

# Uncovering and Testing the Fuzzy Clusters Based on Lumped Markov Chain in Complex Network

Fan Jing<sup>1\*</sup>, Xie Jianbin<sup>2</sup>, Wang Jinlong<sup>1</sup>, Qu Jinshuai<sup>1</sup>

**1** University Key Laboratory of Wireless Sensor Networks in Yunnan Province, Yunnan University of Nationalities, Kunming, P.R. China, **2** School of Urban Construction and Management, Yunnan University, Kunming, P.R. China

## Abstract

Identifying clusters, namely groups of nodes with comparatively strong internal connectivity, is a fundamental task for deeply understanding the structure and function of a network. By means of a lumped Markov chain model of a random walker, we propose two novel ways of inferring the lumped Markov transition matrix. Furthermore, some useful results are proposed based on the analysis of the properties of the lumped Markov process. To find the best partition of complex networks, a novel framework including two algorithms for network partition based on the optimal lumped 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 and naturally supports the fuzzy partition. Moreover, they are successfully applied to real-world network, including the social interactions between members of a karate club.

**Citation:** Jing F, Jianbin X, Jinlong W, Jinshuai Q (2013) Uncovering and Testing the Fuzzy Clusters Based on Lumped Markov Chain in Complex Network. PLoS ONE 8(12): e82964. doi:10.1371/journal.pone.0082964

**Editor:** Peter Csermely, Semmelweis University, Hungary

**Received:** August 24, 2013; **Accepted:** October 29, 2013; **Published:** December 31, 2013

**Copyright:** © 2013 Jing et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Funding:** This work was supported by the National Natural Science Foundation of China (grant No. 60963026, No. 61163061), and the Application Basic Research Fund of Yunnan Province (grant No. 2011FZ174). The National Natural Science Foundation of China: <http://www.nsf.gov.cn/>. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing Interests:** The authors have declared that no competing interests exist.

\* E-mail: fanjing9476@163.com

## Introduction

The theory of network science has significantly improved our understanding of complex systems. In the last fifteen years, an explosive growth of interests and activities on the structure and dynamics of complex networks is appearing. One of the most favorable but challenging tasks in network science is cluster analysis, which is aimed at revealing possible partitions of a network into subsets of nodes (clusters, or communities) [1]. Markov chains are frequently used as analytic models in the quantitative evaluations of stochastic systems. Examples of their use may be found in diverse areas such as computer, biological, physical and social sciences as well as in business, economics and engineering [2–4]. The fundamental property characterizing the model, referred to as the Markov property, is that given the present, past and future transitions of the system are independent of each other [5,6]. The information that is often sought from this model is either the transient or stationary probability of the system being in a given state. When the number of states is small, it is relatively easy to obtain the transient and stationary solutions allowing the prediction of the system behavior. However, as models become more complex the process of obtaining these solutions becomes much more difficult. There is also a wide class of situations, where the modeler does not need information about each state of the system but about classes of states only. This leads to the consideration of a new process, to be called the aggregated or lumped, whose states are the state classes of the original Markov chain. The new stochastic process need not to be Markovian. In order to be able to utilize all the power of the Markov chain

theory, it is important to be able to claim that for a given initial distribution the aggregated process has the Markov property.

In a previous paper [7], a  $k$ -means approach is proposed to partition the networks based on optimal prediction theory proposed by Chorin and coworkers [8,9]. The basic idea is to associate the network with the random walker Markovian dynamics [10], 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 [11,12] in clustering analysis. This approach is motivated by the diffusion maps [13] and MNCut algorithms in imaging science [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 [5–7,13–20]. And in a broader aspect, it is also closely related to the model reduction theory of differential equations [21,22]. The concept of lumpability on hard partitions is a useful tool to analysis the dynamic of the network based on the a coarse grain view. Lumped Markov chain model of the random walker (i.e., a reduced-order Markov chain in which the clusters of the original network become nodes) which is easily derived from the original (highorder) Markov chain model. This notion of proximity of ‘nodes’ in the lumped Markov chain reflects the intrinsic geometry of the meta-node set in terms of connectivity of the cluster in a diffusion process.

In the current paper, we first analyze the property of lumped markov chain and propose two novel ways of inferring the lumped markov transition matrix. Some useful results are proposed based on the analysis of the properties of the lumped markov process. Furthermore, we construct 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, as well as real-world network, the karate club 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.

### Materials and Methods

We will start with a brief review of markov random walks on complex network [8,9]. 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

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

where  $d(x)$  is the degree of the node  $x$  [10–13,23]. This Markov chain has stationary distribution

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

and it satisfies the detailed balance condition

$$\mu(x)p(x,y) = \mu(y)p(y,x). \quad (3)$$

We take a partition of  $S$  as  $S = \cup_{k=1}^K S_k$  with  $S_k \cap S_l = \emptyset$  if  $k \neq l$ . Our aim is to aggregate the nodes in each cluster in order to give the exact expression of lumped markov chain. To do so, first, we regard each set  $S_k$  in the state space  $\mathbb{S} = \{S_1, \dots, S_K\}$  as corresponding to the nodes of a  $N$ -nodes network  $\hat{G}(\mathbb{S}, \mathbb{E})$ , where  $\mathbb{E} = \{\hat{e}(S_k, S_l)\}_{S_k, S_l \in \mathbb{S}}$ , and the weight  $\hat{e}(S_k, S_l)$  on the edge that connects  $S_k$  and  $S_l$  is defined as

$$\hat{e}(S_k, S_l) = \sum_{x \in S_k, y \in S_l} \mu(x)p(x,y). \quad (4)$$

where the sum involves all the transition probabilities between  $x \in S_k, y \in S_l$ . From the detailed balance condition (4), it can be verified that  $\hat{e}(S_k, S_l) = \hat{e}(S_l, S_k)$ . By setting  $\hat{\mu}_k = \sum_{z \in S_k} \mu(z), k = 1, \dots, K$ , one can define the lumped Markov chain on graph  $\hat{G}(\mathbb{S}, \mathbb{E})$  with stationary distribution  $\hat{\mu}$  and lumped transition probabilities

$$\hat{p}_{kl} = \frac{\hat{e}(S_k, S_l)}{\sum_{m=1}^K \hat{e}(S_k, S_m)} = \frac{1}{\hat{\mu}_k} \sum_{x \in S_k, y \in S_l} \mu(x)p(x,y). \quad (5)$$

It can be easily shown that  $\hat{p}$  is a stochastic matrix on the state space  $\mathbb{S}$  and satisfies a detailed balance condition with respect to  $\hat{\mu}$ , i.e.

$$\hat{\mu}_k \hat{p}_{kl} = \hat{\mu}_l \hat{p}_{lk}. \quad (6)$$

This construction of a lumped markov chain is shown in Figure 1.

To give a more clear probability expression, 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

$$\rho_k(x) \geq 0 \quad \text{and} \quad \sum_{k=1}^K \rho_k(x) = 1 \quad (7)$$

for all  $x \in S$ .

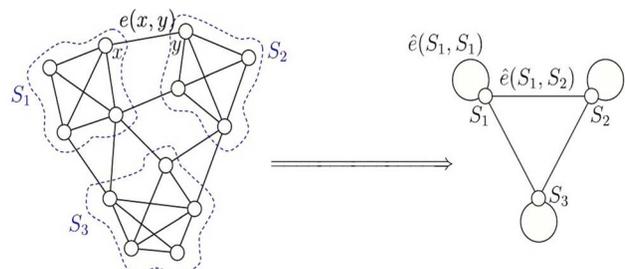
Thus, the lumped markov transition matrix can be rewritten based on  $\mu$

$$\begin{aligned} \hat{p}_{kl} &= \frac{1}{\hat{\mu}_k} \sum_{x \in S_k, y \in S_l} \mu(x) \rho_k(x) p(x,y) \rho_l(y) \\ &= \sum_{x \in S_k, y \in S_l} \mu_k(x) p(x,y) \rho_l(y) \end{aligned} \quad (8)$$

Another approach of deriving the lumped markov transition matrix is based on lifted transition matrix. Any lumped stochastic matrix  $\hat{p}$  can be naturally lifted to the space of stochastic matrices on the original state space  $S$  via

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

where



**Figure 1. Lumped markov process of a network.** For a given partition  $S = S_1 \cup S_2 \cup S_3$  on a network  $G(S,E)$ , we define a lumped network  $\hat{G}(\mathbb{S}, \mathbb{E})$  by aggregating all nodes belonging to a subset  $S_k$  into a meta-node. The new weights  $\hat{e}(S_k, S_l)$  are computed via weight averaging the transition probabilities between nodes  $x \in S_k$  and  $y \in S_l, k, l = 1-3$ , and the lumped Markov chain with transition probabilities  $\hat{p}(S_k, S_l)$  can also be obtained. doi:10.1371/journal.pone.0082964.g001

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

The basic idea in [24,25] is to introduce a metric, also called the Hilbert-Schmidt norm, in the space of stochastic matrices. The optimal partition and the corresponding reduced Markov chain  $\tilde{p}$  is found by minimizing

$$\min_{\rho_k(x), \hat{p}_{kl}} J = \|p - \tilde{p}\|_{\mu}^2 \quad (11)$$

where

$$\begin{aligned} J &= \sum_{x,y \in S} \frac{\mu(x)}{\mu(y)} |p(x,y) - \tilde{p}(x,y)|^2 \\ &= \sum_{x,y \in S} \mu(x)\mu(y) \\ &\quad \cdot \left( \sum_{m,n=1}^K \rho_m(x)\rho_n(y) \frac{\hat{p}_{mn}}{\hat{\mu}_n} - \frac{p(x,y)}{\mu(y)} \right)^2, \end{aligned} \quad (12)$$

subject to the constraint Eq.(7).

To minimize the objective function  $J$  in Eq.(12), we define

$$\begin{aligned} \hat{p}_{kl}^* &= \sum_{x,y \in S} \mu_k(x)p(x,y)\rho_l(y) \\ &= \frac{1}{\hat{\mu}_k} \sum_{x,y \in S} \mu(x)\rho_k(x)p(x,y)\rho_l(y) \end{aligned} \quad (13)$$

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

$$\hat{\mu}_k \hat{p}_{kl}^* = \hat{\mu}_l \hat{p}_{lk}^*. \quad (14)$$

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

The expressions of  $\frac{\partial J}{\partial \hat{p}}$  and  $\frac{\partial J}{\partial \rho}$  according to Eq.(12) are given by

$$\frac{\partial J}{\partial \hat{p}} = 2(\hat{\mu} \hat{p} I_{\hat{\mu}}^{-1} \hat{\mu} I_{\hat{\mu}}^{-1} - I_{\hat{\mu}} \hat{p}^* I_{\hat{\mu}}^{-1}), \quad (15a)$$

$$\begin{aligned} \frac{\partial J}{\partial \rho} &= 2[(\hat{p} I_{\hat{\mu}}^{-1} \hat{\mu} I_{\hat{\mu}}^{-1} \hat{p}^T + I_{\hat{\mu}}^{-1} \hat{p}^T \hat{\mu} \hat{p} I_{\hat{\mu}}^{-1}) \cdot \rho I_{\hat{\mu}} - \\ &\quad (\hat{p} I_{\hat{\mu}}^{-1} + I_{\hat{\mu}}^{-1} \hat{p}^T) \cdot \rho \rho^T I_{\hat{\mu}} \\ &\quad - (I_{\hat{\mu}}^{-2} \hat{p}^T) * (\hat{\mu} I_{\hat{\mu}}^{-1} \hat{p}^T \hat{\mu}) \cdot \rho \times \mu^T + \\ &\quad ((I_{\hat{\mu}}^{-2} \hat{p}^T) * (\hat{p}^*)^T) \cdot \hat{\mu} \mu^T], \end{aligned} \quad (15b)$$

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

$$\hat{\mu}_{kl} = \sum_{z \in S} \mu(z)\rho_k(z)\rho_l(z) = (\rho \cdot I_{\hat{\mu}} \cdot \rho^T)_{kl}. \quad (17)$$

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

$$I_{\mu}(x,y) = \mu(x)\delta(x,y), \quad x,y \in S, \quad (17a)$$

$$(I_{\hat{\mu}})_{kl} = \hat{\mu}_k \delta_{kl}, \quad k,l = 1, \dots, K, \quad (17b)$$

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

The proof of Lemma 1 can be found in Appendix\_S1.pdf. Eq.(15a) and Eq.(15b) give the partial derivatives of the objective function in Eq.(12), which are the critical points of constructing gradient methods.

Based upon this formulation, we can find the optimal  $\hat{p}_{kl}$  for any fixed partition  $\{S_1, \dots, S_N\}$ . From the optimality condition

$$\frac{\partial J}{\partial \hat{p}} = 2(\hat{\mu} \hat{p} I_{\hat{\mu}}^{-1} \hat{\mu} I_{\hat{\mu}}^{-1} - I_{\hat{\mu}} \hat{p}^* I_{\hat{\mu}}^{-1}) = 0, \quad (18)$$

we can obtain Eq.(8). It indicates that when the partition  $\{S_1, \dots, S_K\}$  is known, the minimizer of Eq.(11) is unique, which can be given by Eq.(8), and the corresponding  $\tilde{p}$  is the stochastic matrix in the class Eq.(9) which provides the best rank- $K$  approximation of the original one under the metric Eq.(11).

### SDP

In this section, based on the lumped markov transition matrix  $\hat{p}$  and corresponding results derived before, we devote to find ways to uncover partitions. An obvious choice to solve the constrained optimization Eq.(11) is the steepest descent method [12]. However, the components of  $\hat{p}$  and  $\rho$  may become negative and non-normalized during the descent procedure for our problems, which makes the probabilistic interpretation useless. To ensure the nonnegativity and normalization conditions for  $\hat{p}$  and  $\rho$ , we add a projection step after each renewal. This leads to the following Steepest Descent method with Projections(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 [7],  $n=0$ .
- (2) Perform the following iteration until  $\|\rho^{(n+1)} - \rho^{(n)}\| \leq TOL$ :

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

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

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.

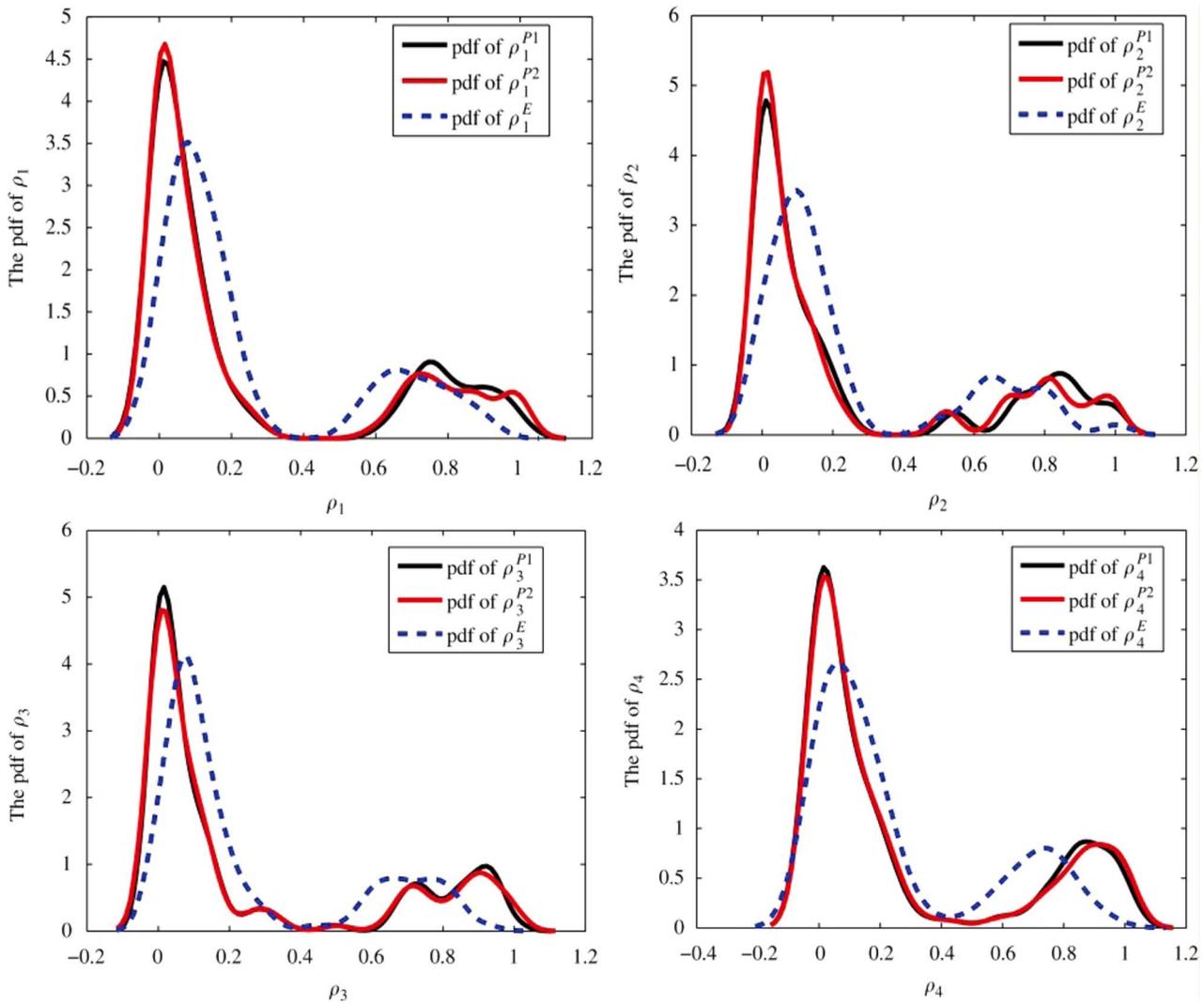
Two choices of the projection operator  $\mathcal{P}$  are used in our computations, but the final results seem to be insensitive to them. Now suppose  $\mathbf{v} = (v_1, v_2, \dots, v_N) \in \mathbb{R}^N$ , and  $v_i < 0$  when  $i \in \Lambda$ , we make projection as any of the following two choices

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



**Figure 2. The pdf of  $\rho_k$  and  $\rho_k^E$  ( $k=1,2,3,4$ ) for the given ad hoc network with 128 nodes and  $z_{out}=5$ .** The solid lines and dashed lines represent the pdf of  $\rho_k$  and  $\rho_k^E$  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.  
doi:10.1371/journal.pone.0082964.g002

$\mathcal{P}v_{i_0}=0$  and make projection again to a dimension reduced hyperplane  $\sum_{i \neq i_0} v_i = 1$ . 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.(11), while to solve it exactly may involve much more complexity. Note that with this SDP method, we can not guarantee that  $\hat{p}$  and  $\rho$  are the exact minimizer of the original problem Eq.(11) with non-negative 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  $\alpha$  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  $0 \leq \alpha(n) \leq 1$ , and

$$\lim_{n \rightarrow \infty} \alpha(n) = 0, \quad \sum_{n=1}^{\infty} \alpha(n) = \infty. \quad (20)$$

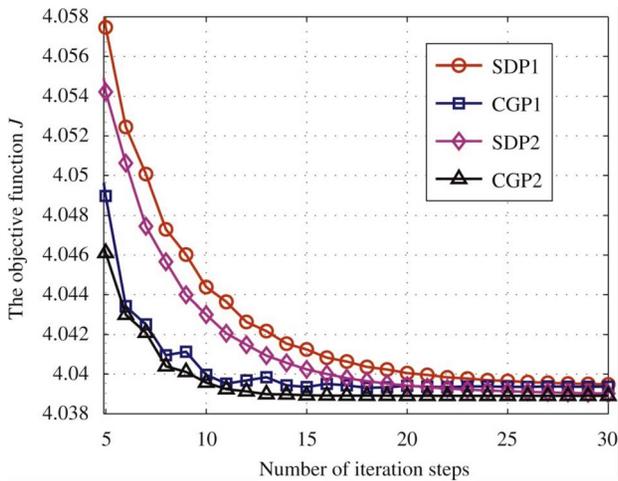
The typical example of such case is  $\alpha(n) = \alpha_0/n$ , where  $\alpha_0$  is a positive constant [26]. Another choice is to fix the learning rate  $\alpha$  as a positive constant [24,25], which we utilize here, since the initial partition is good enough that the objective function Eq.(12) descends much more slowly when the learning rate becomes smaller, while larger values of  $\alpha$  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

**Table 1.  $J_{min}$ ,  $e_{\rho}^m$ ,  $e_{\rho}^{\infty}$ .**

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

The iterations, the value of the objective function  $J_{min}$  and the mean and maximum  $L^{\infty}$ -error of  $\rho$  defined in Eq.(3) for algorithms.  
doi:10.1371/journal.pone.0082964.t001



**Figure 3. The convergence history of the objective function  $J$  during 5–30 iterations of the karate club network.**  
doi:10.1371/journal.pone.0082964.g003

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.(15a). It is easy to find that the computation of  $\hat{\mu}$  costs  $O(KN)$ , and the computation of  $\hat{\mu}$  costs  $O(K^2N)$ . Note that the stochastic matrix  $p$  is sparse with  $M$  entries, so the

**Table 2.  $J_{\min}$  of the karate club.**

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

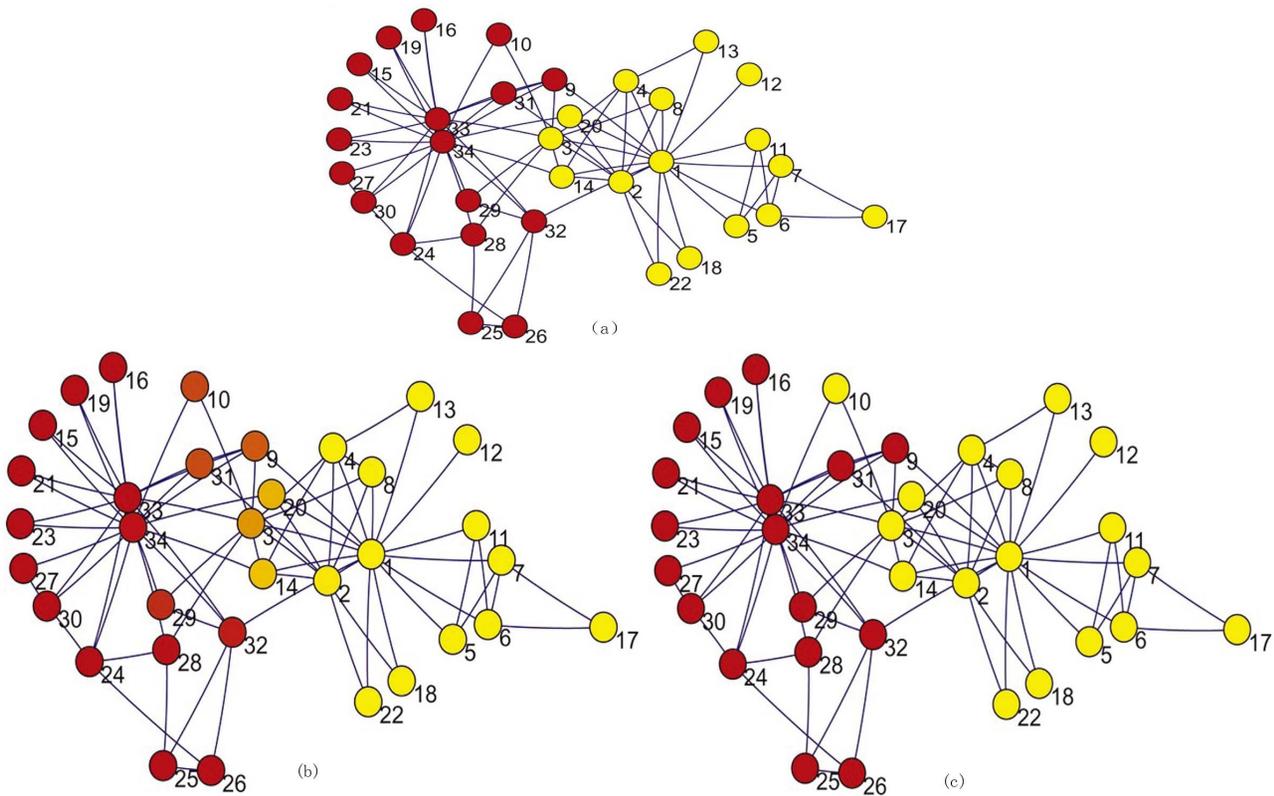
The iterations and minimized objective function values  $J_{\min}$  for the algorithms of the karate club network.  
doi:10.1371/journal.pone.0082964.t002

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  $\rho$  is also  $O(K^2(M+N))$  according to Eq.(15b), since  $\hat{p}^*$  is involved in the equations.

### CGP

Another choice is to minimize the objective function using a simplified formulation of traditional conjugate gradient method, which is frequently used in machine learning [27]. It can be also regard as the above steepest descent method with a non-zero momentum term, which leads to the following reduced Conjugate Gradient method with Projections(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 [13],  $n=0$ .



**Figure 4. The visualization of the partitioning results of the karate club network.** (a)Hard partition with thresholding operation according to the node's maximal weight. (b)Hard partition by  $k$ -means in [13]; (c)Fuzzy partition by CGP2.  
doi:10.1371/journal.pone.0082964.g004

**Table 3.** The association probability.

	Nodes	1	2	3	4	5	6	7	8	9	10	11	12
SDP1&CGP1	$\rho_R$	0.039	0.079	0.44	0	0	0	0	0.004	0.675	0.764	0	0
	$\rho_Y$	0.96	0.92	0.56	1	1	1	1	0.996	0.3254	0.236	1	1
SDP2&CGP2	$\rho_R$	0.052	0.098	0.452	0.018	0	0	0	0.03	0.678	0.766	0	0
	$\rho_Y$	0.948	0.902	0.548	0.982	1	1	1	0.97	0.3222	0.234	1	1
	Nodes	13	14	15	16	17	18	19	20	21	22	23	24
SDP1&CGP1	$\rho_R$	0	0.227	1	1	0	0	1	0.303	1	0	1	1
	$\rho_Y$	1	0.773	0	0	1	1	0	0.697	0	1	0	0
SDP2&CGP2	$\rho_R$	0	0.245	1	1	0	0	1	0.317	1	0	1	1
	$\rho_Y$	1	0.755	0	0	1	1	0	0.683	0	1	0	0
	Nodes	25	26	27	28	29	30	31	32	33	34		
SDP1&CGP1	$\rho_R$	1	1	1	0.965	0.858	1	0.734	0.91	1	0.963		
	$\rho_Y$	0	0	0	0.035	0.142	0	0.266	0.09	0	0.037		
SDP2&CGP2	$\rho_R$	1	1	1	0.963	0.856	1	0.737	0.907	1	0.96		
	$\rho_Y$	0	0	0	0.037	0.144	0	0.263	0.093	0	0.04		

The association probability that each node belongs to different clusters.  $\rho_R$  and  $\rho_Y$  means the probability belonging to red or yellow colored cluster in Figure 4 respectively.

doi:10.1371/journal.pone.0082964.t003

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

$$\hat{p}^{(n+1)} = \mathcal{P} \left( \hat{p}^{(n)} - \alpha \frac{\partial J}{\partial \hat{p}}(\hat{p}^{(n)}, \rho^{(n)}) + \beta(\hat{p}^{(n)} - \hat{p}^{(n-1)}) \right), \quad (21a)$$

$$\rho^{(n+1)} = \mathcal{P} \left( \rho^{(n)} - \alpha \frac{\partial J}{\partial \rho}(\hat{p}^{(n)}, \rho^{(n)}) + \beta(\rho^{(n)} - \rho^{(n-1)}) \right), \quad (21b)$$

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 rate  $\alpha$  and  $\beta$  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  $\rho$ . 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.

### Results

In this section, simulation experiments on artificial network, the ad hoc network with 128 nodes, is carried out to demonstrate the performance of the proposed algorithms, via comparing the clustering results with some priori quantities. Moreover, the algorithms is applied to real-world network, the social interactions between members of a karate club.

#### 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 [7,16,17,22]. It has a known partition and is constructed as follows. Suppose we choose  $N=128$  nodes, split them into 4 clusters with 32 nodes each. Assume that pairs of nodes belonging to the same clusters are linked with probability  $p_{in}$ , and pairs belonging 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

$$31p_{in} + 96p_{out} = 16. \quad (22)$$

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  $z_{out} = 96p_{out} = 5$  and generate the network according to Eq.(1). 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  $\alpha^{P1}=5, \alpha^{P2}=6, \beta^{P1}=0.52, \beta^{P2}=0.51$  in this model computation.  $J_{k-means} = 6.0687$  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  $\rho_k^E(x)$ , which is defined as

$$\rho_k^E(x) = \frac{E_k(x)}{d(x)}, \quad k=1,2,3,4, \quad x \in S, \quad (23)$$

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,  $\rho_k^E(x)$  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.(2) 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  $L^\infty$ -error of  $\rho$

$$e_\rho^m = \frac{1}{N} \sum_{x \in S} \|\rho(x) - \rho^E(x)\|_\infty, \quad (24a)$$

$$e_\rho^\infty = \max_{x \in S} \|\rho(x) - \rho^E(x)\|_\infty, \quad (24b)$$

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  $e_\rho^m$ , while the projection P2 reaches a better minimum which indicates a more accurate result. In Figure 2 we plot the probability distribution function (pdf) of  $\rho_k$  and  $\rho_k^E$  ( $k=1,2,3,4$ ). We observe that the shape of the pdf for  $\rho_k$  or  $\rho_k^E$  are almost the same. Note that all the  $\rho_k$ '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  $\rho_k^E$  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  $m = \arg \max_k \rho_k(x)$  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.

### 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 [28]. 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 [7,14–18].

There are 34 nodes in karate club network, 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  $\rho_R$  and  $\rho_Y$  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 [7]. The numerical results are shown in Table 2. It can be demonstrate again that CGP is more efficient, and P2 can reach a smaller value of  $J_{\text{min}}$ . Figure 3 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. The final association probabilities are presented in Table 3, where  $\rho_R$  and  $\rho_Y$  are the probability of belonging to the red or yellow colored group shown in Figure 4 respectively. Comparing  $\rho_R$  or  $\rho_Y$  between P1 and P2, we find that almost all the errors are less than  $10^{-2}$ , but the association probability  $\rho$  is quite different from the 0–1 distributions obtained in the  $k$ -means algorithm. Now let us compare the association probability  $\rho_R$  and  $\rho_Y$  obtained by our methods with the original partition result obtained by Zachary. In [28], 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 Figure 4(a)). We note that this hard partition deduced by fuzzy partition is more reasonable than the result of  $k$ -means (see Figure 4(b)), since node 10 is classified correctly this time.

However, we have more detailed information in fact. From Table 3, 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  $\rho$  more transparently with the color vector Eq.(23) for each node. We can conclude from Figure 4(c) 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.

### Discussion

We address the expression of lumped markov transition matrix for networks with two novel method in this paper. This can also be considered as a generalization of markov random walk dynamic in statistics for the networks. 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.(12) 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 [7]. 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 [7]. We use two datasets (Ad hoc network with 128 nodes and Karate club network) to validate algorithms and achieve good results. 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 real-world learning tasks.

### Supporting Information

**Text S1 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 4 clusters with 32 nodes each. (TXT)

**Text S2 The file karate.txt contains the network of friendships between the 34 members of a karate club at a US university, as described by Wayne Zachary in 1977.** If you use these data in your work, please cite W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33, 452–473 (1977). (TXT)

**Appendix S1 The Proof of Lemma 1.** (PDF)

## Author Contributions

Conceived and designed the experiments: FJ XJ. Performed the experiments: FJ XJ. Analyzed the data: WJ QJ. Contributed reagents/materials/analysis tools: FJ XJ. Wrote the paper: FJ XJ WJ QJ. Ad hoc

network with 128 nodes: WJ. Karate club network: OJ. Wrote Introduction and Result: FJ. Wrote Discussion and Materials and Methods: XJ. Wrote Author Summary: WJ. Wrote Discussion: QJ.

## References

- Lancichinetti A, Kivela M, Saramaki J, Fortunato S (2010) Characterizing the community structure of complex networks. *PLoS one* 5: e11976.
- Albert R, Barabasi AL (2002) Statistical mechanics of complex networks. *Reviews of modern physics* 74: 47.
- Newman M, Barabasi AL, Watts DJ (2006) *The structure and dynamics of networks*. Princeton University Press.
- Börner K, Sanyal S, Vespignani A (2007) Network science. *Annual review of information science and technology* 41: 537–607.
- Shi J, Malik J (2000) Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22: 888–905.
- Meila M, Shi J (2001) A random walks view of spectral segmentation.
- Weinan E, Li T, Vanden-Eijnden E (2008) Optimal partition and effective dynamics of complex networks. *Proceedings of the National Academy of Sciences* 105: 7907–7912.
- Chorin AJ, Kast AP, Kupferman R (1999) Unresolved computation and optimal predictions. *Communications on pure and applied mathematics* 52: 1231–1254.
- Chorin AJ (2003) Conditional expectations and renormalization. *Multiscale Modeling & Simulation* 1: 105–118.
- Lovász L (1993) Random walks on graphs: A survey. *Combinatorics, Paul erdos is eighty* 2: 1–46.
- Devijver PA, Kittler J (1982) *Pattern recognition: A statistical approach*. Prentice/Hall International Englewood Cliffs, NJ.
- Hastie T, Tibshirani R, Friedman J, Franklin J (2005) The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer* 27: 83–85.
- Lafon S, Lee AB (2006) Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28: 1393–1403.
- Girvan M, Newman ME (2002) Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99: 7821–7826.
- Newman ME (2004) Fast algorithm for detecting community structure in networks. *Physical review E* 69: 066133.
- Newman ME, Girvan M (2004) Finding and evaluating community structure in networks. *Physical review E* 69: 026113.
- Newman ME (2004) Detecting community structure in networks. *The European Physical Journal B-Condensed Matter and Complex Systems* 38: 321–330.
- Newman ME (2006) Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 103: 8577–8582.
- Li T, Liu J, Weinan E (2009) Probabilistic framework for network partition. *Physical Review E* 80: 026106.
- Hofman JM, Wiggins CH (2008) Bayesian approach to network modularity. *Physical review letters* 100: 258701.
- Schilders WH, Van Der Vorst HA, Rommes J (2008) *Model order reduction: theory, research aspects and applications*, volume 13. Springer.
- Danon L, Diaz-Guilera A, Duch J, Arenas A (2005) Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment* 2005: P09008.
- Chung FR (1997) Spectral graph theory, volume 92 of cbms regional conference series in mathematics. American Mathematical Society.
- Ma J, Wang T, Xu L (2004) A gradient by harmony learning rule on gaussian mixture with automated model selection. *Neurocomputing* 56: 481–487.
- Ma J, Gao B, Wang Y, Cheng Q (2005) Conjugate and natural gradient rules for by harmony learning on gaussian mixture with automated model selection. *International Journal of Pattern Recognition and Artificial Intelligence* 19: 701–713.
- Ma J, Wang L (2006) By harmony learning on finite mixture: adaptive gradient implementation and a floating rpl mechanism. *Neural Processing Letters* 24: 19–40.
- Qian N (1999) On the momentum term in gradient descent learning algorithms. *Neural networks* 12: 145–151.
- Zachary WW (1977) An information flow model for conic and fission in small groups. *Journal of anthropological research* : 452–473.