Overlapping community detection using Bayesian non-negative matrix factorization

Ioannis Psorakis,* Stephen Roberts, and Mark Ebden

Pattern Analysis and Machine Learning Research Group, Department of Engineering Science, University of Oxford, Oxford, United Kingdom

Ben Sheldon

Edward Grey Institute, Department of Zoology, University of Oxford, Oxford, United Kingdom

(Received 21 February 2011; revised manuscript received 17 May 2011; published 22 June 2011)

Identifying overlapping communities in networks is a challenging task. In this work we present a probabilistic approach to community detection that utilizes a Bayesian non-negative matrix factorization model to extract overlapping modules from a network. The scheme has the advantage of soft-partitioning solutions, assignment of node participation scores to modules, and an intuitive foundation. We present the performance of the method against a variety of benchmark problems and compare and contrast it to several other algorithms for community detection.

DOI: 10.1103/PhysRevE.83.066114 PACS number(s): 89.75.Hc, 05.10.−a, 87.23.Ge

I. INTRODUCTION

Community structure, or modular organization, is a significant property of real-world networks as it is often considered to account for the functional characteristics of the system under study [1–4]. Although the notion of "community" appears intuitive [2,3] (for example, people form cliques in social networks and web pages of similar content have links to one another) there is no disciplined, context-independent definition of what communities are [2,4]; we adopt here the loose definition that these modules are subgraphs with more links connecting the nodes inside than outside them [2,3,5]. The task of identifying such subgraphs in a given network can be challenging [1,2], both in terms of recognition and computational feasibility.

One of the key issues in community detection is describing the overlapping nature of network modules. Traditional "hard-partitioning" algorithms [6–9] may yield excellent identification results, but omit the important characteristic of real-world networks where a node may participate in more than one group (for example, individuals belong to various social circles and scientists may participate in more than one research group). A popular approach to tackle this problem is the clique percolation method (CPM) by Palla *et al.* [10], which is based on the belief that communities are unions of adjacent k cliques (complete graphs with k nodes) and that intercommunity regions of the network do not possess such strong link density. Because communities are defined as the largest network component containing adjacent k cliques (cliques sharing $k-1$ nodes), overlaps arise naturally between modules. Performance may be compromised for networks with weak clique presence, because many nodes are left out, or for networks with very high link density, because we reach the trivial solution of describing the network as a single community.

Other approaches include the algorithm of Lancichinetti *et al.* [11], which seeks a local maximum of the community "fitness" function (based on internal link density) by modifying nodes' community "appropriateness" scores through a series of inclusion-exclusion moves. The work of Evans and Lambiotte [12] detects communities of links—in contrast to node communities, which occupy the vast body of the literature [2,3]—after losslessly transforming the adjacency matrix to a line graph. By assigning links, rather than nodes, among communities, the method allows a node to participate naturally in more than one group, as determined by the labels assigned to its adjacent links. The advantages of this approach have also been presented by Ahn *et al.* in [13]. Finally, Nepusz *et al.* [14], propose that communities should comprise "similar" nodes, assuming that a distance metric between nodes is defined and that similarity is inversely related to distance. When a partition matrix, representing a reasonable community partition, is multiplied by itself it would then be expected to approximate the similarity matrix; this leads to a nonlinear constrained optimization problem. The number of communities of the proposed incidence matrix is selected by performing multiple runs and selecting the one with the highest fitness score based on a Newman modularitylike function. Further discussion on similar methods, along with a comprehensive review of community detection algorithms in general, is presented in a survey by Fortunato [2].

In this work we propose a probabilistic approach to community detection based on computationally efficient Bayesian non-negative matrix factorization (NMF) [15]. The advantages of this methodology are (i) overlapping or soft-partitioning solutions, where communities are allowed to share members; (ii) soft-membership distributions, which quantify "how strongly" each individual participates in each group; (iii) excellent module identification capabilities; and (iv) the method does not suffer from the drawbacks of modularity optimization methods, such as the resolution limit. In the following section we present the theoretical foundations of our approach along with an illustrative example to provide intuition behind the method. Following the model formulation section, we test our algorithm on a variety of artificial and real-world benchmark problems and present our experimental results.

II. MODEL FORMULATION

A. Generative model

We consider the generative graphical model of Fig. 1. The observed variable v_{ij} denotes the non-negative count

[*] ioannis.psorakis@eng.ox.ac.uk

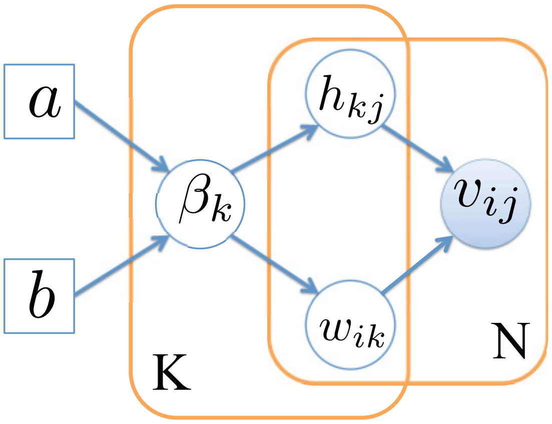


FIG. 1. (Color online) Graphical model showing the generation of count processes **V** from the latent structure **W** and **H**, the components of which have scale hyperparameters β_k. The hyperhyperparameters a, b are fixed.

of interactions between two individuals i,j in a weighted undirected network with adjacency matrix $\mathbf{V} \in \mathbb{R}_+^{N\times N}$. In the community detection context, we assume that there are a number K of "hidden" classes of nodes in the network that affect v_{ij}. Thus we can define allocations of nodes to communities as latent (unobserved) variables that allow us to explain the increased interaction density in certain regions of the network: the more two individuals interact the more likely they are to belong to the same communities, and vice versa.

We assume that the pairwise interactions described in **V** are influenced by an unobserved *expectation network* $\hat{\mathbf{V}}$, where each $\hat{v}_{ij}$ denotes the expected number of interactions (or expected link weight) that take place between i and j. The expectation network is composed of two non-negative matrices $\mathbf{W} \in \mathbb{R}_+^{N\times K}$ and $\mathbf{H} \in \mathbb{R}_+^{K\times N}$ so that $\hat{\mathbf{V}} = \mathbf{WH}$. We hence model each interaction v_{ij} as drawn from a Poisson distribution with rate $\hat{v}_{ij} = \sum_{k=1}^{K} w_{ik}h_{kj}$. The inner rank K denotes the unknown number of communities and each element $k \in \{1,\ldots,K\}$ in row i of **W** and column j of **H** represents the contribution of a single latent community to $\hat{v}_{ij}$. In other words, the expected number of times $\hat{v}_{ij}$ that two individuals i,j interact is a result of their *mutual participation* in the same communities.

In the typical community-detection setting, the value of K, which we call *complexity* or *model order*, is initially unknown. In previous work [16,17], the issue of inferring the appropriate number of communities has been addressed by performing multiple runs for various K and selecting one that yields the highest Newman modularity Q [5]. In our setting, the appropriate model order arises naturally from a *single* run, by placing *shrinkage* or *automatic relevance determination* priors [18] with scale hyperparameters $\boldsymbol{\beta} = \{\beta_k\}$ on the latent variables w_{ik},h_{kj}, as presented in [15]. By starting with a large K (say N, which is the maximum possible number of communities), the effect of these priors is to moderate complexity by "shrinking" close to zero irrelevant columns of **W** and rows of **H** that do not contribute to explaining the observed interactions **V**. This is achieved by placing a distribution over the latent variables w_{ik},h_{kj} whose expectation approaches zero unless nonzero values are required by the data. This approach avoids the computational load of multiple runs and is free of the resolution bias problems [19] of modularity.

Based on the graphical model of Fig. 1, where the distribution of β_k is parametrized by fixed hyperhyperparameters a and b, we express the joint distribution over all variables as

$$p(\mathbf{V},\mathbf{W},\mathbf{H},\boldsymbol{\beta}) = p(\mathbf{V}|\mathbf{W},\mathbf{H})p(\mathbf{W}|\boldsymbol{\beta})p(\mathbf{H}|\boldsymbol{\beta})p(\boldsymbol{\beta}), \tag{1}$$

hence the posterior over model parameters given the observations is

$$p(\mathbf{W},\mathbf{H},\boldsymbol{\beta}|\mathbf{V}) = \frac{p(\mathbf{V}|\mathbf{W},\mathbf{H})p(\mathbf{W}|\boldsymbol{\beta})p(\mathbf{H}|\boldsymbol{\beta})p(\boldsymbol{\beta})}{p(\mathbf{V})}. \tag{2}$$

B. Posterior-based cost function

We aim to maximize the model posterior given the observations, or equivalently, to minimize the negative log posterior, which may be regarded as an energy (or error) function $\mathcal{U}$. Noting that $p(\mathbf{V})$ is a constant with respect to the inference over the model's free parameters, we hence define

$$\begin{aligned}\mathcal{U} = &-\log p(\mathbf{V}|\mathbf{W},\mathbf{H}) - \log p(\mathbf{W}|\boldsymbol{\beta})\\ &- \log p(\mathbf{H}|\boldsymbol{\beta}) - \log p(\boldsymbol{\beta}),\end{aligned} \tag{3}$$

where the first term is the log likelihood of our data, derived from the probability $p(\mathbf{V}|\mathbf{W},\mathbf{H}) = p(\mathbf{V}|\hat{\mathbf{V}})$ of observing every interaction v_{ij} given a Poisson rate $\hat{v}_{ij}$. Therefore we express the negative log likelihood of a single observation v_{ij} as

$$-\log p(v|\hat{v}) = -v\log\hat{v} + \hat{v} + \log v!. \tag{4}$$

Using the Stirling approximation to second order, namely

$$\log v! \approx v\log v - v + \tfrac{1}{2}\log(2\pi v), \tag{5}$$

Eq. (4) can be written as

$$-\log p(v|\hat{v}) \approx v\log\left(\frac{v}{\hat{v}}\right) + \hat{v} - v + \frac{1}{2}\log(2\pi v), \tag{6}$$

thus the full negative log likelihood for all the observed data is

$$\begin{aligned}-\log p(\mathbf{V}|\hat{\mathbf{V}}) &= -\sum_{i=1}^{N}\sum_{j=1}^{N}\log p(v_{ij}|\hat{v}_{ij})\\ &\simeq \sum_{i=1}^{N}\sum_{j=1}^{N}\left(v_{ij}\log\frac{v_{ij}}{\hat{v}_{ij}} + \hat{v}_{ij} - v_{ij}\right.\\ &\quad\left. + \frac{1}{2}\log(2\pi v_{ij})\right) + \kappa,\end{aligned} \tag{7}$$

where κ is a constant.

Following [15] and similar models for probabilistic Principal Component Analysis [20] and Independent Component Analysis [21–23], we place independent half normal priors over the columns of **W** and rows of **H** with precision (inverse variance) parameters $\boldsymbol{\beta} \in \mathbb{R}^K = [\beta_1,\ldots,\beta_K]$. The negative log priors over **W** and **H** are then given by

$$\begin{aligned}-\log p(\mathbf{W}|\boldsymbol{\beta}) &= -\sum_{i=1}^{N}\sum_{k=1}^{K}\log\mathcal{HN}\left(0,\beta_k^{-1}\right)\\ &= \sum_{i=1}^{N}\sum_{k=1}^{K}\left(\frac{1}{2}\beta_k w_{ik}^2\right) - \frac{N}{2}\log\beta_k + \kappa,\end{aligned} \tag{8}$$

Algorithm: Community detection using NMF

```
Require: adjacency matrix $\mathbf{V} \in \mathbb{R}_+^{N\times N}$, initial $K_0$, fixed Gamma hyperparameters $a$, $b$.
Define: matrix operation $\frac{\mathbf{X}}{\mathbf{Y}}$ as element-by-element division.
Define: matrix operation $\mathbf{X}\cdot\mathbf{Y}$ as element-by-element multiplication.
Define: $\mathbf{B} \in \mathbb{R}^{K\times K}$ as a matrix with elements $\beta_k$ in the diagonal and zero elsewhere.
Auxiliary inputs $\mathbf{W}_0,\mathbf{H}_0$ from previous runs. If not present, initialize to random values.
for $i = 1$ to $n_{\mathrm{iter}}$
    $\mathbf{H} \leftarrow \left(\frac{\mathbf{H}}{\mathbf{W}^{\mathsf{T}}\mathbf{1}+\mathbf{B}\mathbf{H}}\right)\cdot\left[\mathbf{W}^{\mathsf{T}}\left(\frac{\mathbf{V}}{\mathbf{W}\mathbf{H}}\right)\right]$
    $\mathbf{W} \leftarrow \left(\frac{\mathbf{W}}{\mathbf{1}\mathbf{H}^{\mathsf{T}}+\mathbf{W}\mathbf{B}}\right)\cdot\left[\left(\frac{\mathbf{V}}{\mathbf{W}\mathbf{H}}\right)\mathbf{H}^{\mathsf{T}}\right]$
    $\beta_k \leftarrow \frac{N+a-1}{\frac{1}{2}\left(\sum_i w_{ik}^2+\sum_j h_{kj}^2\right)+b}$
end for
$K_\star \leftarrow$ number of nonzero columns of $\mathbf{W}$ or rows of $\mathbf{H}$
$\mathbf{W}_\star \leftarrow \mathbf{W}$ with zero columns removed
$\mathbf{H}_\star \leftarrow \mathbf{H}$ with zero rows removed
return $\mathbf{W}_\star \in \mathbb{R}_+^{N\times K_*}, \mathbf{H}_\star \in \mathbb{R}_+^{K_*\times N}$
```

$$
\begin{aligned}
-\log p(\mathbf{H}|\boldsymbol{\beta}) &= -\sum_{k=1}^{K}\sum_{j=1}^{N}\log\mathcal{HN}\left(0,\beta_k^{-1}\right)\\
&= \sum_{k=1}^{K}\sum_{j=1}^{N}\left(\frac{1}{2}\beta_k h_{kj}^2\right) - \frac{N}{2}\log\beta_k + \kappa. \qquad (9)
\end{aligned}
$$

Each β_k controls the importance of community k in explaining the observed interactions; large values of β_k denote that column k of $\mathbf{W}$ and row k of $\mathbf{H}$ have elements lying close to zero and therefore represent irrelevant communities. By assuming the β_k are independent[1] we place a standard Gamma distribution over them with fixed hyperhyperparameters a,b [25]. The negative log hyperpriors are thus

$$
\begin{aligned}
-\log p(\boldsymbol{\beta}) &= -\sum_{k=1}^{K}\log\mathcal{G}(\beta_k|a,b)\\
&= \sum_{k=1}^{K}[\beta_k b - (a-1)\log\beta_k] + \kappa. \qquad (10)
\end{aligned}
$$

The objective function $\mathcal{U}$ of Eq. (3) can be expressed as the sum of Eqs. (7) and (10):

$$
\begin{aligned}
\mathcal{U} = &\sum_i\sum_j\left[v_{ij}\log\left(\frac{v_{ij}}{\hat{v}_{ij}}\right) + \hat{v}_{ij}\right]\\
&+\frac{1}{2}\sum_k\left[\left(\sum_i\beta_k w_{ik}^2\right) + \left(\sum_j\beta_k h_{kj}^2\right) - 2N\log\beta_k\right]\\
&+\sum_k[\beta_k b_k - (a_k - 1)\log\beta_k] + \kappa. \qquad (11)
\end{aligned}
$$

C. Parameter inference

To optimize Eq. (11) for $\mathbf{W}$,$\mathbf{V}$, and $\boldsymbol{\beta}$ we follow Refs. [15,26–28] by adopting the fast fixed-point algorithm presented in [15] that involves consecutive updates of $\mathbf{W}$,$\mathbf{H}$, and $\boldsymbol{\beta}$ until a convergence measure has been satisfied (a maximum number of iterations, or a tolerance on the cost function). The pseudocode is presented in the algorithm below; we discuss memory and computational efficiency in the discussion section of this paper. The solution consists of $\mathbf{W}_\star \in \mathbb{R}_+^{N\times K_\star}$ and $\mathbf{H}_\star \in \mathbb{R}_+^{K_\star\times N}$ for which $\hat{\mathbf{V}} = \mathbf{W}_\star\mathbf{H}_\star$ represents the expectation network given our observation data $\mathbf{V}$ and prior assumptions. The inner rank $K_\star$ denotes the inferred number of latent modules in the network.

In the case of undirected graphs, $\mathbf{W}_\star = \mathbf{H}_\star^{\mathsf{T}}$ (as $\mathbf{V}$ is symmetric) and represents the $N \times K_\star$ incidence matrix of a bipartite graph of N nodes and $K_\star$ communities. Each element $w_{ik}^\star$ (or $h_{ki}^\star$) denotes the *degree of participation* of individual i into community k while each normalized row of $\mathbf{W}_\star$ (or column of $\mathbf{H}_\star$) expresses a *soft-membership* distribution over communities given a certain node. Therefore this bipartite graph describes the *overlapping* mesoscopic structure of our network, where nodes are allocated to multiple groups with varying participation score.

The overall interaction matrix $\mathbf{V}$ is approximated by a sum $\hat{\mathbf{V}} = \sum_k w_{\cdot k}^\star h_{k\cdot}^\star$, where $w_{\cdot k}^\star$ is the column and $h_{k\cdot}^\star$ row vector of the community matrices $\mathbf{W}_\star$ and $\mathbf{H}_\star$, respectively. Therefore $\hat{\mathbf{V}}$ is a summation of K rank 1 matrices $\hat{\mathbf{V}}^{(k)} = w_{\cdot k}^\star h_{k\cdot}^\star$ and each $\hat{\mathbf{V}}^{(k)}$ denotes the expected number of pairwise interactions *in the context of community* k. Thus if two nodes i,j have nonzero participation rates $w_{ik}^\star$, $h_{kj}^\star$ to community k, then the average link weight for this dyad would also be nonzero due to $\hat{\mathbf{V}}_{ij}^{(k)} = w_{ik}^\star h_{kj}^\star$.

Based on the above, our model assumes that the joint membership of two nodes in the same community raises the probability of a link existing between them. Therefore our method performs best when modules are dense, with the best-case scenario being that each community is a fully connected subgraph.

[1] This corresponds to the belief that the existence of one community is not dependent upon others. Clearly, there will be situations in which this can be extended to allow for a full interdependency between communities. We do not consider this here, however. Allowing dependency is similar to the notion of *structure priors* discussed in [24].

In the next section, we present an illustrative example of this community extraction scheme, followed by experimental results from various artificial and real-world networks.

III. APPLICATIONS

A. An illustrative example

Consider the small toy graph of Fig. 2 with $N = 16$ nodes and $M = 25$ edges of varying weights. We extract the mesoscopic (community) structure of this network using NMF, along with the popular extremal optimization (EO) [9], spectral partitioning (SP) [29] and weighted clique percolation method (wCPM) [30].

Although a trivial problem at first glance, each community detection method we applied yielded different modules and node allocations, as seen in Fig. 3. Hard-partitioning methods such as EO and SP produce such inconsistencies mainly due to the "broker" nature of nodes such as 6,9, or 10 that lie on high-flow paths in the network, making them difficult to assign on one module or the other [2]. Although this issue is addressed by wCPM, which allows node membership to multiple modules, it does not provide some measure of "participation strength" or "degree of belief" in membership.

By applying NMF we extracted $K_\star = 4$ overlapping groups as shown in Fig. 4. We can see that our method does not force node allocations to a single group, but instead allows the broker individuals described above to participate in more than one community. This *soft-partitioning* solution allows us to describe the different aspects of an individual's sociality as a collection of (possibly intersecting) sets of nodes, where each set may play a different role or function in the whole network [2].

Allocating nodes to multiple modules, as in Fig. 4, is only one part of the solution. We also capture the *degree of participation* of nodes in each community by using the incidence matrix $\mathbf{W}_\star$ described in the previous section. Figure 5(a) shows $W_\star \in \mathbb{R}_+^{16\times 4}$ where different colors (gray shades) indicate various levels of participation of nodes in communities. We can see that the matrix is not of a clear block diagonal form, as an individual can have some form of membership in multiple groups.

In our framework, community allocation is not a Boolean decision but a *belief*; each node is assigned a membership distributed over communities, as seen in Fig. 5(b). We can see that mediator nodes of high "betweenness," such as $i = 6$, have a more entropic distribution (similar to the concept of "bridgeness" [14]) while for nodes such as $i = 4$ or $i = 14$ we have much more confident allocations.

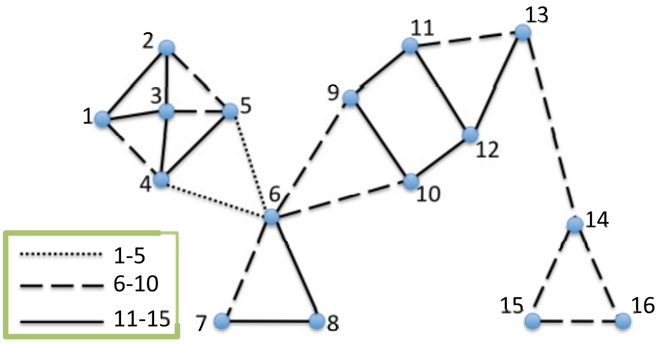


FIG. 2. (Color online) An undirected weighted toy graph with 16 nodes. The three different line styles denote the differing strengths of interaction within each pair of nodes.

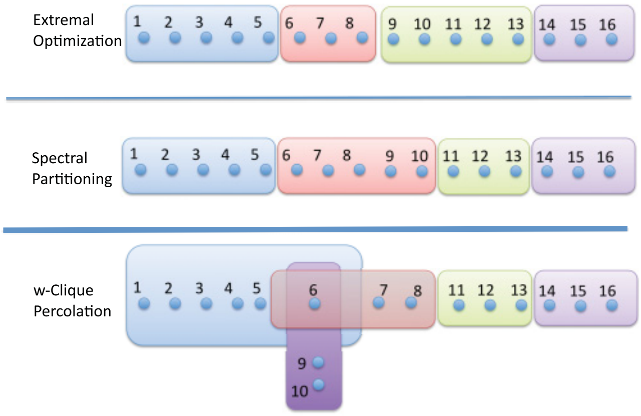


FIG. 3. (Color online) Node allocations to communities for three different community detection methodologies.

B. Benchmark graphs with community structure

Having soft-membership distributions not only allows us to describe our confidence in assigning node i to community k, but also to quantify the degree of "fuzziness" in the network. In Fig. 5(b), nodes such as $i = 6$ that lie on community boundaries have a membership distribution that is closer to uniform. We hence expect our method to indicate networks with little degree of modular organization. We apply the NMF method to realizations of the very popular Newman-Girvan (NG) random graph [31]. This benchmark tests the module identification capabilities of a method against an artificial graph of $N = 128$ nodes, observed solution of $C = 4$ communities (with $n = 32$ nodes each), average degree of

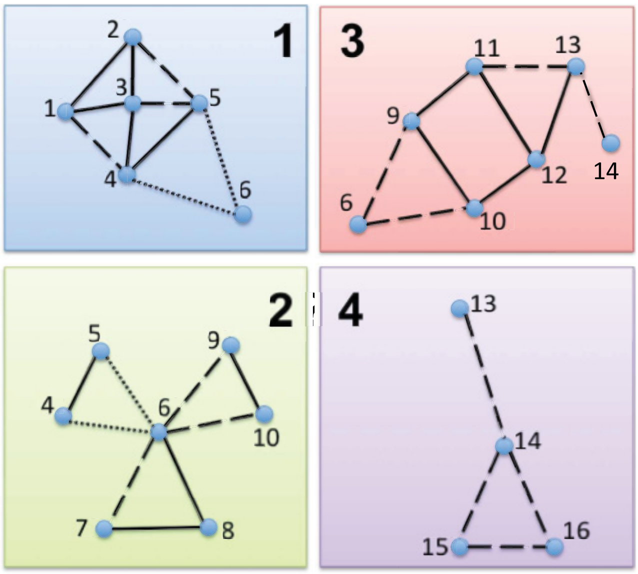
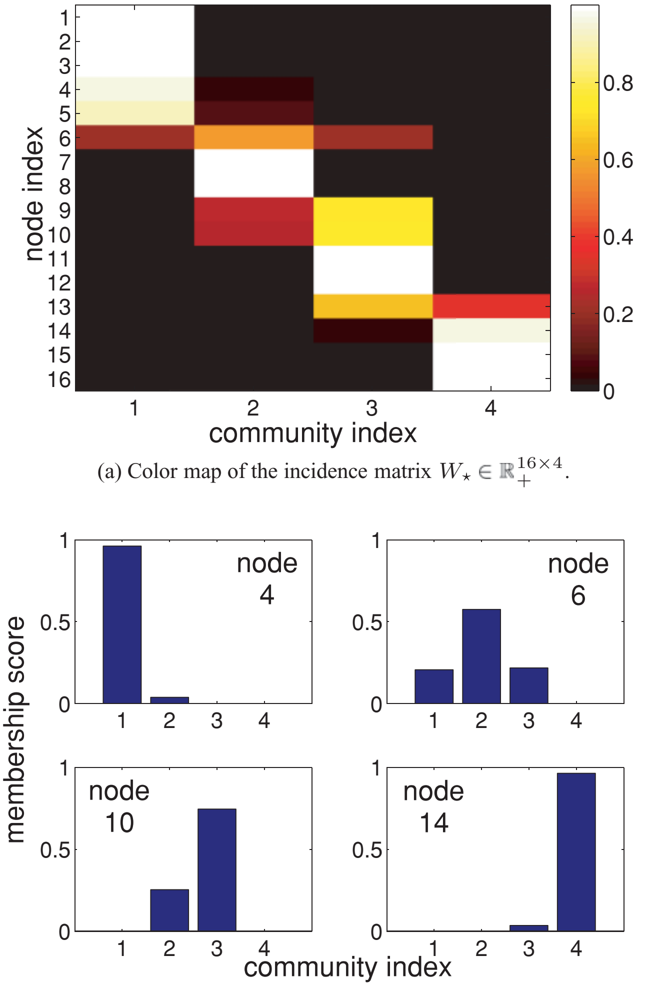


FIG. 4. (Color online) Our toy graph decomposed into $K_\star = 4$ overlapping communities using NMF.



(a) Color map of the incidence matrix $W_\star \in \mathbb{R}_+^{16\times 4}$.

(b) Soft membership distributions for various nodes in our toy network.

FIG. 5. (Color online) Panel (a) shows the node allocations proposed by our algorithm. Colors (gray shades) close to white indicate strong participation of node i (vertical axis) to community k (horizontal axis). Panel (b) shows example (normalized) rows of $\mathbf{W}_\star$ that correspond to the membership distribution of different nodes.

$\langle k\rangle = 16$, and a variable *intercommunity* degree $\langle k_{\text{out}}\rangle$ that controls the module cohesiveness of the network.

In Fig. 6(a) we plot our module identification performance based on the normalized mutual information (NMI) criterion [32], a real number between 0 and 1 which is maximal when the detected communities exactly meet expectations. In Fig. 6(b) we monitor our allocation confidence based on the mean entropy (in bits) $H = -\sum_{k=1}^{K} w_{ik}\log_2 w_{ik}$ of each node membership distribution. We can see that as we make the network fuzzier by increasing $\langle k_{\text{out}}\rangle$, our method "responds" by increasing the degree of node participation to multiple communities. An attractive aspect of this test is that the increase in entropy [see Fig. 6(b)] does not affect the module identification performance [we see from Fig. 6(a) that NMI remains close to unity] and is stable for the vast majority of $\langle k_{\text{out}}\rangle$ values. For comparison, we also provide in Fig. 6(a) the NMI performance of some popular hard-partitioning methods: extremal optimization [9], spectral partitioning [29], and hierarchical clustering [2]. For hierarchical clustering, angular distance acted as node similarity and complete-linkage clustering acted as group similarity; this combination has been empirically found to be optimal [2].

We extend the above test to the case of Lancichinetti-Fortunato random graphs (LFR) [33], which reflect more accurately the properties of real-world networks. In this setting, community cohesion is controlled by *mixing parameters* μ_k and μ_w, which denote the expected fraction of intercommunity degrees and weights per node. Other configuration parameters include the total number of nodes N, the average degree $\langle k\rangle$, the exponent of the degree distribution γ_1, and the exponent of the community-size distribution γ_2. We tested our method for a (decaying) range of values for μ_k,μ_w (where we set $\mu_k = \mu_w$), in weighted graphs of $N = 1000$ nodes and various values of $\langle k\rangle$, as seen in Fig. 7(a). In the same spirit as the NG graph case, in Fig. 7(b) we monitor the mean entropy of membership distributions per node (in bits) to quantify the confidence of our node allocations to communities. In Fig. 7(a) we can see that our model has an excellent module identification performance and starts to fail only when the mixing coefficients μ have values greater than 0.5, denoting no community organization in the graph. On the other hand, the increasing fuzziness of the network (based on μ) is captured in the mean entropy of the membership distributions; as the community structure is less cohesive, we are less confident in the allocation of nodes to groups.

TABLE I. Real world datasets.

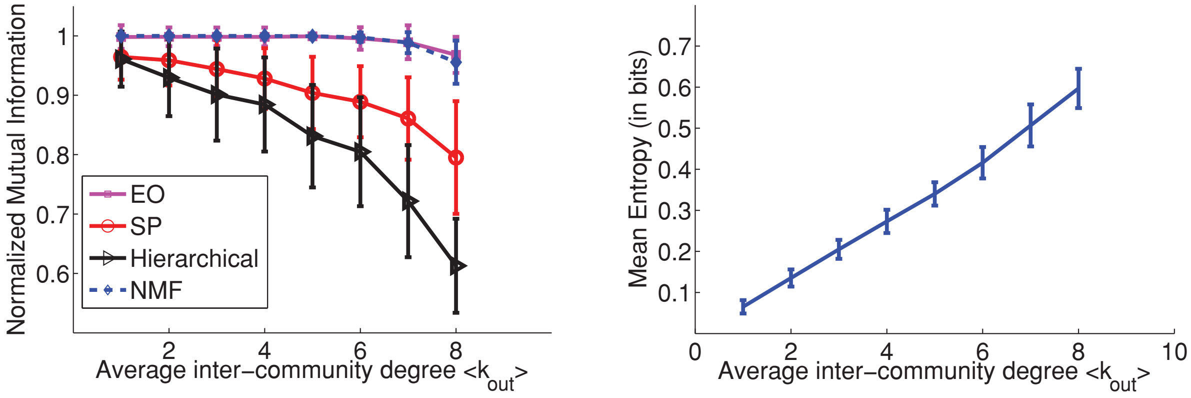
Dataset	N	M
Dolphins [34]	62	159
Books US politics [35]	105	441
Les Misérables [36]	77	254
College football [31]	115	613
Jazz musicians [37]	198	2742
C. elegans metabolic [9]	453	2025
Network science [5]	1589	2742
Facebook Caltech [38]	769	16656

C. Real-world datasets

We present the performance of our community detection method on a variety of popular benchmark datasets (see Table I) and compare it against the extremal optimization (EO) [9] and Louvain [6] methods. In contrast to the artificial graphs we used above, the absence of an "observed solution" for

TABLE II. Modularity results for NMF, EO, and Louvain methods.

Dataset	NMF	EO	Louvain
Dolphins	0.47 ± 0.03	0.51 ± 0.01	0.52
Books US politics	0.52 ± ϵ	0.48 ± 0.01	0.50
Les Misérables	0.53 ± 0.02	0.53 ± 0.01	0.57
College football	0.60 ± ϵ	0.58 ± 0.01	0.60
Jazz musicians	0.43 ± 0.01	0.42 ± 0.01	0.44
C. elegans metabolic	0.36 ± 0.01	0.40 ± 0.09	0.43
Network science	0.83 ± 0.01	0.86 ± 0.01	0.95
Facebook Caltech	0.38 ± 0.01	0.37 ± 0.01	0.37



(a) Normalized Mutual Information, value range 0–1.

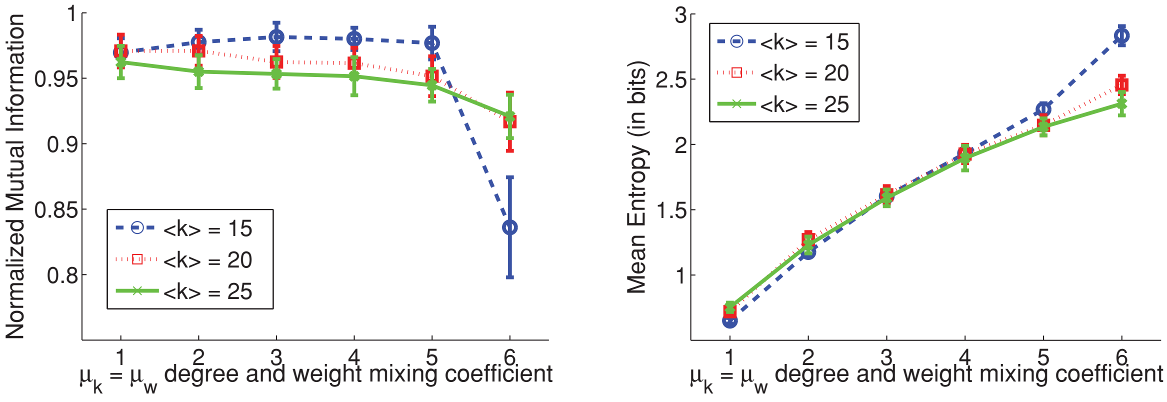
(b) Mean entropy of membership distribution.

FIG. 6. (Color online) Panel (a) compares the NMF (dashed ◇ line at the top) approach against extremal optimization (EO) (pale □ line at the top), spectral partitioning (SP) (○ line), and hierarchical clustering (hierarchical) (▹ line) in identifying the communities of Newman-Girvan artificial graphs. Each point is the mean of 100 graph realizations. Panel (b) shows the increase in uncertainty in assigning nodes to communities, as we increase the fuzziness of modular organization in NG graphs. Each point is the mean of 100 graph realizations.

these problems prevents us from using the normalized mutual information criterion for performance evaluation. Instead we use the popular *modularity* Q [5], which is a measure of how "statistically surprising" the intracommunity link density is for a proposed network partition. For the purposes of the experiment, we remove the overlapping aspect of the NMF solutions by assigning a node to a single community; the one for which it has the maximum degree of membership. Although this "greedy allocation" scheme omits the wealth of information provided by our model solutions, it is necessary in order to perform modularity comparisons against hard-partitioning methods. Comparison with clique percolation is also absent, as it provides a uniform participation score of nodes to modules, thus no greedy allocation can by applied. For each dataset, we ran the three methods 100 times, recording the values of *modularity* Q along with the number of extracted communities $K_\star$. The values are reported in Tables II and III; because the Louvain method demonstrated stable behavior across different runs, its standard deviations have been omitted. For NMF initialization we used $K_0 = N$ with hyperparameters $a = 5$ and $b = 2$, giving a vague prior. We note that the results are not very sensitive to changes in these values.

From Table II we can see that our approach performs competitively despite not being designed with the aim of maximizing modularity, unlike EO and the Louvain method. Additionally, it has the advantage of providing *soft-partitioning* solutions and node *membership scores* to each community. Finally, although our method favors sparse solutions, it does not suffer from the resolution limit [19] of modularity optimization methods such as EO, where smaller groups are merged together [3,19], leading to a smaller number of communities, as seen in Table III.

Figure 8 illustrates the first network in Table III , in which vertices are situated according to the Kamada-Kawai free-energy technique in Pajek software [39]. The hard partitioning of the Louvain method can be contrasted with the soft partitioning of an example run of the NMF method, in which vertices near the boundary of two or more communities are



(a) Normalized Mutual Information, value range 0–1.

(b) Mean entropy of membership distribution.

FIG. 7. (Color online) Results of the NMF method on realizations of the LFR random graphs for $N = 1000$ and different values for the average degree $\langle k \rangle$ and community cohesion μ parameters. Each point represents the mean and standard deviation over 100 graph realizations.

TABLE III. Number of communities from the NMF, EO, and Louvain methods.

Dataset	NMF	EO	Louvain
Dolphins	6.67 ± 0.83	4 ± 0	5
Books US politics	6.23 ± 0.62	4.04 ± 0.4	3
Les Misérables	9.97 ± 0.78	4.96 ± 1.72	6
College football	8.86 ± 0.79	8 ± 0	10
Jazz musicians	8.57 ± 8.89	4 ± 0	4
C. elegans metabolic	15.69 ± 1.14	7.96 ± 1.06	10
Network science	342.53 ± 5.28	58.24 ± 12.36	418
Facebook Caltech	24.28 ± 1.72	6.84 ± 1.82	10

represented by pie charts in a manner similar to that used by Ball *et al.* [40]. With the aid of the aforementioned greedy allocation scheme, the NMF community assignments agree with the Louvain community assignments for 55 of the 62 nodes. Of the seven mismatches, six correspond to the putative additional community (here colored dark green, in the dense central portion of the figure) postulated by the Louvain method; NMF replaces this tiny community with soft partitioning among the other communities. The seventh mismatch occurred for a node connected to two nodes of the eight-node community (upper portion of the graph) and two nodes of the eighteen-node community (right portion of the graph); the Louvain method allocated it to the eight-node community whereas NMF allocated it to both communities in the approximate proportion of 49:51, respectively.

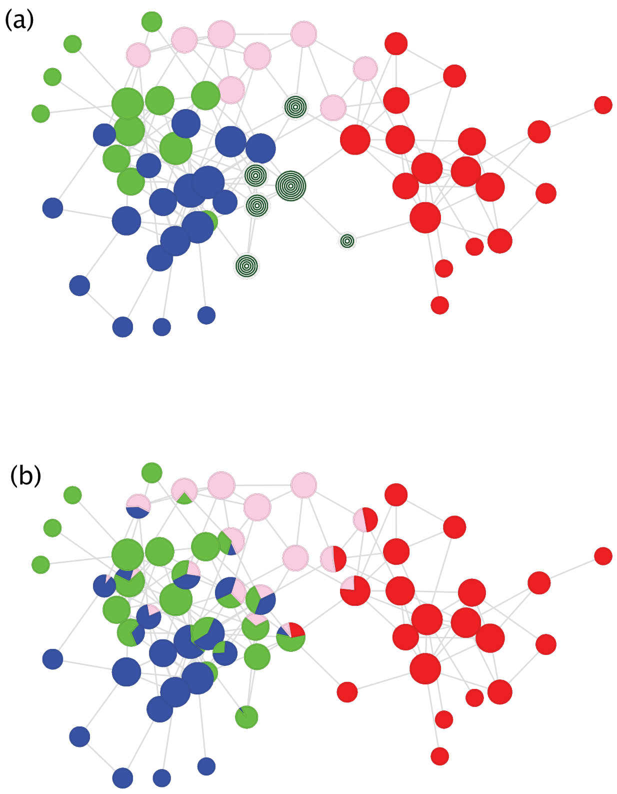


FIG. 8. (Color online) The Dolphins network [34], with (a) hard partitioning as per the Louvain method and (b) soft partitioning as per the NMF method. Node size increases nonlinearly with vertex degree, and soft partitions are shown as pie charts.

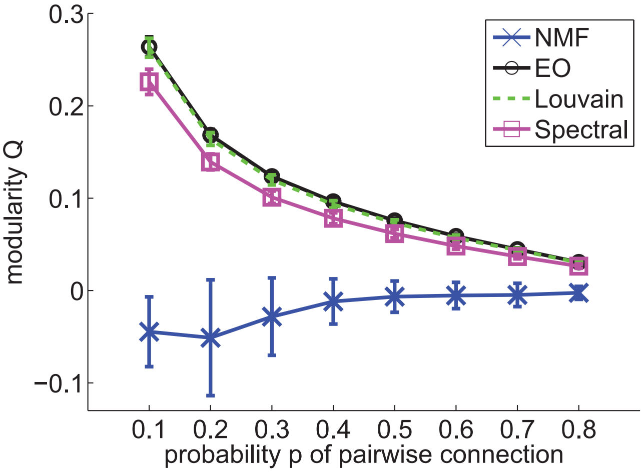


FIG. 9. (Color online) Modularity of network partitions of four community detection algorithms, ran on realizations of an ER graph family $\mathbf{G}(100,p)$. Each point represents the mean and standard deviation of modularity over 100 instances of $\mathbf{G}(100,p)$.

D. Graphs without community structure

We present the behavior of NMF in cases in which there is no community structure in the network, specifically focusing on the popular Erdős-Rényi (ER) random graphs. In such graphs, each link exists with a probability p which is common for any pair of nodes in the graph. Additionally, the probability of link formation at a given pair of nodes is independent of the presence of other links. This eliminates the tendency to form closed triangles and cliques that characterize real-world networks.

Therefore given various realizations of an ER graph family $\mathbf{G}(N,p)$ (N number of nodes and p probability of pair connection), we want our method to be able to capture such absence of mesoscopic organization, instead of declaring community structure when there is none. In Fig. 9 we compare NMF against three modularity-based methods: extremal optimization (EO), the Louvain method, and spectral partitioning, based on the Q value of their extracted network partitions, in realizations of an ER graph class $\mathbf{G}(100,p)$. We control the "network load" (number of links in the graph) by changing the value of p. For each value of p we generate 100 graphs, run community detection with each algorithm, and record the modularity values. The generated ER graphs we used have no disconnected components.

In Fig. 9 we can see that EO (black ○ line), Louvain (light dashed line), and SP (□line) produce significantly higher modularity values than NMF (bottom ×line), especially for sparse realizations of the Erdős-Rényi random graph, denoting the presence of modular organization. However, those high Q values do not correspond to any community structure, as Erdős-Rényi random graphs do not possess it by design. On the other hand, NMF has a more stable behavior as all modularity values are close to zero, indicating that nodes have no "preference" of being connected with members of the same group or otherwise. Especially for the case of sparse graphs with $p \simeq 0.1$, EO and Louvain achieve higher modularity values; in particular, they are very close to $Q = 0.3$, a threshold above which Newman and Girvan consider community structure to be present [5]. This overestimation of modular organization can be very misleading, especially when studying real-world networks which are usually sparse [41] due to their power-law degree distribution. Therefore if certain modularity optimization methods produce higher Q values than NMF, it might not mean necessarily that they have found a node configuration that denotes better community structure.

IV. IMPLEMENTATION DETAILS AND COMPLEXITY

As discussed in Sec. II C, parameter inference is performed by a series of update equations for the latent variables in the model. The computational load is governed chiefly by the matrix multiplication $\mathbf{WH}$ appearing in the denominator of the element-by-element division $\frac{\mathbf{V}}{\mathbf{WH}}$ in steps 3 and 4 of the algorithm above, which is of order $\mathcal{O}(N^2K)$. In practice, such cost can be significantly reduced if we exploit the sparse nature of adjacency matrices [42]: the dot products $\sum_k w_{ik}h_{kj}$ within $\mathbf{WH}$ need not be calculated when $v_{ij} = 0$, thus reducing significantly the effect of the quadratic term N^2 in our theoretical complexity expression. For the case of undirected networks, in which $\mathbf{V} = \mathbf{V}^{\mathrm{T}}$, the dot product operations are halved because $\mathbf{WH}$ is symmetric, and halved again because step 4 of the algorithm is redundant ($\mathbf{W} = \mathbf{H}^{\mathrm{T}}$).

Holistic community detection methods such as NMF, which operate upon the full adjacency matrix $\mathbf{V}$, can be memory inefficient when implemented naively. The quadratic complexity, $\mathcal{O}(N^2)$, can be mitigated by loading into memory only certain columns/rows of $\mathbf{V}$ when needed, as no holistic operations (such as inversion or multiplication) are required by the algorithm for $\mathbf{V}$ or $\hat{\mathbf{V}}$. In addition, all element-by-element division and multiplication operations should be parallelized, as there are no data dependencies among the threads.

V. DISCUSSION AND FUTURE WORK

In the present work we described a probabilistic approach to community detection that adopts a Bayesian non-negative matrix factorization model to achieve soft partitioning of a network in a computationally efficient manner. We have demonstrated how community detection can be seen as a generative model in a probabilistic framework in which priors exist over the model parameters. This enables model order selection, which in our framework is the number of latent communities (or classes of nodes) in the data. We also showed that the degree of participation of two individuals in various communities is a latent generator of the expected number of interactions between them.

Following the model formulation section, we demonstrated how NMF not only captures the membership of a node in multiple communities, but also quantifies how strongly that individual participates in each of the groups. By using the entropy of the node membership distribution, we can identify core nodes in each community or, inversely, broker nodes that act as mediators between different groups. At a global level, the mean entropy of the membership distributions can help us quantify the degree of fuzziness in the network, or the clarity of community structure. Network visualization tools can also be improved in this manner, as the degree of membership over different communities can be utilized to position an individual in a cloud of nodes.

We also showed that NMF has a competitive performance against popular community detection methods, on various popular network datasets. Although NMF is not a method aiming to maximize modularity Q, it competes well with methods that directly maximize modularity and we have showed that it can even outperform these methods in several module identification problems, while at the same time having the advantage of providing soft-partitioning solutions.

This work addresses the issue of extracting community partitions from a single interaction network defined by $\mathbf{V}$. We acknowledge that in many problems, this matrix describes only a snapshot $\mathbf{V}^{(t)}$ of a time-evolving, dynamic complex system. Therefore we seek to extend our community detection method to allow for a time-evolving solution space. At present we are approaching this via a jump-diffusion model (based around a Markov model), in which rate parameters are allowed to evolve with time and the structure of the community solutions may also have abrupt changepoints [43]. Our aim is to evaluate this approach in time-evolving systems in order to model community drifts and the transitions from one community structure to another.

Our current method produces point estimates for the model parameters via a maximum *a posteriori* (MAP) scheme. A fully Bayesian treatment can be employed via reversible jump Markov-chain Monte Carlo as presented in [44,45], or via the use of variational Bayes as derived in [45]. The advantage of a posterior distribution over quantities such as the inner rank dimensionality K is that we can see at which resolutions modular organization is most prevalent.

We also acknowledge that NMF, along with the majority of community-detection methods, assumes a fully observed adjacency matrix. This is not the case in many real-world applications in which data-collection limitations arise; for example when the system under study is sampled or when sensors fail to record every observation. However, NMF can be easily extended to allow for missing data [45].

Finally, in this paper we considered cases of undirected networks with symmetric interaction matrices $\mathbf{V}$. Although NMF does not allow the presence of negative links in the graph, it is still possible to consider the popular cases of asymmetric communication rates that arise in systems such as email or telephone networks.

ACKNOWLEDGMENTS

The authors would like to thank Nick Jones, Mason Porter, and Ron Daniel for valuable comments. Ioannis Psorakis is funded from a grant via Microsoft Research, for which we are most grateful. We gratefully acknowledge funding from the UK Research Council for project “Orchid,” Grant No. EP/I011587/1.

[1] M. E. J. Newman, *Networks: An Introduction* (Oxford University Press, New York, 2010).

[2] S. Fortunato, Phys. Rep. **486**, 75 (2010).

[3] M. A. Porter, J. P. Onnela, and P. J. Mucha, Not. Am. Math. Soc. **56**, 1082 (2009).

[4] J. Reichardt and S. Bornholdt, Physica D **224**, 20 (2006).

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

[6] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, J. Stat. Mech.: Theor. Exp. (2008) P10008.

[7] J. Reichardt and S. Bornholdt, Phys. Rev. Lett. **93**, 218701 (2004).

[8] M. Rosvall and C. T. Bergstrom, Proc. Natl. Acad. Sci. USA **104**, 7327 (2007).

[9] J. Duch and A. Arenas, Phys. Rev. E **72**, 027104 (2005).

[10] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, Nat. Lett. **435**, 814 (2005).

[11] A. Lancichinetti, S. Fortunato, and J. Kertsz, New J. Phys. **11**, 033015 (2009).

[12] T. S. Evans and R. Lambiotte, Phys. Rev. E **80**, 016105 (2009).

[13] Y. Y. Ahn, J. P. Bagrow, and S. Lehmann, Nature (London) **466**, 761 (2010).

[14] T. Nepusz, A. Petróczi, L. Négyessy, and F. Bazsó, Phys. Rev. E **77**, 016107 (2008).

[15] V. Tan and C. Févotte, in *SPARS09 - Signal Processing with Adaptive Sparse Structured Representations*, edited by Rémi Gribonval (Inria Rennes-Bretagne Atlantique, Saint Malo, Royaume-Uni, 2009), pp. 1–19.

[16] R.-S. Wang, S. Zhang, Y. Wang, X.-S. Zhang, and L. Chen, Neurocomputing **72**, 134 (2008).

[17] S. Zhang, R.-S. Wang, and X.-S. Zhang, Phys. Rev. E **76**, 046103 (2007).

[18] D. J. C. MacKay, Network: Comput. Neural Syst. **6**, 469 (1995).

[19] S. Fortunato and M. Barthelemy, Proc. Natl. Acad. Sci. USA **104**, 36 (2007).

[20] M. E. Tipping and C. M. Bishop, J. R. Stat. Soc. Ser. B **61**, 611 (1999).

[21] R. A. Choudrey and S. J. Roberts, Neural Comput. **15**, 213 (2003).

[22] S. J. Roberts and R. A. Choudrey, in *Deterministic and Statistical Methods in Machine Learning*, Lecture Notes in Computer Science Vol. 3635, edited by J. Winkler, M. Niranjan, and N. Lawrence (Springer, Berlin/Heidelberg, 2005), pp. 159–179.

[23] S. J. Roberts and R. A. Choudrey, Pattern Recognit. **36**, 1813 (2003).

[24] W. Penny and S. J. Roberts, IEE Proc. Vision, Image Signal Proc. **149**, 33 (2002).

[25] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory* (Wiley, Chichester, New York, 1994).

[26] D. D. Lee and H. S. Seung, Nature (London) **401**, 788 (1999).

[27] D. D. Lee and H. S. Seung, in *NIPS* (MIT, Cambridge, MA, 2000), pp. 556–562.

[28] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, Comput. Stat. Data Anal. **52**, 155 (2007).

[29] M. E. J. Newman, Proc. Natl. Acad. Sci. USA **103**, 8577 (2006).

[30] I. Farkas, D. Abel, G. Palla, and T. Vicsek, New J. Phys. **9**, 180 (2007).

[31] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. USA **99**, 7821 (2002).

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

[33] A. Lancichinetti and S. Fortunato, Phys. Rev. E **80**, 016118 (2009).

[34] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, Behav. Ecol Sociobiol. **54**, 396 (2003).

[35] V. Krebs, [http://www.orgnet.com/].

[36] D. Knuth, *The Stanford GraphBase: A Platform for Combinatorial Computing* (ACM Press, New York, 1993).

[37] P. Gleiser and L. Danon, Adv. Complex Syst. **6**, 565 (2003).

[38] A. L. Traud, E. D. Kelsic, P. J. Mucha, and M. A. Porter, e-print arXiv:0809.0960.

[39] V. Batagelj and A. Mrvar, Connections **21**, 47 (1998).

[40] B. Ball, B. Karrer, and M. Newman, e-print arXiv:1104.3590v1.

[41] C. Faloutsos, K. S. McCurley, and A. Tomkins, in *Proceeding of SIAM International Conference on Data Mining, Workshop on Link Analysis, Counter-terrorism and Privacy* (Lake Buena, Vista Florida, 2004).

[42] A. Clauset, M. E. J. Newman, and C. Moore, Phys. Rev. E **70**, 066111 (2004).

[43] R. Garnett, M. Osborne, S. Reece, A. Rogers, and S. Roberts, Comput. J. **53**, 1430 (2010).

[44] M. Zhong and M. Girolami, in *Twelfth International Conference on Artificial Intelligence and Statistics* (MIT Press, Clearwater, Florida 2009), p. 8.

[45] A. T. Cemgil, Intell. Neuroscience **2009**, 4 (2009).