Optimal partition and effective dynamics of complex networks

Weinan E[†‡], Tiejun Li[§¶], and Eric Vanden-Eijnden[‖]

[†]Department of Mathematics and Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08544; [‡]School of Mathematical Sciences, Peking University, Beijing 100871, China; [§]Laboratory of Mathematics and Applied Mathematics and School of Mathematical Sciences, Peking University, Beijing 100871, China; and [‖]Courant Institute, New York University, New York, NY 10012

Edited by Alexandre J. Chorin, University of California, Berkeley, CA, and approved December 12, 2007 (received for review August 10, 2007)

Given a large and complex network, we would like to find the best partition of this network into a small number of clusters. This question has been addressed in many different ways. Here we propose a strategy along the lines of optimal prediction for the Markov chains associated with the dynamics on these networks. We develop the necessary ingredients for such an optimal partition strategy, and we compare our strategy with the previous ones. We show that when the Markov chain is lumpable, we recover the partition with respect to which the chain is lumpable. We also discuss the case of well-clustered networks. Finally, we illustrate our strategy on several examples.

partitioning | lumpability | MNCut | *k* means

In recent years we have seen an explosive growth of interest and activity on the structure and dynamics of complex networks (see refs. 1–3 for a review of these activities). This growth 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 these 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 (4–6).

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 (see, e.g. refs. 4–25 for a recent comparative review). A popular approach is to associate the network with some Markov chains. Information on the topology and dynamics of the network can then be extracted by analyzing this Markov chain, and this can be used to partition the network (4, 5, 9–12, 16, 19, 24). Particularly relevant to our work is the approach of Lafon and Lee (24) in which the stochastic matrix of the Markov chain is used to introduce a metric on the network, the diffusion distance, which can then be used to partition the network into important components (see *Comparison with Other Partitioning Strategies* for details).

In this paper, we will address such questions using the framework of optimal prediction introduced by Chorin and coworkers (26–28). In particular, we will look for the optimal partition of a large network. For that purpose, we will define distance between networks instead of distance on networks. Our strategy is different from existing approaches to graph partitioning developed in the computer science literature, such as the MNCut algorithm of Meila and Shi (5) and the algorithm of Lafon and Lee (24). It is also different from the approaches developed by physicists for network analysis (see ref. 25 for a recent review). These differences will be elucidated *Comparison with Other Partitioning Strategies*.

In what follows we develop optimal prediction theory in the context of networks and show how this framework can be used to optimally reduce the dimensionality of the network. In particular, we show how optimal prediction can be used for partitioning the networks into communities and we will compare our strategy with other dimension reduction techniques currently used in network partitioning. We also show that our approach becomes asymptotically equivalent to approaches based on spectral partitioning in the simple case when the network is well clustered, in the sense that the Markov chain presents a spectral gap with a few nearly piecewise constant eigenvectors with eigenvalues close to one. Finally, we propose an algorithm to partition networks according to our strategy and illustrate it with several examples.

APPLIED MATHEMATICS

Networks and Markov Chains

Let $G(S,E)$ be a network (or a finite weighted directed graph) with n nodes, where $E = \{e(x,y)\}_{x,y\in S}$ 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 on whether x and y are connected.

Assuming that $e(x,y) \geq 0$ for any $x,y \in S$, one can relate this network to a discrete-time Markov chain with stochastic matrix P with entries $p(x,y)$ given by

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

where $d(x)$ is the degree of the node x (29, 30). If the network is not directed, i.e., if $e(x,y) = e(y,x)$, and if we define

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

then μ satisfies the detailed balance condition

$$\mu(x)p(x,y) = \mu(y)p(y,x). \qquad [3]$$

and

$$\sum_{x\in S} \mu(x)p(x,y) = \mu(y); \qquad [4]$$

i.e., $\mu(x)$ is a stationary distribution of this Markov chain. For simplicity, throughout this paper we will assume that the network is undirected, although most of our results can be easily generalized to directed networks.

A basic idea is to infer properties of the network from those of the random walkers moving on it, and this is the idea that we will exploit. We will use the following elementary facts about Markov chains. Assume that the initial distribution of the walkers

Author contributions: W.E, T.L., and E.V.-E. designed research, performed research, and wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

¶To whom correspondence should be addressed. E-mail: tieli@pku.edu.cn

This article contains supporting information online at www.pnas.org/cgi/content/full/0707563105/DC1.

on S is $\mu_0(x)$. At a later time $t \in \mathbb{N}$, their probability distribution is $\mu_t(x) = \sum_{y\in S} \mu_0(y)p_t(y,x)$, where $p_t(x,y)$ denote the entries of the matrix P^t. To compute P^t, it is convenient to use the spectral representation. Let $\{\varphi_k\}_{k=0}^{n-1}$ and $\{\psi_k\}_{k=0}^{n-1}$ be the right and left eigenvectors of P, respectively:

$$P\varphi_k = \lambda_k\varphi_k, \quad \psi_k^T P = \lambda_k\psi_k^T, \qquad k = 0,1,\ldots,n-1. \qquad [5]$$

In the reversible case, all eigenvalues and eigenvectors are real and lie in the interval $[-1,1]$. We will order them such that $1 = \lambda_0 \geq |\lambda_1| \geq \cdots \geq |\lambda_{n-1}|$. Note that $\psi_0 = \mu$ and φ_0 is a constant vector. We also have $\psi_k(y) = \varphi_k(y)\mu(y)$. The spectral decomposition of P^t is given by

$$p_t(x,y) = \sum_{k=0}^{n-1} \lambda_k^t \varphi_k(x)\varphi_k(y)\mu(y). \qquad [6]$$

Optimal Prediction

The main question we will address is the following: Given N, which is much smaller than n, how do we find a Markov chain on a network with N nodes that best represent the dynamics of the original Markov chain? We will address this question from the viewpoint of optimal prediction introduced by Chorin and coworkers (26–28). This is a framework for carrying out model reduction from a variational viewpoint. The idea is to find a reduced model that best approximates the original model in the sense that some objective function is minimized. We will develop a framework for optimally partitioning a complex network along these lines. The space of models will be the space of Markov chains on the network S, represented by their associated stochastic matrices. The reduced models will be the lumped Markov chains on partitions of S with N elements. Such Markov chains are naturally embedded into the space of Markov chains on S itself.

Let us introduce a notion of metric on the space of Markov chains (stochastic matrices) on S with stationary distribution μ. If $p_1(x,y)$ [not necessarily equal to $p(x,y)$] is a stochastic matrix with stationary distribution μ, we define its norm as

$$\|p_1\|_\mu^2 = \sum_{x,y\in S} \frac{\mu(x)}{\mu(y)} |p_1(x,y)|^2. \qquad [7]$$

It is easy to see [see supporting information (SI) *Text*)] that if p_1 satisfies detailed balance this norm is in fact the sum of the amplitude square of the eigenvalues of p_1. The norm (Eq. **7**) allows one to define the distance between two stochastic matrices, p_1 and p_2, both with invariant distribution μ, as

$$\rho_\mu(p_1,p_2) = \|p_1 - p_2\|_\mu. \qquad [8]$$

Taking $p_1 = p$ (the stochastic matrix of the original Markov chain) and p_2 to be a stochastic matrix in a certain class, we can then find the stochastic matrix in this class, which best approximates the original one by minimizing the distance (8).

Next take a partition of S as $S = \cup_{i=1}^N S_i$ with $S_i \cap S_j = \emptyset$ if $i \neq j$. Let $\hat{p}(S_i,S_j)$ be a stochastic matrix on the state space $\mathbb{S} = \{S_1,\ldots,S_N\}$. This matrix can be naturally lifted to the space of stochastic matrices on the original state space S via

$$\tilde{p}(x,y) = \sum_{i,j=1}^N \mathbf{1}_{S_i}(x)\hat{p}(S_i,S_j)\mu_j(y), \qquad [9]$$

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

$$\mu_i(x) = \frac{\mu(x)\mathbf{1}_{S_i}(x)}{\hat{\mu}(S_i)}, \qquad \hat{\mu}(S_i) = \sum_{x\in S_i} \mu(x). \qquad [10]$$

Eq. **9** says that the probability of jump from any state in S_i is the same, and the walker enters S_j according to the equilibrium distribution. This idea is consistent with trying to coarsen the original dynamics onto the new state-space $\mathbb{S} = \{S_1,\ldots,S_N\}$ and ignore the details of the dynamics within the sets S_i. Note that $\tilde{p}$ is a stochastic matrix on S with stationary distribution μ if $\hat{p}$ is a stochastic matrix on $\mathbb{S}$ with equilibrium distribution $\hat{\mu}$ (see *SI Text*).

We now ask: Given the partition $\mathbb{S}$ and given some $t \geq 1$, what is the $\tilde{p}$ that best approximates p_t? If optimality is understood in terms of the metric (Eq. **7**), the best $\tilde{p}$ is the minimizer of

$$E(\tilde{p}) = \|\tilde{p} - p_t\|_\mu^2 \qquad [11]$$

over all $\tilde{p}$ of the form (Eq. **9**). A direct calculation (see *SI Text*) shows that the minimizer of $E(\tilde{p})$ is one in which $\hat{p}$ is given by

$$\begin{aligned}\hat{p}^*(S_i,S_j) &= \sum_{\substack{x\in S_i\\ y\in S_j}} \mu_i(x)p_t(x,y)\\ &= \sum_{k=0}^{n-1} \lambda_k^t \tilde{\varphi}_k(S_i)\tilde{\varphi}_k(S_j)\hat{\mu}(S_j), \end{aligned} \qquad [12]$$

where Eq. **6** was used and where

$$\tilde{\varphi}_k(S_i) = \frac{\sum_{x\in S_i} \mu(x)\varphi_k(x)}{\sum_{x\in S_i} \mu(x)}. \qquad [13]$$

We also have $E(\tilde{p}^*) \equiv E^*$ with

$$\begin{aligned}E^* &= \sum_{x,y\in S} \frac{\mu(x)}{\mu(y)} |p_t(x,y)|^2 - \sum_{i,j=1}^N \frac{\hat{\mu}(S_i)}{\hat{\mu}(S_j)} |\hat{p}^*(S_i,S_j)|^2\\ &\equiv \|p_t\|_\mu^2 - \|\hat{p}^*\|_{\hat{\mu}}^2. \end{aligned} \qquad [14]$$

It can be checked (see *SI Text*) that $\hat{p}^*$ and, hence,

$$\tilde{p}^*(x,y) = \sum_{i,j=1}^N \mathbf{1}_{S_i}(x)\hat{p}^*(S_i,S_j)\mu_j(y). \qquad [15]$$

are both stochastic matrices and that $\hat{\mu}(S_i)$ is an equilibrium distribution for the Markov chain on $\mathbb{S}$ with transition matrix $\hat{p}^*$. It can also be checked that $\hat{p}^*$ satisfies a detailed balance condition with respect to $\hat{\mu}$. The matrix $\tilde{p}^*$ is the stochastic matrix in the class (Eq. **9**) that best approximates the original one.

Notice that the rank of the matrix $\tilde{p}$ in Eq. **9** is at most N. Thus, the residual $E^* \geq 0$ obtained by using Eq. **12** in Eq. **9** cannot be less than what is obtained by minimizing Eq. **11** over all rank-N matrices $\tilde{p}(x,y)$. A direct calculation shows that the minimizer of Eq. **11** over all such matrices is

$$\tilde{p}^{**}(x,y) = \sum_{k=0}^{N-1} \lambda_k^t \varphi_k(x)\varphi_k(y)\mu(y). \qquad [16]$$

Note that, in general, Eq. **16** is not a stochastic matrix because some of the entries $\tilde{p}^{**}(x,y)$ can be negative. Nevertheless, $E^{**} = \|p_t - \tilde{p}^{**}\|_\mu^2$ gives a lower bound for the residual E^*.

Lumpability

A chain with stochastic matrix $p(x,y)$ is lumpable with respect to the partition $\mathbb{S} = \{S_1,\ldots,S_N\}$ iff the law of a walker (in the original chain) on these sets is itself Markov. We have the following.

Theorem 1. *Assume that $\hat{p}^*$ defined by Eq. **12** is nonsingular. Then $E^* \geq E^{**}$, and $E^* = E^{**}$ iff the Markov chain is lumpable with respect to the partition $\mathbb{S}$. In this case, we also have $\tilde{p}^{**}(x,y) = \tilde{p}^*(x,y)$.*

This theorem is an easy consequence of a result by Meila and Shi (5) who proved that the Markov chain with stochastic matrix p is lumpable on the sets $\{S_1, S_2, \ldots, S_N\}$ iff either of the following two conditions is satisfied:

1. For each S_i, $\sum_{y\in S_j} p(x,y)$ is independent of $x \in S_i$ and the matrix $\hat{p}^*(S_i,S_j) = \sum_{y\in S_j} p(x,y)$ with $x \in S_i$ is nonsingular.
2. The first N eigenvectors $\varphi_k(x), k = 0,\ldots,N-1$, are piecewise constant with respect to the partition $\{S_1, S_2, \ldots, S_N\}$.

In this case, it is easy to see that $\varphi_k(x)$ for $k = 0,\ldots,N-1$ must be of the form

$$\varphi_k(x) = \sum_{j=1}^{N} c_{k,j} \mathbf{1}_{S_j}(x), \qquad [17]$$

where $c_{0,j} = 1$ and, for $k = 1,\ldots,N-1$, the coefficients $c_{k,j}$ satisfy

$$\sum_{j=1}^{N} c_{k,j}\hat{\mu}(S_j) = 0, \qquad \sum_{j=1}^{N} c_{k,j}c_{l,j}\hat{\mu}(S_j) = \delta_{k,l}. \qquad [18]$$

The orthogonality condition of the eigenvectors implies that, for $k = N,\ldots,n-1$ and any $j = 1,\ldots,N$, we have

$$\sum_{x\in S_j} \varphi_k(x)\mu(x) = 0. \qquad [19]$$

Consequently, we have

$$\begin{aligned} \tilde{\varphi}_k(S_j) &= c_{k,j}, \qquad k = 0,\ldots,N-1 \\ \tilde{\varphi}_k(S_j) &= 0, \qquad k = N,\ldots,n-1, \end{aligned} \qquad [20]$$

which, from Eqs. **12** and **15**, implies that

$$\begin{aligned} \tilde{p}^*(x,y) &= \sum_{i,j=1}^{N} \mathbf{1}_{S_i}(x) \sum_{k=0}^{N-1} \lambda_k^t c_{k,i}c_{k,j}\mathbf{1}_{S_j}(y)\mu(y) \\ &\equiv \tilde{p}^{**}(x,y). \end{aligned} \qquad [21]$$

In the general case, the Markov chain will not be lumpable with respect to the partition $\{S_1, S_2, \ldots, S_N\}$ and Eq. **16** will not be a stochastic matrix and, hence, not an acceptable approximation. However, Eq. **15** remains the optimal approximation of the stochastic matrix of the original Markov chain, and $E^* - E^{**}$ gives a measure of the quality of this approximation in terms of lumpability. Later, we will see that the condition of *Theorem 1* are approximately satisfied for well clustered networks.

Optimal Partitioning

The next question we address is: Given N, what is the best partition $\{S_1,\ldots,S_N\}$? To answer this question, we view Eq. **14** as a function of $\{S_1,\ldots,S_N\}$, $E^* \equiv E(S_1,\ldots,S_N)$ and compute

$$\begin{aligned} &\min_{\{S_1,\ldots,S_N\}} E(S_1,\ldots,S_N) \\ &\quad = -\max_{\{S_1,\ldots,S_N\}} \sum_{i,j=1}^{N} \frac{\hat{\mu}(S_i)}{\hat{\mu}(S_j)} |\hat{p}^*(S_i,S_j)|^2. \end{aligned} \qquad [22]$$

As a direct generalization of *Theorem 1*, we have the following.

Theorem 2. *Denote by* $\{S_1^*,\ldots,S_N^*\}$ *the partition that minimizes Eq.* ***22*** *and let* $E^{**} = \|p_t - \tilde{p}^{**}\|_\mu^2$ *where* $\tilde{p}^{**}(x,y)$ *is given by Eq.* ***16****. Then* $E(S_1^*,\ldots,S_N^*) \geq E^{**}$ *and* $E(S_1^*,\ldots,S_N^*) = E^{**}$ *iff the Markov chain is lumpable with respect to the partition* $\mathbb{S}^* = \{S_1^*,\ldots,S_N^*\}$.

In other words, if the Markov chain is lumpable with respect to a partition with N sets, then the minimization problem in Eq. **22** will identify these sets. Below, we propose a variant of a k-means algorithm to solve this minimization problem. But before doing so, let us compare our criterion with other criteria introduced before for partitioning networks.

Comparison with Other Partitioning Strategies. In ref. 24, a unified framework for partitioning networks is proposed. The basic idea is to introduce the following diffusion distance between nodes on the network (this should be contrasted with Eq. **8**, which is a distance between networks):

$$\begin{aligned} D_t^2(x,y) &= \sum_{z\in S} \frac{(p_t(x,z) - p_t(y,z))^2}{\mu(z)} \\ &= \sum_{k=0}^{n-1} \lambda_k^{2t}(\varphi_k(x) - \varphi_k(y))^2. \end{aligned} \qquad [23]$$

Based on this diffusion distance, Lafon and Lee (24) suggest to partition the network by minimizing the following distortion:

$$\min_{\{S_1,\ldots,S_N\}} \sum_{i=1}^{N} \sum_{x\in S_i} \mu(x) \sum_{k=0}^{n-1} \lambda_k^{2t}(\varphi_k(x) - \tilde{\varphi}_k(S_i))^2, \qquad [24]$$

where $\tilde{\varphi}_k(S_i)$ is defined in Eq. **13**. This object, or more precisely the vector

$$\left(\lambda_1^t\tilde{\varphi}_1(S_i),\ldots,\lambda_{n-1}^t\tilde{\varphi}_{n-1}(S_i)\right), \qquad [25]$$

is called the geometric centroid in ref. 24. By expanding Eq. **24** and using Eqs. **6** and **12**, it is easy to see that Eq. **24** can be reexpressed as

$$\min_{\{S_1,\ldots,S_N\}} \left(\sum_{x\in S} p_t(x,x) - \sum_{i=1}^{N} \hat{p}^*(S_i,S_i) \right) \qquad [26]$$

or, equivalently,

$$\begin{aligned} &\max_{\{S_1,\ldots,S_N\}} \sum_{i=1}^{N} \hat{p}^*(S_i,S_i) \\ &\quad = \max_{\{S_1,\ldots,S_N\}} \sum_{i=1}^{N} \frac{\sum_{x\in S_i, y\in S_i} \mu(x)p_t(x,y)}{\sum_{x\in S_i}\mu(x)}. \end{aligned} \qquad [27]$$

This criterion is also the one used in the MNcut algorithm proposed in ref. 5 and the one of almost (or most) invariant sets introduced in ref. 31 and further developed and used in refs. 32 and 33.

To see the difference between Eqs. **27** and **22**, we note that in the case when the Markov chain is lumpable with respect to $\{S_1,\ldots,S_N\}$, the minimizer of Eq. **27** might be a partition $\{S_1',\ldots,S_N'\}$, which is different from $\{S_1,\ldots,S_N\}$, unlike the minimizer of Eq. **22**.

Here is a simple example which illustrates this point. Suppose that $S = \{1,\ldots 2n\}$ and assume that $p(x,y) = \frac{1}{2}$ if $x = 2,\ldots,2n-1$ and $y = x \pm 1$, $p(1,2) = 1 = p(2n,2n-1) = 1$, and $p(x,y) = 0$ in all other cases (i.e., each node is connected to its two direct neighbors on the line). This chain is lumpable onto a two-state chain with $S_1 = \{1,3,\ldots,2n-1\}$ and $S_2 = \{2,4,\ldots,2n\}$ with

$$\begin{aligned} \hat{p}^*(S_1,S_1) &= \hat{p}^*(S_2,S_2) = 0, \\ \hat{p}^*(S_1,S_2) &= \hat{p}^*(S_2,S_1) = 1, \end{aligned} \qquad [28]$$

and indeed the residual $E(S_1,S_2) = E^{**}$ with this choice, consistent with *Theorem 2*. On the other hand, Eq. **27** leads to the optimal partition $S_1' = \{1,2,\ldots,n\}$ and $S_2' = \{n+1,n+2,\ldots,2n\}$, and the Markov chain is not lumpable with respect to this partition. Thus partition algorithms based on Eqs. **22** and **27** are indeed different. We suspect that Eq. **27** might be more useful in the context of data segmentation (i.e., if the dynamics on the network is irrelevant),

and Eq. **22** is more preferable if one is interested in dynamical properties of the network.

Finally, it is interesting to note that although Eqs. **22** and **27** are different, they become equivalent for well clustered networks, as shown below.

The Case of Well Clustered Networks. By definition, we call a network well clustered if the associated Markov chain has a spectral gap, i.e., if the eigenvalue of P defined in Eq. **5** satisfy

$$\begin{aligned} \lambda_k &= 1 - \eta_k \delta \qquad \text{for } k = 0, \dots, N-1 \\ |\lambda_k| &< \lambda_\star \qquad \text{for } k = N \dots, n-1, \end{aligned} \qquad [29]$$

where $0 < \delta \ll 1$, and $\eta_k > 0$ and $\lambda_\star \in (0,1)$ are $O(1)$ in δ. One can show that, in this case, there exists a partition of S in N sets, $\{S_1, \dots, S_N\}$, such that the first N eigenvectors of p are approximately piecewise constant over these sets

$$\varphi_k(x) = \sum_{i=1}^{m} c_{k,i} \mathbf{1}_{S_i}(x) + o(1), \qquad k = 0, \dots, N-1, \qquad [30]$$

where $c_{0,i} = 1$ and the coefficients $c_{k,i}$ satisfy Eq. **18** for $k = 1, \dots, N-1$. This shows that the Markov chain is approximately lumpable over the sets $\{S_1, \dots, S_N\}$ and, by *Theorem 2*, the residual over these sets, $E(S_1, \dots, S_N)$, tends to E^{**} as $\delta \to 0$. In fact, in this case, we have that

$$\|\tilde{p}^* - p_{t(\delta)}\|_\mu^2 \to 0 \quad \text{as } \delta \to 0, \qquad [31]$$

where $t(\delta) = 1/\delta$ and $\tilde{p}^*(x,y)$ is given by Eq. **15**. This is Khasminskii's averaging theorem (34) in the discrete-time setting.

A similar calculation shows that $\{S_1, \dots, S_N\}$ is also the optimal partition according to Eq. **24** with $t(\delta) = 1/\delta$ in $p_t(x,y)$. Thus, for well clustered networks, Eqs. **22** and **24** are asymptotically equivalent.

Algorithmic Aspects

In practice, it is important that the minimization problem in Eq. **22** be tractable. In ref. 24, it is shown that minimization problem Eq. **24** can be solved using the k-means algorithm (35). Here we show that a variant of this algorithm can also be used to handle Eq. **22**. Given an initial partition $\{S_i^{(0)}\}_{i=1}^N$, for $n \geq 0$ use

$$S_i^{(n+1)} = \{x : i = \operatorname*{argmin}_j \bar{E}(x, S_j^{(n)})\}, \qquad [32]$$

where

$$\bar{E}(x, S_j) = \sum_{k=1}^{N} \sum_{y \in S_k} \mu(x)\mu(y) \left| \frac{p_t(x,y)}{\mu(y)} - \frac{\hat{p}^*(S_j, S_k)}{\hat{\mu}(S_k)} \right|^2. \qquad [33]$$

This algorithm has the advantage that it converges very fast even though, like all k-means algorithms, it may not converge to the global minimum. In fact, the situation is slightly worse than usual here because $\bar{E}(x, S_j)$ depends implicitly on the previous partition due to the presence of S_k in Eq. **33**. As a result, the objective function $E(S_1, \dots, S_N)$ is not guaranteed to decrease at every iteration. This problem can be solved by terminating the iteration if the objective function increases or remains constant, then repeat the calculation with several initial partitions, $\{S_i^{(0)}\}_{i=1}^N$ and keep the best result (as is usually done with k-means algorithms). This is the procedure we used on the examples *Results*.

Results

Zachary's Karate Club Network. As a first test, we used the well known example of the karate club network (36). This network was constructed by Wayne Zachary after he observed social interactions between members of a karate club at an American university. Soon after, a dispute arose between the club's administrator and main teacher and the club split into two smaller clubs. We used Zachary's original network in our procedure. As an initial condition for the k-means algorithm, we took random partitions. The partition that we obtained is shown in Fig. 1 and corresponds to $S_1 = \{1 : 8, 10 : 14, 17, 18, 20, 22\}$ and $S_2 = \{9, 15, 16, 19, 21, 23 : 34\}$. This is very similar to the actual structure of the smaller clubs observed by Zachary after the split: $S_1 = \{1 : 8, 11 : 14, 17, 18, 20, 22\}$ and $S_2 = \{9, 10, 15, 16, 19, 21, 23 : 34\}$. Only one node, node 10, is misclassified.

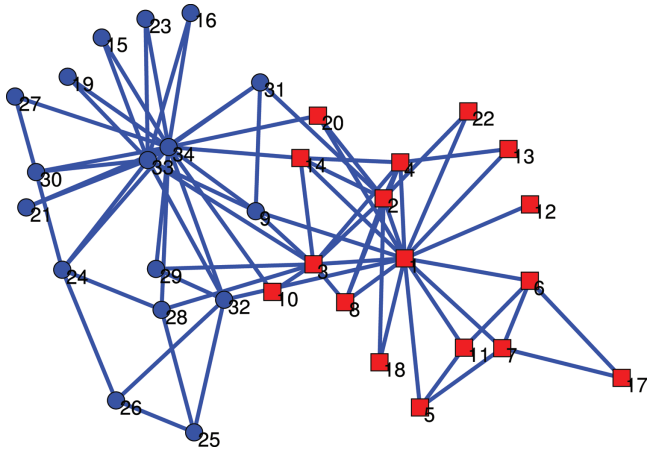

Fig. 1. The two clusters identified by our approach in Zachary's karate club network (36). The administrator and the instructor are represented by nodes 1 and 33, respectively. As initial condition for our k-means algorithm, we took random partitions. We obtained $S_1 = \{1 : 8, 10 : 14, 17, 18, 20, 22\}$ and $S_2 = \{9, 15, 16, 19, 21, 23 : 34\}$, which is very similar to Zachary's actual observation: Only one node, node 11, is misclassified.

Ad Hoc Networks. As second test, we used the example of the ad hoc networks which were originally proposed in ref. 16 and were used to compare the performances of various partitioning strategies in ref. 25. These networks have a known community structure and are constructed as follows. They have $n = 128$ nodes, split into 4 communities containing 32 nodes each. Pairs of nodes belonging to the same communities are linked with probability p_{in}, and pairs belonging to different communities with probability p_{out}. These values are chosen so that the average node degree, k, is fixed at $k = 16$. In other words, p_{in} and p_{out} are related as

$$31p_{\text{in}} + 96p_{\text{out}} = 16. \qquad [34]$$

Typically, we define z_{out} as the average number of links a node has to nodes belonging to any other communities, i.e. $z_{\text{out}} = 96p_{\text{out}}$, and we use this quantity as a control parameter. The larger the z_{out}, the more diffuse the communities become.

We first used our strategy to partition the network by assuming that the number of communities is known, $N = 4$. We considered several values of z_{out} between 0 and 8 and calculated the fraction f of correctly identified nodes by our procedure (to compute this fraction f, we used the criterion proposed in ref. 16 and used in ref. 25 for a comparative study). To make our results less dependent on the specific network chosen, for each value of z_{out}, we took 100 realizations of the network and computed the mean and standard deviation of f over these 100 realizations. To apply our k-means algorithm, in each case, we took 100, 300, 500, and finally 1,000 random initial partitions and kept the best result (i.e., the one with the smallest residual E^*). The final result for the mean of f is shown in Fig. 2. It shows that our procedure performs very well at identifying the right communities all the way up to $z_{\text{out}} = 7.5$ (where $z_{\text{out}}/k = 0.4688$ and $f = 0.93$ with 500 trials) and only deteriorates for $z_{\text{out}} = 8$ (where $z_{\text{out}}/k = 0.5$

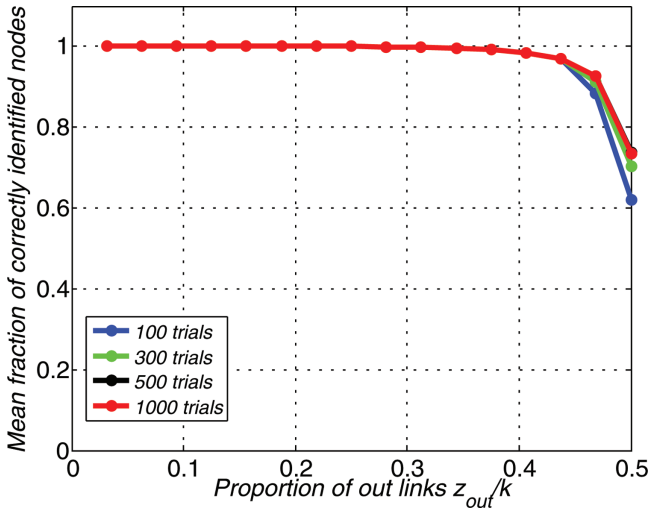

Fig. 2. The mean fraction of correctly identified nodes versus the proportion of links towards the other communities. The four curves show the results of our *k*-means algorithm with 100, 300, 500, and 1,000 random initial partitions. As can be seen, the results improve as the number of initial conditions is increased, but the results eventually saturate (the curve with 500 trials can barely be distinguished below the one with 1,000 trials). The results show that our algorithm is among the very best of those compared in ref. 25.

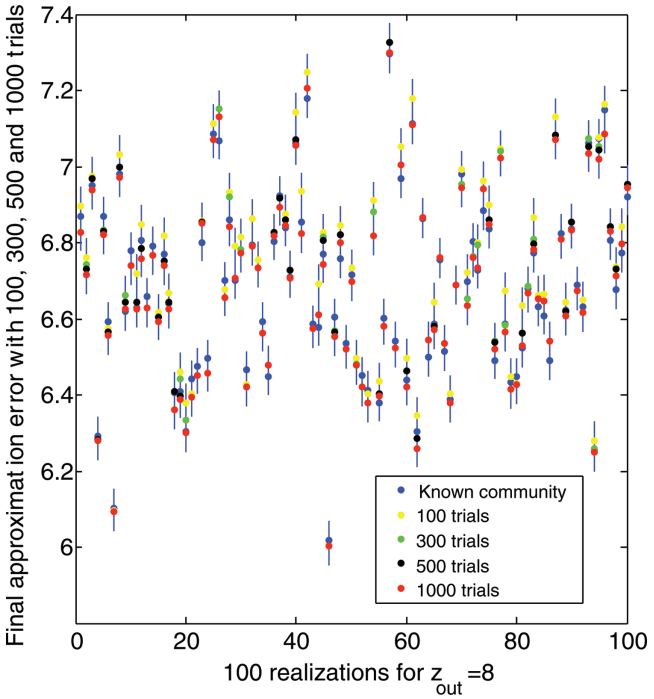

Fig. 3. The residual E^* of the *k*-means algorithm obtained for 100 independent realizations of the ad hoc network with $z_{out} = 8$ using 100, 300, 500, and finally 1,000 random initial partitions. Also shown is the residual E^* calculated with the known partition of the network. As can be seen, the actual residual E^* identified by our *k*-means algorithm is often smaller than the one computed with the known community. This result reflects the very diffuse nature of the community structure in the ad hoc network when $z_{out} = 8$. The vertical lines on the graph act as a visual aid to identify the various points associated with the different realizations.

and $f = 0.7367$ with 500 trials). Even in this case, however, our method remains very competitive compared with the techniques listed in ref. 25, being outperformed only by two of these techniques at the last data point, $z_{out}/k = 0.5$. Next, we discuss in more details the performance of our technique in terms of accuracy and efficiency.

Accuracy. The results shown in Fig. 2 indicate that our technique identifies a fraction f of >90% of the modes correctly up to $z_{out} = 7.5$. Our method does deteriorate, however, when $z_{out} = 8$, and it is interesting to investigate what happens then. The first issue that we need to address is whether our *k*-means algorithm does identify the minimum E^* of the objective function (Eq. **22**) because this depends on the number of random initial partitions used. The result in Fig. 2 indicates that this is indeed the case, at least if the number of initial partitions is >500. The result shown in Fig. 3 also corroborates this finding by displaying the residual E^* of the *k*-means algorithm obtained for 100 independent realizations of the ad hoc network with $z_{out} = 8$ using 100, 300, 500, and finally 1,000 random initial partitions. Fig. 3 also explains the difficulty inherent in partitioning networks with a diffuse community structure such as the ad hoc networks when $z_{out} = 8$. Indeed, it can be seen that the residual E^* calculated from the known community structure is typically larger than the residual identified by our *k*-means algorithm. Thus, at least in terms of lumpability, the community structure has often become so diffusive when $z_{out} = 8$ that another set of communities is actually better, and this is why the fraction f of correctly identified nodes becomes smaller in that case.

Efficiency. Our procedure is very competitive with respect to those compared in ref. 25 in terms of accuracy, but how does it do in terms of efficiency? It can be shown (see *SI Text*) that the cost of every iteration in our *k*-means algorithm (i.e., the cost of evaluating Eqs. **32** and **33**) is $O(N(n + m))$, where N is the number of communities, n is the number of nodes in the network, and m is the numbers of edges. This number thus provides a lower bound on the cost of our *k*-means algorithm. To estimate the algorithm's actual cost, we still need to estimate how many random initial partitions must be used to identify the actual minimum and how many iterations the algorithm takes before reaching a local minimum. These numbers are much harder to compute analytically. In tests with ad hoc networks of increasing sizes, we observed that these numbers seem to depend only weakly on the size of the network: Typically, 500 random initial partitions are enough, and for each the algorithm converges in a few tenths of an iteration, even for large networks (i.e., with n between 128 and 1,280). Should these results be generic, this would make our method one of the least expensive among the techniques compared in ref. 25. This point, however, requires further study.

Determining the number N of communities. So far, we have assumed that the number of communities, N, was given. In many applications, however, this number is unknown beforehand and needs to be determined by the partitioning technique itself. One way to do so with our method is to apply it with several values of N and compare the results. For instance, setting $z_{out} = 6$ and applying our method with $N = 2, 3, \ldots, 8$ communities, we observed the following. When $N = 2$, our technique identified one community with 32 nodes which was one of the correct communities, and one with 96 nodes, which was the union of the remaining three correct communities. When $N = 3$, our technique correctly identified two communities with 32 nodes and lumped together the other two. When $N = 4$, we got the correct classification. When $N = 5$, we got the correct classification, except that one of the communities of 32 nodes was split into two. Similarly, when $N = 6$, $N = 7$, and $N = 8$, two, three, and four of the correct communities, respectively, with 32 nodes were split into two.

How can we *a priori* identify which value of N was the actual one from these results? The most natural method is to look at the relative residual, $E^* - E^{**}$, in each case. In the example of the

ad hoc network with $z_{out} = 6$, we observed that $E^* - E^{**}$ increases slightly from $N = 2$ to $N = 4$, then faster when $N > 4$. This result seems to corroborate that $N = 4$ is the optimal choice (because using more communities worsen the result more significantly), but this point too deserves further study.

Discussion

In summary, we have proposed an approach to partition complex networks based on the framework of optimal prediction. The approach is tailored to situations for which the dynamics on the network matters and it gives the coarse network which respects best the dynamics on the original network. As we have shown, however, our approach can also be used in the context of network partitioning. In this context, it may be an attractive alternative to existing techniques both in terms of accuracy and computational cost.

ACKNOWLEDGMENTS. This work was supported in part by Office of Naval Research Grants N00014-01-0674 (to W.E) and N00014-04-1-0565 (to E.V.-E.), National Science Foundation of China Grants 10401004 (to. T.L.) and DMS02-09959 and DMS02-39625 (to E.V.-E.), and National Basic Research Program Grant 2005CB321704 (to T.L.). This work began while W.E was visiting the Beijing International Center for Mathematical Research, and it was completed while T.L. was visiting the Program in Applied and Computational Mathematics at Princeton University. Support and hospitality from these organizations are gratefully acknowledged. The network figures were produced with the help of the software PAJEK (V. Batagelj and A. Mrvar; available at http://vlado.fmf.uni-lj.si/pub/networks/pajek).

1. Barabási AL, Albert R (2002) *Rev Mod Phys* 74:47–97.
2. Newman MEJ, Barabási AL, Watts DJ (2005) *The Structure and Dynamics of Networks* (Princeton University, Princeton).
3. National Research Council (2005) *Network Science* (Natl Acad Press, Washington, DC).
4. Shi J, Malik J (2000) *IEEE Trans Pattern Anal Mach Intell* 22:888–905.
5. Shi J, Meilă M (2001) in *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics* (Kaufmann, San Francisco), pp 92–97.
6. Coifman RR, Lafon S (2006) *Appl Comput Harmon Anal* 21:5–30.
7. van Dongen S (2000) PhD thesis (Univ of Utrecht, Netherlands).
8. Eckmann J-P, Moses E (2002) *Proc Nat Acad Sci USA* 99:5825–5829.
9. Girvan M, Newman MEJ (2002) *Proc Nat Acad Sci USA* 99:7821–7826.
10. Vukadinovic D, Huang P, Erlebach T (2002) *Lect Notes Comput Sci* 2346:83–95.
11. Capocci A, Servedio V, Colaiori F, Caldarelli G (2004) *Lect Notes Comput Sci* 3243:181–187.
12. Donetti L, Muñoz MA (2004) *J Stat Mech* 2004:P10012.
13. Fortunato S, Latora V, Marchiori M (2004) *Phys Rev E* 70:056104.
14. Guimerà R, Sales M, Amaral LAN (2004) *Phys Rev E* 70:025101.
15. Newman MEJ (2004) *Phys Rev E* 69:066133.
16. Newman MEJ, Girvan M (2004) *Phys Rev E* 69:026113.
17. Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D (2004) *Proc Natl Acad Sci USA* 101:2658–2663.
18. Reichardt J, Bornholdt S (2004) *Phys Rev Lett* 93:218701.
19. Wu F, Huberman B (2004) *Eur Phys J B* 38:331–338.
20. Zhou H, Lipowsky R (2004) *Lect Notes Comput Sci* 3038:1062–1069.
21. Duch J, Arenas A (2005) *Phys Rev E* 72:027104.
22. Palla G, Derenyi I, Farkas I, Vicsek T (2005) *Nature* 435:814–818.
23. Newman MEJ (2006) *Proc Natl Acad Sci USA* 103:8577–8582.
24. Lafon S, Lee A (2006) *IEEE Trans Pattern Anal Mach Intell* 28:1393–1403.
25. Danon L, Díaz-Guilera A, Duch J, Arenas A (2005) *J Stat Mech* 2005:P09008.
26. Chorin A, Kast A, Kupferman R (1999) *Commun Pure Appl Math* 52:1231–1254.
27. Chorin A, Hald OH, Kupferman R (2000) *Proc Nat Acad Sci USA* 97:2968–2973.
28. Chorin A (2003) *Multiscale Model Simulat* 1:105–118.
29. Lovász L (1993) in *Combinatorics, Paul Erdös is Eighty* (Janos Bolyai Math Soc, Budapest), Vol 2, pp 1–46.
30. Chung FRK (1997) *Spectral Graph Theory* (Am Math Soc, Providence, RI).
31. Dellnitz M, Junge O (1999) *SIAM J Numer Anal* 36:491–515.
32. Deuhard P, Dellnitz M, Junge O, Schütte Ch (1999) *Lect Notes Comp Sci Eng* 4:98–115.
33. Deuhard P, Huisinga W, Fischer A, Schütte Ch (2000) *Linear Alg Appl* 315:39–59.
34. Khasminskii RZ (1966) *Theor Probab Appl* 11:211–228.
35. Hastie T, Tibshirani R, Friedman J (2001) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, New York).
36. Zachary WW (1977) *J Anthrop Res* 33:452–473.