794	0.1191	0.2389
SDP2&CGP2	119&54	5.8757	0.1207	0.2374

4. Experimental results

In this section, simulation experiments on artificial networks, including the ad hoc network with 128 nodes and sample network generated from the Gaussian mixture model, are carried out to demonstrate the performance of the proposed algorithms, via comparing the clustering results with some priori quantities. Moreover, the algorithms are applied to two real-world networks, including the social interactions between members of a karate club and the relationships of some books on American politics bought from Amazon.com.

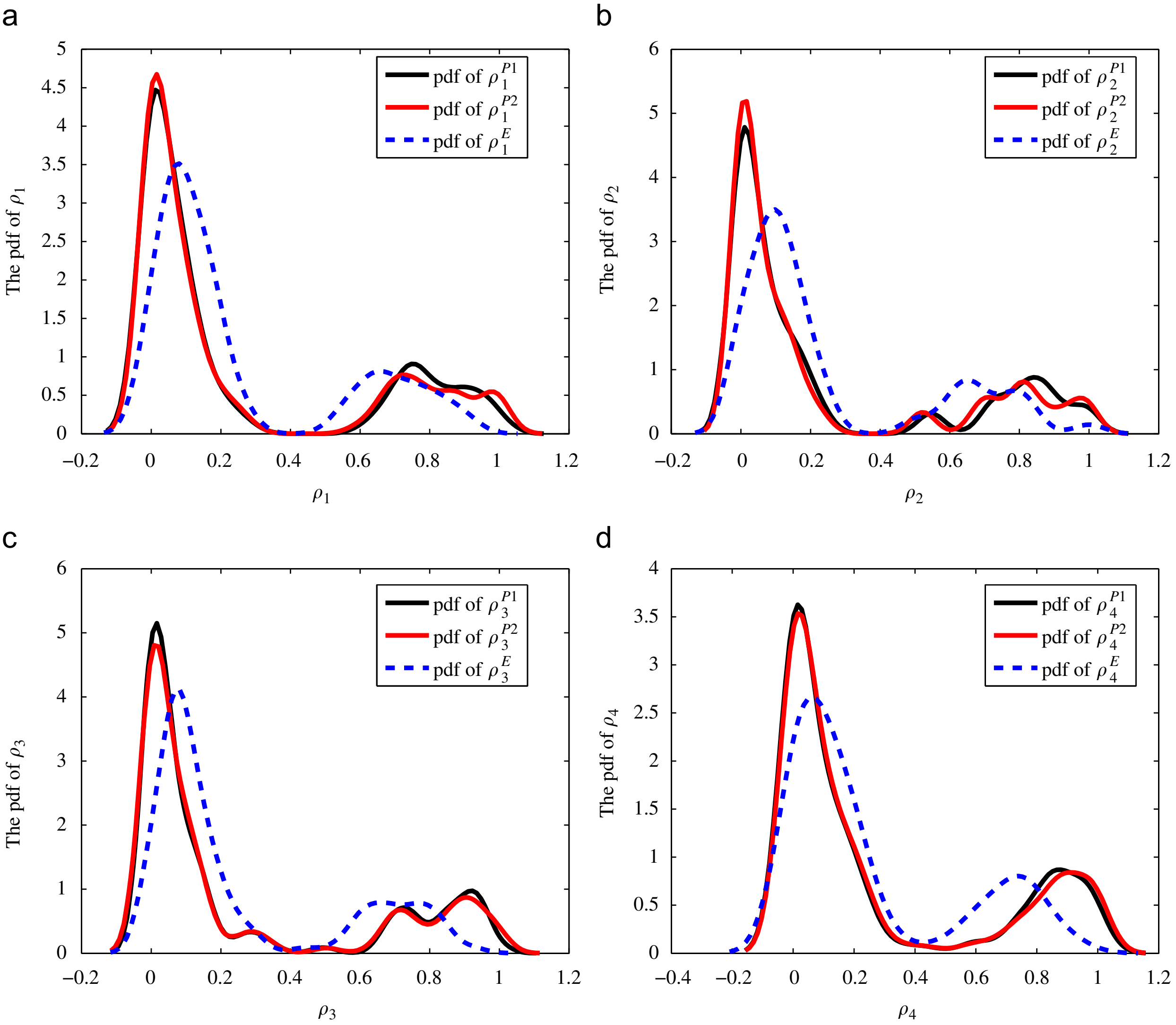

Fig. 1. The pdf of ρ_k and ρ_k^{E} $(k=1,2,3,4)$ for the given ad hoc network with 128 nodes and $z_{\mathrm{out}}=5$. The solid lines and dashed lines represent the pdf of ρ_k and ρ_k^{E}, respectively. In each figure, the lower peak corresponds to the nodes in this cluster, and the higher peak corresponds to the other nodes outside of this cluster.

4.1. Ad hoc network with 128 nodes

The first example is the ad hoc network with 128 nodes. The ad hoc network is a benchmark problem used in many papers [8,9,12,16]. It has a known partition and is constructed as follows. Suppose we choose $N=128$ nodes, split them into four clusters with 32 nodes each. Assume that pairs of nodes belonging to the same clusters are linked with probability p_{in}, and pairs belonging

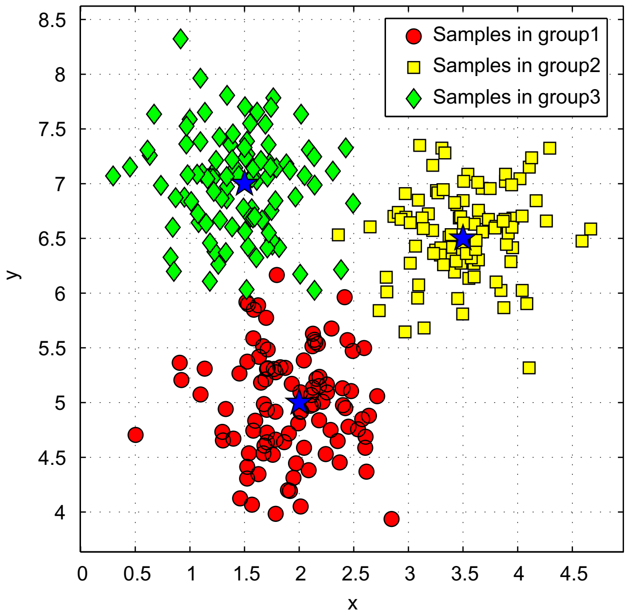

Fig. 2. 300 sample points generated from the given 3-Gaussian mixture distribution. The star symbols represent the centers of each Gaussian component. The circle, square and diamond shaped symbols represent the position of sample points in each component, respectively.

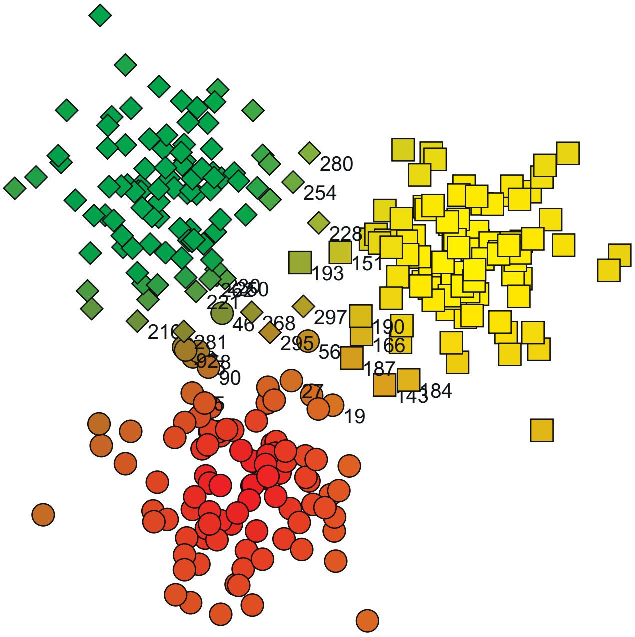

Fig. 4. The visualization of the weights $\{\rho_k^{\mathrm{FCM}}(x)\}$ obtained by traditional FCM for the given sample points in Fig. 2. The nodes which have more diffusive weights than the others are shown in transition colors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

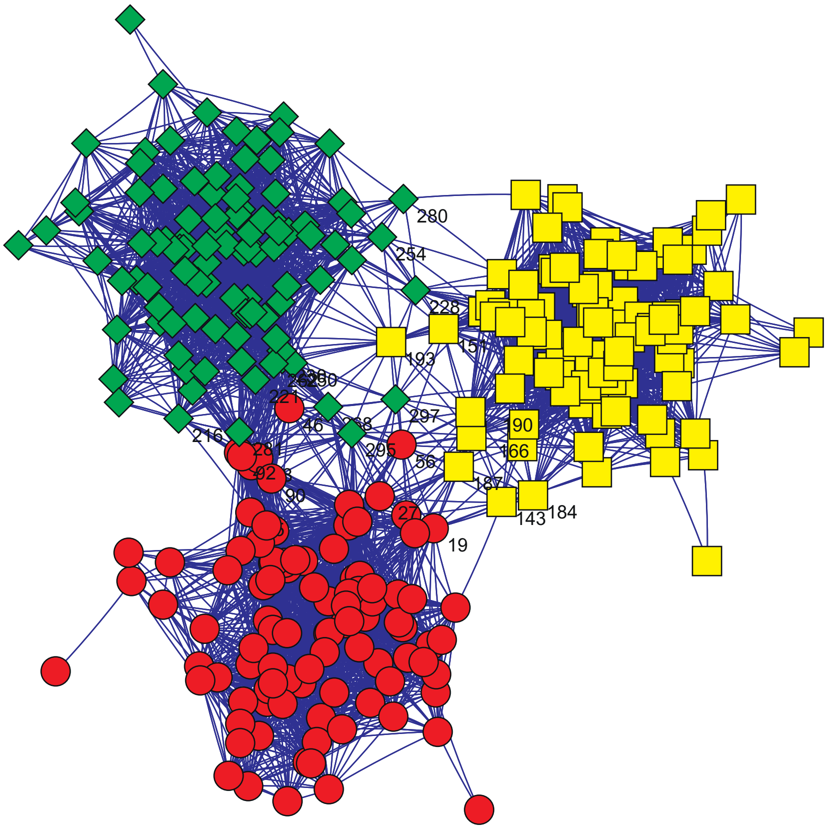

Fig. 3. The network generated from the sample points in Fig. 2 with the parameter $dist=0.7$.

Table 2
The iteration steps, minimized objective function values $J_{\min}$, the mean and maximal L^{∞}- error of ρ for the algorithms compared with the traditional FCM algorithm.

	Iterations	$J_{\min}$	e_{ρ}^{m}	e_{ρ}^{∞}
SDP1&CGP1	36&21	4.0278	0.1050	0.3689
SDP2&CGP2	35&20	4.0276	0.1029	0.3689

Table 3
The association probability that nodes which have intermediate weights belongs to different clusters.

	Nodes	15	19	25	27	29	46	56	58	90	92
SDP1&CGP1	ρ_R	0.8994	0.8529	0.6589	0.8815	0.5108	0.2921	0.5626	0.6158	0.7283	0.5396
	ρ_Y	0	0.1471	0	0.0880	0	0.0232	0.3431	0	0	0
	ρ_G	0.1006	0	0.3411	0.0305	0.4892	0.6847	0.0943	0.3842	0.2717	0.4604
SDP2&CGP2	ρ_R	0.8893	0.8461	0.6494	0.8597	0.5069	0.2896	0.5485	0.6074	0.7150	0.5344
	ρ_Y	0	0.1539	0.0067	0.0986	0.0054	0.0374	0.3474	0.0069	0.0089	0.0062
	ρ_G	0.1107	0	0.3439	0.0417	0.4877	0.6730	0.1041	0.3857	0.2761	0.4594
	Nodes	143	151	166	184	187	190	193	216	221	228
SDP1&CGP1	ρ_R	0.4218	0.0259	0.2034	0.1466	0.5632	0.1331	0.0892	0.1618	0.1184	0
	ρ_Y	0.5782	0.8762	0.7708	0.8534	0.4201	0.8489	0.4027	0	0	0.6282
	ρ_G	0	0.0978	0.0258	0	0.0167	0.0179	0.5081	0.8382	0.8816	0.3718
SDP2&CGP2	ρ_R	0.4217	0.0325	0.2045	0.1490	0.5491	0.1361	0.0938	0.1675	0.1234	0
	ρ_Y	0.5783	0.8606	0.7595	0.8510	0.4215	0.8359	0.4026	0.0012	0.0063	0.6216
	ρ_G	0	0.1069	0.0359	0	0.0295	0.0280	0.5036	0.8313	0.8703	0.3784
	Nodes	230	250	254	262	268	280	281	295	297	
SDP1&CGP1	ρ_R	0.1050	0.1114	0	0.1026	0.3815	0	0.2898	0.5702	0.3567	
	ρ_Y	0.0110	0.0128	0.1826	0.0015	0.0351	0.2151	0	0.1071	0.3554	
	ρ_G	0.8840	0.8758	0.8174	0.8959	0.5834	0.7849	0.7102	0.3227	0.2879	
SDP2&CGP2	ρ_R	0.1095	0.1157	0	0.1073	0.3739	0	0.2914	0.5561	0.3509	
	ρ_Y	0.0238	0.0258	0.1765	0.0148	0.0515	0.2066	0.0060	0.1211	0.3588	
	ρ_G	0.8666	0.8585	0.8235	0.8779	0.5747	0.7934	0.7027	0.3228	0.2904	

ρ_R, ρ_Y and ρ_G represent the probability belonging to red, yellow or green colored group, respectively. For other nodes, though they have not 0–1weights, one dominate component have strength weight more than 0.90.

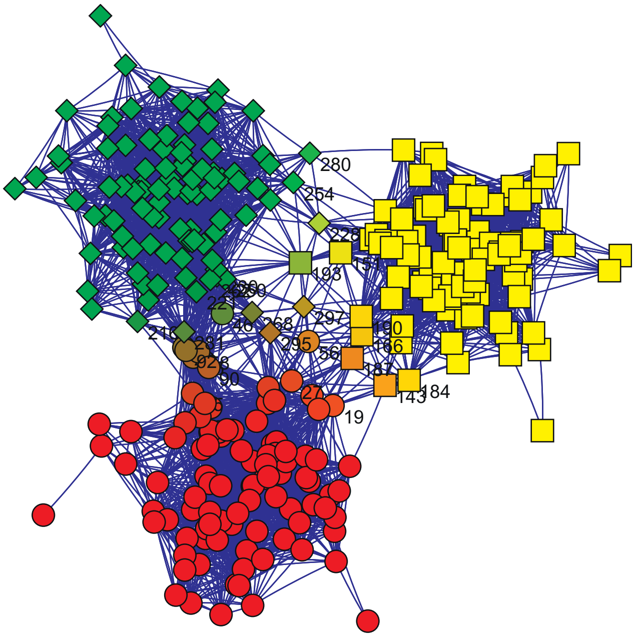

Fig. 5. The visualization of the weights $\{\rho_k(x)\}$. The color vector for each node is given by the weighted average Eq. (31). Both P1 and P2 give nearly the same visualization since the results are approximative. The nodes {15, 19, 25, 27, 29, 46, 56, 58, 90, 92, 143, 151, 166, 184, 187, 190, 193, 216, 221, 228, 230, 250, 254, 262, 268, 280, 281, 295, 297} have observable transition colors, and they play the role of transition in the network. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

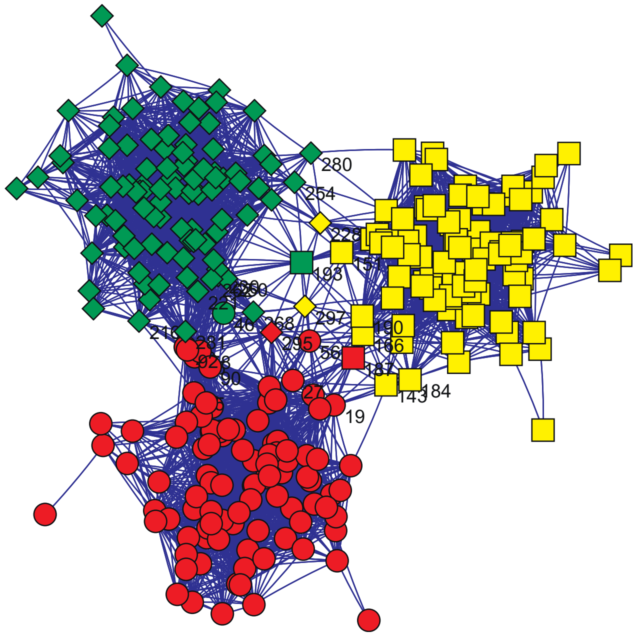

Fig. 6. Partition of the network with thresholding operation according to the node's maximal weight by P2. Both projections P1 and P2 lead to nearly the same partition except for node 297, while red by P1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

to different clusters with probability p_{out}. These values are chosen so that the average node degree d is fixed at $d=16$. In other words, p_{in} and p_{out} are related as

$$31p_{\text{in}}+96p_{\text{out}}=16. \tag{23}$$

We will denote $S_1=\{1:32\}$, $S_2=\{33:64\}$, $S_3=\{65:96\}$, $S_4=\{97:128\}$.

To test on a less diffusive network, we take $z_{\text{out}}=96p_{\text{out}}=5$ and generate the network according to Eq. (23). This network has a fuzzy clustering structure that some nodes should have immediate weights belonging to different clusters. We set the parameters by $K=4$, $TOL=10^{-6}$, and the learning rates $\alpha^{\text{P1}}=5, \alpha^{\text{P2}}=6$, $\beta^{\text{P1}}=0.52$, $\beta^{\text{P2}}=0.51$ in this model computation. $J_{k-\text{means}}=6.0687$ is obtained after initialization [12]. The numerical results are shown in Table 1. Here we compare $\rho_k(x)$ with an interesting quantity, the degree fraction $\rho_k^{\text{E}}(x)$, which is defined as

$$\rho_k^{\text{E}}(x)=\frac{E_k(x)}{d(x)}, \quad k=1,2,3,4, \; x\in S, \tag{24}$$

where $E_k(x)$ is the number of nodes that are connected with x and lie in cluster S_k. Thus we have $\sum_{k=1}^4 E_k(x)=d(x)$. With this definition, $\rho_k^{\text{E}}(x)$ means the fraction of the edges connected with the node x in the k-th cluster. Note that this is not the same as the clustering probability, even though it is an interesting quantity to be compared with. We expect that the degree fraction Eq. (24) is close to our result $\rho_k(x)$ for network though generally this assertion needs to be justified or disconfirmed theoretically. To verify this fact, we define the mean and maximal L^∞- error of ρ:

$$e_\rho^m=\frac{1}{N}\sum_{x\in S}\|\rho(x)-\rho^{\text{E}}(x)\|_\infty, \tag{25a}$$

$$e_\rho^\infty=\max_{x\in S}\|\rho(x)-\rho^{\text{E}}(x)\|_\infty, \tag{25b}$$

for error comparing. Table 1 shows that the deviation between these two is about 0.2. Obviously CG algorithm improves the convergence rate of SD. The projection P1 has the smallest e_ρ^m, while the projection P2 reaches a better minimum which indicates a more accurate result.

Table 4
The iterations and minimized objective function values $J_{\min}$ for the algorithms of the karate club network.

	Iterations	$J_{\min}$
SDP1&CGP1	153&55	4.0394
SDP2&CGP2	173&53	4.0389

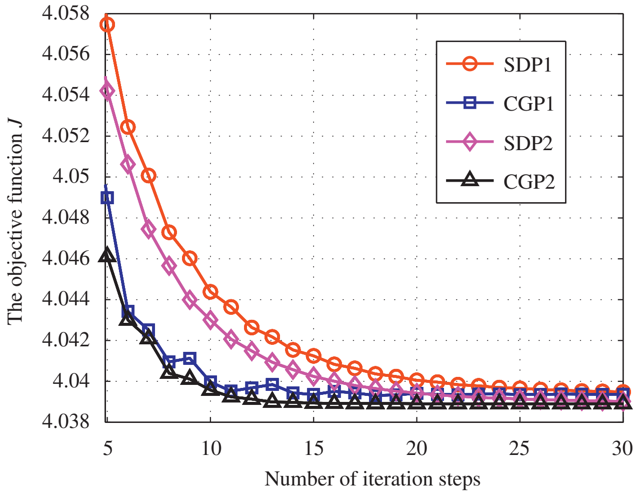

Fig. 7. The convergence history of the objective function J during 5–30 iterations of the karate club network.

In Fig. 1 we plot the probability distribution function (pdf) of ρ_k and ρ_k^{E} $(k=1,2,3,4)$. We observe that the shape of the pdf for ρ_k or ρ_k^{E} is almost the same. Note that all the ρ_k's have a lower peak centered at about 0.85, which corresponds to the nodes in this cluster, and a higher peak centered at about 0.05, which corresponds to the other nodes outside of this cluster. The case for ρ_k^{E} is similar but with the lower peak centered at about 0.7 and the higher peak centered at about 0.1. We note here that the center 0.7 corresponds to the choice of the parameters $z_{\text{out}}/d=5/16 \doteq 0.3$. If we classify the nodes according to the majority rule, i.e. if $m=\text{argmax}_k\,\rho_k(x)$ for a given node x then we set $x\in S_m$, we obtain the 4-cluster partition exactly for this model. This also verifies the accuracy of our algorithms, but fuzzy algorithms give more detailed information for each node.

Table 5
The association probability that each node belongs to different clusters.

	Nodes	1	2	3	4	5	6	7	8	9	10	11	12
SDP1& CGP1	ρ_R	0.0386	0.0782	0.4396	0	0	0	0	0.0037	0.6746	0.7640	0	0
	ρ_Y	0.9614	0.9218	0.5604	1.000	1.0000	1.0000	1.0000	0.9963	0.3254	0.2360	1.0000	1.0000
SDP2& CGP2	ρ_R	0.0522	0.0979	0.4518	0.0183	0	0	0	0.0301	0.6783	0.7662	0	0
	ρ_Y	0.9478	0.9021	0.5482	0.9817	1.0000	1.0000	1.000	0.9699	0.3217	0.2338	1.0000	1.0000
	Nodes	13	14	15	16	17	18	19	20	21	22	23	24
SDP1& CGP1	ρ_R	0	0.2271	1.0000	1.0000	0	0	1.0000	0.3030	1.0000	0	1.0000	1.0000
	ρ_Y	1.0000	0.7729	0	0	1.0000	1.0000	0	0.6970	0	1.0000	0	0
SDP2& CGP2	ρ_R	0	0.2452	1.0000	1.0000	0	0	1.0000	0.3174	1.0000	0	1.0000	1.0000
	ρ_Y	1.0000	0.7548	0	0	1.0000	1.0000	0	0.6826	0	1.0000	0	0
	Nodes	25	26	27	28	29	30	31	32	33	34		
SDP1& CGP1	ρ_R	1.0000	1.0000	1.0000	0.9651	0.8579	1.0000	0.7339	0.9103	1.0000	0.9631		
	ρ_Y	0	0	0	0.0349	0.1421	0	0.2661	0.0897	0	0.0369		
SDP2& CGP2	ρ_R	1.0000	1.0000	1.0000	0.9626	0.8557	1.0000	0.7369	0.9070	1.0000	0.9601		
	ρ_Y	0	0	0	0.0374	0.1443	0	0.2631	0.0930	0	0.0399		

ρ_R and ρ_Y means the probability belonging to red or yellow colored cluster in Fig. 8, respectively.

4.2. Sample network generated from the Gaussian mixture model

To further test the validity of the algorithms, we apply them to a sample network generated from a Gaussian mixture model [25–27]. This model is quite related the concept of random geometric graph proposed by Penrose [29] except that we take Gaussian mixture here compared with uniform distribution in [29].

We generate N sample points $\{\mathbf{x}_i\}$ in two dimensional Euclidean space subject to a K-Gaussian mixture distribution

$$\sum_{k=1}^{K} q_k G(\boldsymbol{\mu}_k, \Sigma_k), \tag{26}$$

where $\{q_k\}$ are mixture proportions satisfying $0<q_k<1$, $\sum_{k=1}^{K} q_k = 1$. $\boldsymbol{\mu}_k$ and Σ_k are the mean positions and covariance matrices for each component respectively. Then we generate the network with a thresholding strategy. That is, if $|\boldsymbol{x}_i - \boldsymbol{x}_j| \le dist$, we set an edge between the i-th and j-th nodes; otherwise they are not connected. With this strategy, the topology of the network is induced by the metric. As a consequence, some properties of the network, say the clustering nature, may be inherited from the case with metric. This is our basic motivation with this model.

We take $N=300$ and $K=3$, then generate the sample points with the means

$$\boldsymbol{\mu}_1 = (2.0, 5.0)^T, \quad \boldsymbol{\mu}_2 = (3.5, 6.5)^T, \quad \boldsymbol{\mu}_3 = (1.5, 7.0)^T, \tag{27}$$

and the covariance matrices

$$\Sigma_1 = \Sigma_2 = \Sigma_3 = \begin{pmatrix} 0.18 & 0 \\ 0 & 0.18 \end{pmatrix}. \tag{28}$$

Here we pick nodes 1:100 in group 1, nodes 101:200 in group 2 and nodes 201:300 in group 3 for simplicity. With this choice, approximately $q_1 = q_2 = q_3 = 100/300$. The thresholding is chosen as $dist = 0.7$ in this example. The sample points are shown in Fig. 2 and the corresponding network is shown in Fig. 3.

To analyze the result obtained by our methods, we first apply the fuzzy c-means algorithm [22,23] to classify our samples in Euclidean space. The main idea of traditional fuzzy c-means algorithm is to minimize the objective function

$$J_{\mathrm{FCM}} = \sum_{k=1}^{K} \sum_{i=1}^{N} \rho_k(\boldsymbol{x}_i)^b \|\boldsymbol{x}_i - \boldsymbol{m}_k\|^2, b \ge 1, \tag{29}$$

where $\boldsymbol{x}_i$ are samples and $\boldsymbol{m}_k$ are centers. We choose $b=2$ in our computations. $\rho_k(\boldsymbol{x}_i)$ denotes the probability of $\boldsymbol{x}_i$ belonging to cluster k, which satisfies the condition

$$\rho_k(\mathbf{x}_i) \ge 0, \quad \sum_{k=1}^{K} \rho_k(\mathbf{x}_i) = 1, \quad i = 1, 2, \ldots, N. \tag{30}$$

We can derive the Euler–Lagrange equations for this objective function with respect to $\mathbf{m}$ and ρ, and iterate until the fixed points are achieved. The readers may be referred to [22,23] for more details. In fact that is the motivation of the current research. We use the traditional FCM algorithm to our sample points given in Fig. 2, the result of weights $\{\rho_k^{\mathrm{FCM}}(x)\}$ is shown in Fig. 4. This is done as follows. Assume that the vectorial representations for different colors in the visualization tool are $C_k, k=1,\ldots,K$. Then the color vector for the node x is given by the weighted average

$$c(x) = \sum_{k=1}^{K} \rho_k(x) C_k, \quad x \in S. \tag{31}$$

Here the vectorial representations for the colors red, yellow and green in the visualization tool are C_R, C_Y and C_G, respectively, and the color vector for the node x is given by $\rho_R(x)C_R + \rho_Y(x)C_Y + \rho_G(x)C_G$. This shows more clearly the transition between different clusters.

As mentioned at the beginning of this subsection, since the network topology is induced by the metric, we expect that the fuzzy clustering in the Euclidean space is close to our result for network though generally this assertion needs to be justified or disconfirmed theoretically. To verify this fact,we denote e_ρ^m and e_ρ^∞ as the mean and maximal L^∞- error of ρ in Eq. (25), with respect to ρ^{FCM} for our methods compared with the traditional FCM algorithm.

Now we implement the algorithms by setting $K=3$, $TOL = 10^{-6}$, $\alpha = 32$, $\beta = 0.22$. Note that $J_{k-\mathrm{means}} = 4.0965$ is obtained after initialization [12]. The numerical results are shown in Table 2 and the intermediate association probability ρ is listed in Table 3.

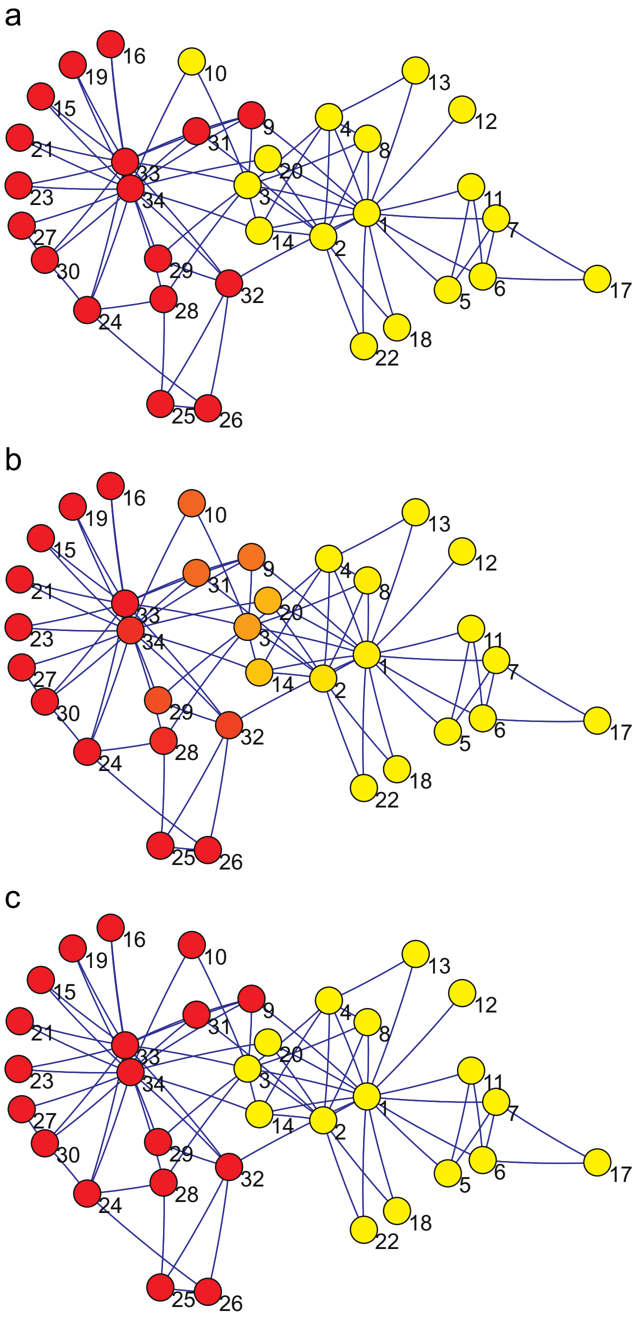

Fig. 8. The visualization of the partitioning results of the karete club network. (a) Hard partition by k-means in [12]; (b) fuzzy partition by CGP2; (c) hard partition with thresholding operation according to the node's maximal weight.

It can be obviously seen that CGP have the higher convergence rate than SDP, and P2 can obtain a smaller value of $J_{\min}$. 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 0.105 while the maximal deviation is about 0.37. The detailed inspection shows that the nodes with large deviations are all located in the transition region. From the above comparisons, the algorithms lead to reasonable results which fits the intuition from the network topology visualization.

Table 6
The iterations and minimized objective function values $J_{\min}$ for the algorithms of the political books network.

	Iterations	$J_{\min}$
SDP1&CGP1	116&47	10.3786
SDP2&CGP2	107&50	10.3782

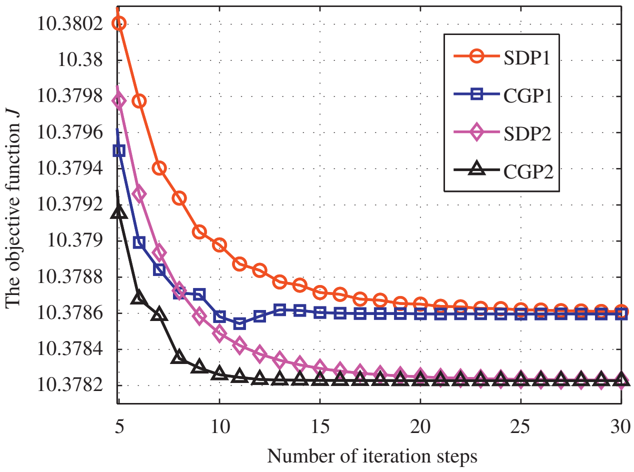

Fig. 9. The convergence history of the objective function J during 5–30 iterations of the political books network.

The weights $\{\rho_k(x)\}$ are shown in Fig. 5 according to the color vector of visualization Eq. (31). This shows more clearly the transition between different clusters. In particular, nodes $\{15, 19, 25, 27, 29, 46, 56, 58, 90, 92, 143, 151, 166, 184, 187, 190, 193, 216, 221, 228, 230, 250, 254, 262, 268, 280, 281, 295, 297\}$ show clearly transitional behavior according to Table 3. If we partition the network by the majority rule, that is, classify the nodes according to their maximal weight, we obtain Fig. 6 and both projections P1 and P2 can lead to nearly the same partition except for node 297, since from Table 3 we see that node 297 has almost equal probabilities of belonging to the red or yellow clusters. The result is reasonable to show that our algorithms go smoothly with several hundreds of nodes.

4.3. Karate club network

This network was constructed by Wayne Zachary after he observed social interactions between members of a karate club at an American university [30]. 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 clusters in networks [6–10,12].

There are 34 nodes in karate club network (see Fig. 8), where each node represents one member in the club. In Zachary's original partition, each node belongs to only one sub-club after splitting. We label it as red or yellow color in the figures to show its attribute in the graph representation. From the viewpoint of the fuzzy 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_R and ρ_Y means the probability of each node belonging to red or yellow colored cluster, respectively.

We set the parameters by $K=2$, $TOL=10^{-6}$, $\alpha^{P1}=1.5$, $\alpha^{P2}=2.0$, $\beta=0.6$. Here $J_{k-\text{means}}=4.1798$ is obtained after initialization [12]. The numerical results are shown in Table 4. It can be demonstrate again that CGP is more efficient, and P2 can reach a smaller value of $J_{\min}$. Fig. 7 shows the convergence history during 5–30 iterations for the methods. It can be obviously seen that CGP, which decreased the objective function faster than SDP, performs more efficiently.

Table 7
The association probability that nodes which have intermediate weights of belonging to different clusters of the political books network.

	Nodes	1	3	5	6	7	8	10	15	20	29
SDP1&CGP1	ρ_R	0.0777	0.2449	0.4011	0.1312	0.0429	0.5122	0.0682	0.0856	0.1198	0.8490
	ρ_Y	0.9223	0.7551	0.5989	0.8688	0.9571	0.4878	0.9318	0.9144	0.8802	0.1510
SDP2& CGP2	ρ_R	0.1096	0.2737	0.4190	0.1599	0.0664	0.5173	0.0774	0.0928	0.1237	0.8491
	ρ_Y	0.8904	0.7263	0.5810	0.8401	0.9336	0.4827	0.9226	0.9072	0.8763	0.1509
	Nodes	30	32	47	49	50	51	52	53	54	58
SDP1& CGP1	ρ_R	0.0479	0.9400	0.1622	0.0975	0.4748	0.2208	0.6896	0.4718	0.1407	0.0556
	ρ_Y	0.9521	0.0600	0.8378	0.9025	0.5252	0.7792	0.3104	0.5282	0.8593	0.9444
SDP2& CGP2	ρ_R	0.0684	0.9300	0.1673	0.1187	0.4786	0.2218	0.6638	0.4602	0.1490	0.0736
	ρ_Y	0.9316	0.0700	0.8327	0.8813	0.5214	0.7782	0.3362	0.5398	0.8510	0.9264
	Nodes	59	65	66	69	70	77	78	86	103	
SDP1& CGP1	ρ_R	0.7111	0.9056	0.9211	0.9467	0.9014	0.9385	0.8644	0.8821	0.8376	
	ρ_Y	0.2889	0.0944	0.0789	0.0533	0.0986	0.0615	0.1356	0.1179	0.1624	
SDP2& CGP2	ρ_R	0.6904	0.8781	0.8874	0.9148	0.8692	0.9280	0.8495	0.8636	0.8326	
	ρ_Y	0.3096	0.1219	0.1126	0.0852	0.1308	0.0720	0.1505	0.1364	0.1674	

ρ_R and ρ_Y represent the probability belonging to red or yellow colored group in Fig. 10, respectively. The other nodes have one dominate component of weights more that 0.95.

The final association probabilities are presented in Table 5, where ρ_R and ρ_Y are the probability of belonging to the red or yellow colored group shown in Fig. 8, respectively. Comparing ρ_R or ρ_Y between P1 and P2, we find that almost all the errors are less than 10^{-2}, but the association probability ρ is quite different from the 0–1 distributions obtained in the *k*-means algorithm. Now let us compare the association probability ρ_R and ρ_Y obtained by our methods with the original partition result obtained by Zachary. In [30], Zachary gave the partition $S_Y=\{1:8,11:14,17,18,20,22\}$ and $S_R=\{9,10,15,16,19,21,23:34\}$. If we classify the nodes according to the majority rule, i.e., if $\rho_R(x)>\rho_Y(x)$ then we set $x\in S_R$, otherwise we set $x\in S_Y$, we obtain the same partition as Zachary's (see Fig. 8(c)). We note that this hard partition deduced by fuzzy partition is more reasonable than the result of *k*-means (see Fig. 8(a)), since node 10 is classified correctly this time.

However, we have more detailed information in fact. From Table 5, we find $\rho_Y=1$ for nodes $\{5:7,11:13,17:18,22\}$, which lie at the boundary of the yellow colored group; and $\rho_R=1$ for nodes $\{15:16,19,21,23:27,30,33\}$, which mostly lie at the boundary of the red colored group. The others belong to the red and yellow colored groups with nonzero probability, especially the nodes $\{3,9,10,14,20,31\}$ have more diffusive probability and they play the role of transition nodes between the red and yellow colored groups. We can visualize the data ρ more transparently with the color vector Eq. (31) for each node. We can conclude from Fig. 8(b) that how much probability each of the 34 members stands by both parts with. Members $\{5:7,11:13,17:18,22\}$ and $\{15:16,19,21,23:27,30,33\}$ have an obvious attitude on following their leader, i.e. the club administrator or the main teacher. Others such as $\{3,9,10,14,20,31\}$ hold neutralism that they can support either leader according their weights.

4.4. Political books network

The last example we consider the network of books on politics, which is compiled by Krebs (unpublished, but can be found in www.orgnet.com) [10]. In this network the nodes represent 105 recent books on American politics bought from the on-line bookseller Amazon.com, and the edges join pairs of books that are frequently purchased by the same buyer, as indicated by the feature that customers who bought this book also bought these other books. As shown in Fig. 10, nodes have been given whether they are conservative (box) or liberal (diamond), except for a small number of books which are neutral(ellipse). These alignments were assigned separately by Newman [10] based on a reading of the descriptions and reviews of the books posted on Amazon.

We set $K=2$, $TOL=10^{-6}$, $\alpha=10$, $\beta=0.5$ for this model. Here $J_{k-\text{means}}=10.4532$ is obtained after initialization [12]. The numerical results are shown in Table 6. We again point out that CGP is more efficient, and P2 can reach a smaller value of $J_{\min}$. The optimal convergence rate is also reached when CGP2 is operated. Fig. 9 gives the convergence history during 5–30 iterations for the methods. The fact that CGP performs more efficiently than SDP is shown again here.

The association probabilities ρ_R and ρ_Y that nodes have intermediate weights of belonging to different clusters obtained by our methods are shown in Table 7, which can give us more detailed information. We find that the nodes mentioned in this table have diffusive probability, especially nodes $\{3,5,8,50,51,52,53,59\}$ have more diffusive probability with the dominate weights less than 0.8. All of them play the role of transition between the red and yellow colored clusters. In Fig. 10(b), we can see that one of these clusters consists almost entirely of liberal books and one almost entirely of conservative books. Most of ellipse nodes are colored with orange in some grade. It means that the neutral books share commonalities between conservative books and liberal ones. If we classify the nodes according to the majority rule, we obtain the partition shown in Fig. 10(c). The result is nearly the same as the partition of *k*-means shown in Fig. 10(a) except nodes $\{8,50,53\}$, since they have almost equal probabilities of belonging to either cluster according to Table 7. Thus these books appear to form clusters of copurchasing that align closely with political views that

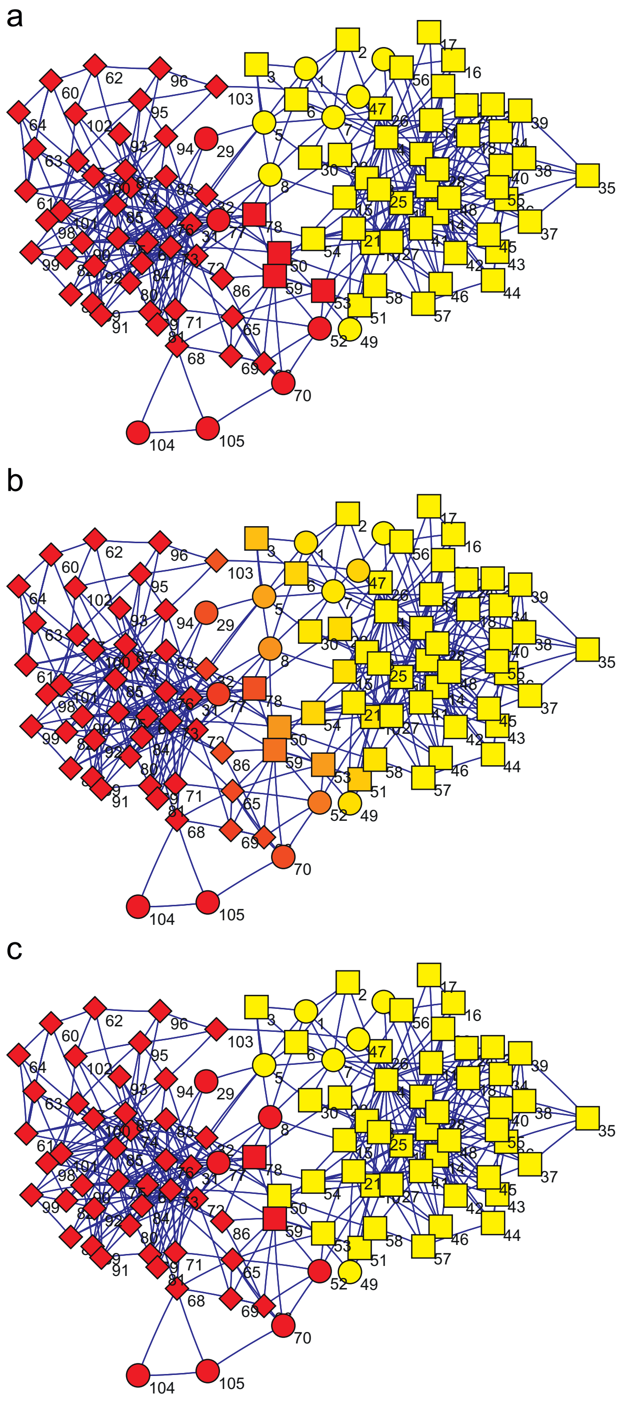

Fig. 10. The visualization of the partitioning results of the political books network. (a) Hard partition by *k*-means in [12]; (b) fuzzy partition by CGP2; (c) hard partition with thresholding operation according to the node's maximal weight.

encourage us to believe our algorithms are capable of extracting meaningful results from raw network data. The algorithm in [10] found four clusters of nodes. The same as our results, one of them consists almost liberal books and one almost of conservative ones. Most of the centrist books fall in the two remaining clusters, and this correspond to the orange nodes in Fig. 10(b) which play the role of transition.

5. Conclusions

We address the fuzzy clustering problem for networks with gradient methods in this paper. This can also be considered as a generalization of fuzzy *c*-means in statistics for the networks. The hard clustering concept, a node belongs to only one cluster, is extended to the fuzzy clustering concept that each node may belong to different clusters with nonzero probability. It is meaningful in many diffusive cases that nodes at the boundary among clusters share commonalities with more than one cluster and play a role of transition, and such probabilities can give people more detailed information. We successfully constructed the steepest descent method with projection (SDP) and the reduced conjugate gradient method with projection (CGP). They are derived to search for the local minimum of the objective function in Eq. (14) under the fuzzy clustering framework, which is extended from a deterministic framework for network partition based on the optimal prediction of a random walker Markovian dynamics [12]. The simulation experiments have shown that the algorithms can efficiently determine the fuzzy partition matrix. Partitioning the network with thresholding operation according to the node's maximal weight can give a more reasonable clustering result than the previous *k*-means algorithm [12]. Numerical results show that our algorithms with two different projections produce similar results, while the CGP2 algorithm has better efficiency and accuracy. Moreover, the algorithms succeed in two real-world learning tasks, including of the karate club network and the political books network.

Acknowledgements

This work is supported by the Natural Science Foundation of China under Grant 10871010 and the National Basic Research Program of China under Grant 2005CB321704.

Appendix A. Proof of Lemma 1

Firstly we take the variation of J in Eq. (14) with respect to $\hat{p}_{kl}$

$$\begin{aligned}\frac{\partial J}{\partial \hat{p}_{kl}} &= 2\sum_{x,y\in S}\mu(x)\mu(y)\left(\sum_{m,n=1}^{K}\rho_m(x)\rho_n(y)\frac{\hat{p}_{mn}}{\hat{\mu}_n}-\frac{p(x,y)}{\mu(y)}\right)\\ &\quad\cdot\sum_{s,t=1}^{K}\rho_s(x)\rho_t(y)\frac{1}{\hat{\mu}_t}\delta_{sk}\delta_{tl} = 2\left(\frac{1}{\hat{\mu}_l}\sum_{x,y\in S}\sum_{m,n=1}^{K}\mu(x)\mu(y)\rho_k(x)\rho_l(y)\right.\\ &\quad\left.\rho_m(x)\rho_n(y)\frac{\hat{p}_{mn}}{\hat{\mu}_n}-\frac{1}{\hat{\mu}_l}\sum_{x,y\in S}\mu(x)p(x,y)\rho_k(x)\rho_l(y)\right)\\ &= 2\left(\frac{1}{\hat{\mu}_l}\sum_{m,n=1}^{K}\hat{\hat{\mu}}_{km}\frac{\hat{p}_{mn}}{\hat{\mu}_n}\hat{\hat{\mu}}_{nl}-\frac{\hat{\mu}_k}{\hat{\mu}_l}\hat{p}^*_{kl}\right).\end{aligned}\qquad(32)$$

Representing the above result with matrix form gives Eq. (17a).

Now we take the variation of J with respect to $\rho_r(z)$, which gives

$$\begin{aligned}\frac{\partial J}{\partial \rho_r(z)} &= 2\sum_{x,y\in S}\mu(x)\mu(y)\left(\sum_{m,n=1}^{K}\rho_m(x)\rho_n(y)\frac{\hat{p}_{mn}}{\hat{\mu}_n}-\frac{p(x,y)}{\mu(y)}\right)\\ &\quad\cdot\sum_{k,l=1}^{K}\left[\delta_{kr}\delta(x,z)\rho_l(y)\frac{\hat{p}_{kl}}{\hat{\mu}_l}+\delta_{lr}\delta(y,z)\rho_k(x)\frac{\hat{p}_{kl}}{\hat{\mu}_l}\right.\\ &\quad\left.-\rho_k(x)\rho_l(y)\frac{\hat{p}_{kl}}{\hat{\mu}_l^2}\sum_{w\in S}\delta_{lr}\delta(w,z)\mu(w)\right].\end{aligned}\qquad(33)$$

With the detail balance condition in Eq. (3) and the definition of $\hat{p}^*$ in Eq. (15), we actually have

$$\begin{aligned}\frac{\partial J}{\partial \rho_r(z)} &= 2\left[\sum_{y\in S}\mu(z)\mu(y)\left(\sum_{m,n=1}^{K}\rho_m(z)\rho_n(y)\frac{\hat{p}_{mn}}{\hat{\mu}_n}-\frac{p(z,y)}{\mu(y)}\right).\right.\\ &\quad\cdot\sum_{k=1}^{K}\rho_l(y)\frac{\hat{p}_{rl}}{\hat{\mu}_l}+\sum_{x\in S}\mu(z)\mu(x)\left(\sum_{m,n=1}^{K}\rho_m(x)\rho_n(z)\frac{\hat{p}_{mn}}{\hat{\mu}_n}-\frac{p(x,z)}{\mu(z)}\right)\\ &\quad\cdot\sum_{k=1}^{K}\rho_k(x)\frac{\hat{p}_{kr}}{\hat{\mu}_r}-\sum_{x,y\in S}\mu(x)\mu(y)\left(\sum_{m,n=1}^{K}\rho_m(x)\rho_n(y)\frac{\hat{p}_{mn}}{\hat{\mu}_n}-\frac{p(x,y)}{\mu(y)}\right)\\ &\quad\left.\cdot\sum_{k=1}^{K}\mu(z)\rho_k(x)\rho_r(y)\frac{\hat{p}_{kr}}{\hat{\mu}_r^2}\right] = 2\left[\sum_{l,m,n=1}^{K}\hat{\hat{\mu}}_{ln}\rho_l(y)\frac{\hat{p}_{mn}}{\mu_n}\frac{\hat{p}_{rl}}{\hat{\mu}_l}\right.\\ &\quad-\sum_{y\in S}\sum_{l=1}^{K}p(z,y)\rho_l(y)\frac{\hat{p}_{rl}}{\hat{\mu}_l}.+\sum_{k,m,n=1}^{K}\hat{\hat{\mu}}_{km}\rho_k(x)\frac{\hat{p}_{mn}}{\hat{\mu}_n}\frac{\hat{p}_{kr}}{\hat{\mu}_r}\\ &\quad-\sum_{x\in S}\sum_{k=1}^{K}p(z,x)\rho_k(x)\frac{\hat{p}_{kr}}{\hat{\mu}_r}-\sum_{k,m,n=1}^{K}\hat{\hat{\mu}}_{mk}\hat{\hat{\mu}}_{rn}\frac{\hat{p}_{mn}}{\hat{\mu}_n}\frac{\hat{p}_{kr}}{\hat{\mu}_r^2}\\ &\quad\left.+\sum_{k=1}^{K}\mu_k\hat{p}^*_{kr}\frac{\hat{p}_{kr}}{\hat{\mu}_r^2}\right]\mu(z).\end{aligned}\qquad(34)$$

After suitable manipulations we obtain Eq. (17b) finally.

References

[1] R. Albert, A.-L. Barabási, Statistical mechanics of complex networks, Rev. Mod. Phys. 74 (1) (2002) 47–97.

[2] M. Newman, A.-L. Barabási, D.J. Watts, The Structure and Dynamics of Networks, Princeton University Press, Princeton, NJ, 2005.

[3] National Research Council, Network Science, National Academy of Sciences, Washington DC, 2005.

[4] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intel 22 (2000) 888–905.

[5] M. Meilă, J. Shi, A random walks view of spectral segmentation, in: Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics, 2001, pp. 92–97.

[6] M. Girvan, M. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99 (12) (2002) 7821–7826.

[7] M. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (2004) 066133.

[8] M. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 26113.

[9] M. Newman, Detecting community structure in networks, Eur. Phys. J. B 38 (2) (2004) 321–330.

[10] M. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. USA 103 (23) (2006) 8577–8582.

[11] S. Lafon, A.B. Lee, Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization, IEEE Trans. Pattern Anal. Mach. Intel (2006) 1393–1403.

[12] W.E.T. Li, E. Vanden-Eijnden, Optimal partition and effective dynamics of complex networks, Proc. Natl. Acad. Sci. USA 105 (2008) 7907–7912.

[13] T. Li, J. Liu, W. E, Probabilistic framework for network partition, Phys. Rev. E 80 (2009) 026106.

[14] J.M. Hofman, C.H. Wiggins, A Bayesian approach to network modularity, Phys. Rev. Lett. 100 (2008) 258701.

[15] W. Schilders, H. van der Vorst, J. Rommes, Model Order Reduction: Theory, Research Aspects and Applications, Springer, Berlin, Heidelberg, 2008.

[16] L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, J. Stat. Mech. 9 (2005) P09008.

[17] A. Chorin, A. Kast, R. Kupferman, Unresolved computation and optimal predictions, Commun. Pure Appl. Math. 52 (10) (1999) 1231–1254.

[18] A. Chorin, Conditional expectations and renormalization, Multi. Model. Simul. 1 (2003) 105–118.

[19] L. Lovasz, Random walks on graphs: a survey, Combinatorics, Paul Erdos is Eighty, vol. 2, 1993, pp. 1–46.

[20] P.A. Devijver, J. Kittter, Pattern Recognition: A Statistical Approach, Prentice-Hall, Englewood Cliffs, NJ, 1982.

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

[22] J. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, Cybern. Syst. 3 (3) (1973) 32–57.

[23] J. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.

[24] F. Chung, Spectral Graph Theory, American Mathematical Society, Rhode Island, 1997.

[25] J. Ma, L. Wang, BYY harmony learning on finite mixture: adaptive gradient implementation and a floating RPCL mechanism, Neural Process. Lett. 24 (2006) 19–40.

[26] J. Ma, T. Wang, L. Xu, A gradient BYY Harmony learning rule on Gaussian mixture with automated model selection, Neurocomputing 56 (2004) 481–487.

[27] J. Ma, B. Gao, Y. Wang, Q. Cheng, Conjugate and natural gradient rules for BYY harmony learning on Gaussian mixture with automated model selection, Int. J. Pattern Recognition Artif. Intell. 19 (2005) 701–713.

[28] N. Qian, On the momentum term in gradient descent learning algorithms, Neural Networks 12 (1) (1999) 145–151.

[29] M. Penrose, Random Geometric Graphs, Oxford University Press, Oxford, 2003.

[30] W. Zachary, An information flow model for conflict and fission in small groups, J. Anthrop. Res. 33 (4) (1977) 452–473.

About the Author—JIAN LIU received the Bachelor of Science in information and computational science at the School of Mathematics, Jilin University in 2006. Currently, she is a Ph.D. candidate at the School of Mathematical Sciences, Peking University. Her interests include pattern recognition, model reduction of complex networks, learning theory and algorithm.