Community detection, link prediction, and layer interdependence in multilayer networks

Caterina De Bacco,[1] Eleanor A. Power,[1] Daniel B. Larremore,[1] and Cristopher Moore[1]

[1]*Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, USA*[*]

Complex systems are often characterized by distinct types of interactions between the same entities. These can be described as a multilayer network where each layer represents one type of interaction. These layers may be interdependent in complicated ways, revealing different kinds of structure in the network. In this work we present a generative model, and an efficient expectation-maximization algorithm, which allows us to perform inference tasks such as community detection and link prediction in this setting. Our model assumes overlapping communities that are common between the layers, while allowing these communities to affect each layer in a different way, including arbitrary mixtures of assortative, disassortative, or directed structure. It also gives us a mathematically principled way to define the interdependence between layers, by measuring how much information about one layer helps us predict links in another layer. In particular, this allows us to bundle layers together to compress redundant information, and identify small groups of layers which suffice to predict the remaining layers accurately. We illustrate these findings by analyzing synthetic data and two real multilayer networks, one representing social support relationships among villagers in South India and the other representing shared genetic substrings material between genes of the malaria parasite.

I. INTRODUCTION

Networks that describe real-world relationships often have different types of links that connect their nodes. For example, links in a social network may represent friendships, marriages, or collaborations, while links in a transportation network may represent planes, trains, or automobiles. Such networks are called multitype or multilayer networks since each type of link can be separated into its own layer, thereby connecting the same set of nodes in multiple ways. However, understanding the large-scale structure of multilayer networks is made difficult by the fact that the patterns of one type of link may be similar to, uncorrelated with, or different from the patterns of another type of link. These differences from layer to layer may exist at the level of individual links, connectivity patterns among groups of nodes, or even the hidden groups themselves to which each node belongs. Therefore, finding community structure in multilayer networks requires simultaneously considering three related problems: (i) the *multilayer community detection problem*, in which we seek a description of the network that divides the nodes according to groups hidden in the link patterns of multiple layers; (ii) the *layer interdependence problem*, in which we seek a description of the relationships between the layers containing different types of links; and (iii) the *link prediction problem*, in which we seek to accurately predict missing link data by making use of all relevant layers of the network.

These three problems are fundamentally intertwined. Multilayer community detection requires knowing which layers have related structure and which layers are unrelated, since redundant information across layers may pro-

vide stronger evidence for clear communities than each layer would on its own. However, measuring layer interdependence requires a working definition of interdependence and a method to measure it. The performance of a model or algorithm on the link prediction task provides exactly such a measure: specifically, whether it is possible to integrate information across layers, using links of one type to help predict those of another type. Thus, in settings where data are well-represented as a multilayer network, as is relevant in inferring genetic and protein-protein interactions in cells [1], characterizing interdependent infrastructures [2] and understanding the impact of different social ties in human relationships [3, 4], we desire a model capable of recognizing interdependencies between layers, integrating and merging information between them, and using this information both to classify nodes and to predict links.

In this paper we propose an approach that performs all three of these tasks. We define a generative model for multilayer community detection that is applicable to both directed and undirected networks, as well as networks with integer-weighted links. We provide a highly-scalable expectation-maximization algorithm that fits this model to data, taking as its input a multilayer network and yielding a "mixed-membership" partition: nodes are divided into communities or groups, but each node may belong to some extent to multiple groups. While this partition is shared by all the layers, the model allows for different connectivity patterns in each layer, including arbitrary mixtures of assortative, disassortative, and directed structure.

In addition to classifying nodes, our model also makes link prediction possible: given an incomplete dataset where not all links are known, it assigns probabilities to each pair of nodes that they have an unobserved link of each type. Finally, by sequentially fitting the model to single layers and multiple layers, we show how to deter-

mine whether the information provided by additional layers improves link prediction performance, thereby quantifying layer interdependence. We show how to use this method to identify which layers of the network are redundant, and which provide independent information. For instance, we can identify small sets of layers which together capture most of the information about the network. This may be useful in contexts where gathering layers requires an investment of limited resources in the laboratory or the field—for instance, if a social scientist can ask their subjects a limited number of questions, identifying a limited number of types of social relationships.

Community detection is a fundamental element of network science, yet most community detection algorithms have been developed for single-layer networks. We can use these methods to analyze multilayer networks, either by aggregating all layers to create a single-layer network [5, 6] or by analyzing each layer independently. Multilayer-specific methods that maximize community quality functions such as modularity [7] provide non-overlapping partitions, but inherit the issues of their single-layer counterparts, namely a dependence on proper choice of a null model [8, 9]. Moreover, these methods do not typically provide a framework to perform link prediction, since without additional assumptions they do not assign probabilities to the presence or absence of a link.

More recent methods for multilayer community detection are based on fitting various generative models via Bayesian inference or maximum likelihood estimation [10–15]. Our algorithm falls into this category, but differs from most of these models by not assuming *a priori* any specific network structure. Many network models assume *assortative* or *homophilic* community structure, meaning that nodes are more likely to be connected to others in the same community. This assumption is often incorrect in food webs, technological networks, and social networks where links consist of nominations, for instance, of a trustworthy or powerful person that they might ask for advice, help or work. Our model avoids this assumption: while it looks for an (overlapping) community structure that is consistent with every layer, it deals happily with networks where some layers are assortative, others are disassortative, yet others have core-periphery or directed structure, and so on. In the case of directed links, it also recognizes that nodes might play different roles, and thus effectively belong to different groups, when forming incoming or outgoing links.

By measuring the extent to which one layer helps us predict links in another layer, our method also gives a mathematically principled way to measure the relationships between the layers of a multilayer network, including identifying layers which are redundant with others or highly independent from them. This problem arises even in naive algorithms that aggregate layers into a single-layer network. For instance, the multilayer versions of eigenvector centrality or modularity [7, 16] use weighted averages over layers, requiring them to infer or choose a weight for each layer. A number of recent works have taken a compression approach, aggregating layers with similar structure [17, 18]; in particular, Ref. [18] uses a generative model that jointly assigns community memberships to nodes and groups of layers, which they call *strata*. This is similar in spirit to our approach, but our model handles overlapping communities, directed networks, and weighted networks in a unified way.

In Section II we describe our model and in Section III give an efficient algorithm that fits its parameters to network data. In Section IV, we provide performance results on synthetic benchmarks and compare with other algorithms, and in Section V we discuss how to perform link prediction and measure layer interdependencies. We then apply these concepts to two real world networks, drawn from anthropology and biology, and discuss the results in Section VI before concluding.

II. THE MULTILAYER MIXED-MEMBERSHIP STOCHASTIC BLOCK MODEL

In this section we describe our model and fix our notation. The network consists of N nodes and has L layers. Each layer has an adjacency matrix $A^{(\alpha)}$, where $A_{ij}^{(\alpha)}$ is the number of edges from i to j of type α; alternatively, we can think of A as an $N \times N \times L$ tensor. Our model generates these networks probabilistically, assuming an underlying structure consisting of K overlapping groups.

Each node belongs to each group to an extent described by a K-dimensional vector. Since we are interested in directed networks, we give each node i two membership vectors, u_i and v_i, which determine how i forms outgoing and incoming links respectively. (When modeling undirected networks, we set $u = v$.) Each layer α has a $K \times K$ affinity matrix $w^{(\alpha)}$ describing the density of edges between each pair of groups. The expected number of edges in layer α from i to j is then given by a bilinear form,

$$M_{ij}^{(\alpha)} = \sum_{k,\ell=1}^{K} u_{ik}\, v_{j\ell}\, w_{k\ell}^{(\alpha)} \,. \tag{1}$$

Finally, for each i, j and α, we choose $A_{ij}^{(\alpha)}$ independently from the Poisson distribution with mean $M_{ij}^{(\alpha)}$.

Note that while we assume that the membership vectors have nonnegative entries, we do not normalize them. This allows us to account easily for heterogeneous degree distributions, since multiplying u_i or v_i by a constant increases the expected out- or in-degree without changing the distribution of neighbors to which a given one of i's edges connects.

Note also that while this model supposes that nodes have the same group membership in all layers, it allows the structure of each layer to vary arbitrarily with respect to these groups. For instance, some layers could be assortative and others disassortative, with affinity matrices $w^{(\alpha)}$ which are large on or off the diagonal; other

layers could have strongly directed structure, with asymmetric $w^{(\alpha)}$, or core-periphery structure, where $w^{(\alpha)}$ has one large entry on the diagonal.

For a single layer, our model is similar to existing mixed-membership block models [19–25]. Some of these use mixtures of Bernoulli random variables; we follow [22] in using the Poisson distribution since it leads to a tractable and efficient expectation-maximization algorithm. The Poisson distribution also allows us to model multigraphs or integer-weighted networks. However, in our applications here we will focus on the sparse case where $M_{ij}^{(\alpha)}$ is small, and assume for simplicity that $A_{ij}^{(\alpha)}$ is 0 or 1.

Our model also bears a close mathematical relationship to topic models [26, 27], which generate bipartite weighted graphs of documents and words based on their relevance to mixtures of topics. More generally, it can be viewed as a variant of non-negative tensor factorization (see e.g. [28] for a review) and in particular of Poisson tensor factorization [12, 13, 29–34]. However, the affinity matrices $w^{(\alpha)}$, which allow different layers to be assortative or disassortative, correspond to a kind of Tucker decomposition [35] of tensor rank K^2. This is more general than the PARAFAC/CANDECOMP decomposition [36, 37], which corresponds to the special case of our model where $w^{(\alpha)}$ is diagonal for each α. In the undirected case where $u = v$, PARAFAC thus assumes a purely assortative structure, where a link between two nodes can only exist if their membership vectors overlap.

Various kinds of Poisson Tucker decomposition for dynamic and multilayer networks have also been proposed very recently in the machine learning community, particularly in [12–14]. Indeed, our model is very nearly a special case of that of [14], which has additional parameters intended to model datasets with both multiple types of links and multiple time steps. The main difference between these works and our approach is that they impose priors on the "core tensor" parameters [analogous to $w_{k\ell}^{(\alpha)}$] and they use Monte Carlo sampling for Bayesian inference. In contrast, we find point estimates of these parameters using an expectation-maximization algorithm, detailed in the next section. Below we compare our results to the algorithm of [13], which is designed for the same kinds of datasets as ours.

III. THE EXPECTATION-MAXIMIZATION ALGORITHM

Given an observed multilayer network with adjacency tensor A, our goal is to simultaneously infer the nodes' membership vectors and the affinity matrices for each layer. In this section, we describe an efficient algorithm which does this by the method of maximum likelihood.

Let Θ be shorthand for all $2NK + K^2L$ model parameters, i.e., the u_{ik}, $v_{i\ell}$, and $w_{k\ell}^{(\alpha)}$. Assuming that all Θ are equally likely *a priori*, the probability of Θ given A is proportional to the probability of A given Θ. Using the Poisson distribution function gives

$$P(\Theta \mid A) \propto P(A \mid \Theta) = \prod_{i,j=1}^{N} \prod_{\alpha=1}^{L} \frac{\mathrm{e}^{-M_{ij}^{(\alpha)}}(M_{ij}^{(\alpha)})^{A_{ij}^{(\alpha)}}}{A_{ij}^{\alpha}!} \quad (2)$$

(One could also impose a prior $P(\Theta)$ and perform maximum *a posteriori* inference [13, 31, 32], but we have not done this here.) The log-likelihood is then

$$\mathcal{L}(\Theta) = \sum_{i,j,\alpha} \left[A_{ij}^{(\alpha)} \log \sum_{k\ell} u_{ik} v_{j\ell} w_{k\ell}^{(\alpha)} - \sum_{k\ell} u_{ik} v_{j\ell} w_{k\ell}^{(\alpha)} \right] \quad (3)$$

where we omit the terms $\log A_{ij}^{(\alpha)}!$ since they depend only on the data.

We wish to find the Θ that maximizes Eq. (3). This is computationally difficult, but we can make it more tractable with a classic variational approach. For each i, j, α with $A_{ij}^{(\alpha)} = 1$, consider a probability distribution $\rho_{ijk\ell}^{(\alpha)}$ over pairs of groups k, ℓ: this is our estimate of the probability that that edge exists due to i and j belonging to groups k and ℓ respectively. (If the network is a multigraph and i, j have multiple links of the same type, we give each one its own distribution ρ; below we assume for simplicity that this does not occur.) Jensen's inequality $\log \overline{x} \geq \overline{\log x}$ then gives

$$\log \sum_{k\ell} u_{ik} v_{j\ell} w_{k\ell}^{(\alpha)} = \log \sum_{k\ell} \rho_{ijk\ell}^{(\alpha)} \frac{u_{ik} v_{j\ell} w_{k\ell}^{(\alpha)}}{\rho_{ijk\ell}^{(\alpha)}}$$
$$\geq \sum_{k\ell} \rho_{ijk\ell}^{(\alpha)} \log \frac{u_{ik} v_{j\ell} w_{k\ell}^{(\alpha)}}{\rho_{ijk\ell}^{(\alpha)}}$$
$$= \sum_{k\ell} \rho_{ijk\ell}^{(\alpha)} \log u_{ik} v_{j\ell} w_{k\ell}^{(\alpha)} - \sum_{k\ell} \rho_{ijk\ell}^{(\alpha)} \log \rho_{ijk\ell}^{(\alpha)} . \quad (4)$$

Moreover, this holds with equality when

$$\rho_{ijk\ell}^{(\alpha)} = \frac{u_{ik} v_{j\ell} w_{k\ell}^{(\alpha)}}{\sum_{k'\ell'} u_{ik'} v_{j\ell'} w_{k'\ell'}^{(\alpha)}} . \quad (5)$$

Thus maximizing $\mathcal{L}(\Theta)$ is equivalent to maximizing

$$\mathcal{L}(\Theta, \rho) = \sum_{i,j,\alpha,k,\ell} \left[A_{ij}^{(\alpha)} \left(\rho_{ijk\ell}^{(\alpha)} \log u_{ik} v_{j\ell} w_{k\ell}^{(\alpha)} - \rho_{ijk\ell}^{(\alpha)} \log \rho_{ijk\ell}^{(\alpha)} \right) - u_{ik} v_{j\ell} w_{k\ell}^{(\alpha)} \right] \quad (6)$$

with respect to both Θ and ρ.

The expert reader will recognize that this variational argument is simply classical thermodynamics in disguise. Fix the parameters Θ and consider a spin system where each edge, i.e., each triple (i, j, α) with $A_{ij}^{(\alpha)} = 1$, has a state consisting of a pair of groups (k, ℓ). Define the

Hamiltonian as

$$H = - \sum_{(i,j,\alpha):A_{ij}^{(\alpha)}=1} \log u_{ik(i,j,\alpha)} v_{j\ell(i,j,\alpha)} w_{k(i,j,\alpha),\ell(i,j,\alpha)}^{(\alpha)}$$
$$- \sum_{i,j,\alpha,k',\ell'} u_{ik'} v_{j\ell'} w_{k'\ell'}^{(\alpha)}$$

(note that the second term is constant). Then the Boltzmann distribution is a product distribution of the distributions $\rho_{ij}^{(\alpha)}$ given by Eq. (5) on each edge. Moreover, $-\mathcal{L}(\Theta, \rho)$ is the free energy $E - TS$ where $T = 1$, and we recover the familiar fact that this is minimized by the Boltzmann distribution. In this context, finding the maximum-likelihood estimate of the parameters Θ corresponds to minimizing the free energy of this spin system.

We can maximize $\mathcal{L}(\Theta, \rho)$ by alternatively updating ρ and Θ. This general approach is called an expectation-maximization (EM) algorithm: the expectation step computes the marginals of the Boltzmann distribution for the current estimate of the parameters, and the maximization step finds the most-likely value of the parameters given those marginals. The fact that the Boltzmann distribution takes a simple product form makes the expectation step especially simple, making the algorithm highly efficient.

The update equations for Θ in the maximization step can be can be derived by computing the partial derivative of $\mathcal{L}(\Theta, \rho)$ with respect to the various parameters. For instance,

$$\frac{\partial \mathcal{L}(A, \rho)}{\partial u_{ik}} = \sum_{j,\ell,\alpha} \left[\frac{A_{ij}^{(\alpha)} \rho_{ijk\ell}^{(\alpha)}}{u_{ik}} - v_{j\ell} w_{k\ell}^{(\alpha)} \right] . \qquad (7)$$

Setting this to zero, and doing the same for the partial derivatives with respect to $v_{j\ell}$ and $w_{k\ell}^{(\alpha)}$, gives

$$u_{ik} = \frac{\sum_{j,\alpha} A_{ij}^{(\alpha)} \sum_\ell \rho_{ijk\ell}^{(\alpha)}}{\sum_\ell \left(\sum_j v_{j\ell} \right) \left(\sum_\alpha w_{k\ell}^{(\alpha)} \right)} \qquad (8)$$

$$v_{j\ell} = \frac{\sum_{i,\alpha} A_{ij}^{(\alpha)} \sum_k \rho_{ijk\ell}^{(\alpha)}}{\sum_k \left(\sum_i u_{ik} \right) \left(\sum_\alpha w_{k\ell}^{(\alpha)} \right)} \qquad (9)$$

$$w_{k\ell}^{(\alpha)} = \frac{\sum_{ij} A_{ij}^{(\alpha)} \rho_{ijk\ell}^{(\alpha)}}{\left(\sum_i u_{ik} \right) \left(\sum_j v_{j\ell} \right)} . \qquad (10)$$

The EM algorithm thus consists of randomly initializing the parameters Θ, and then repeatedly alternating between updating ρ using Eq. (5) and updating Θ using Eqs. (8)-(10) until it reaches a fixed point. This fixed point is a local maximum of $\mathcal{L}(\Theta, \rho)$, but it is not guaranteed to be the global maximum. Therefore, we perform multiple runs of the algorithm with different random initializations for Θ, taking the fixed point with the largest value of $\mathcal{L}(\Theta, \rho)$. The computational complexity per iteration scales as $O(MK^2)$ where M is the total number of edges summed over all layers and K is the number of groups. In practice we find that our algorithm converges within a fairly small number of iterations. Thus it is highly scalable, with a total running time roughly linear in the size of the dataset.

Once we converge to a fixed point, we can assign nodes to communities by normalizing the membership vectors to $\bar{u}_i = u_i / \sum_k u_{ik}$, so that for each i we have $\sum_k \bar{u}_{ik} = 1$. This approach was used in [22] as an method for classifying nodes in overlapping communities; however, since we allow u_i and v_i to be distinct, the "outgoing" and "incoming" assignments of a node might differ. These are *soft* assignments, meaning that nodes can belong to more than one community. If one wishes to obtain a *hard* assignment, one can assign each node to the single community corresponding to the maximum entry of u or v, but the overlapping character of the community structure is then lost.

We call our model and its associated algorithm MULTITENSOR. A numerical implementation is available for use under an open source license [38].

IV. RESULTS ON SYNTHETIC NETWORKS

We tested MULTITENSOR's ability to detect community structure synthetic networks using the multilayer benchmark proposed in Ref. [8]. This model is somewhat different from ours: rather than having mixed-membership vectors that are the same in every layer, they use hard partitions where each node belongs to a single group, but they allow these partitions to vary from layer to layer. The partitions in different layers are correlated by a so-called layer interdependence tensor. For simplicity, we use a one-parameter version of this tensor with dependency $p \in [0, 1]$: when $p = 0$ the partitions between layers are independent, and when $p = 1$ they are identical. Once these partitions are chosen, they generate edges in each layer according to a degree-corrected block model [39], with a user-specified degree distribution and affinity matrix.

We used this benchmark to generate synthetic networks with $N = 300$ nodes, $L = 4$ layers, and $K = 5$ communities. For each layer we used a truncated power-law degree distribution with exponent $\gamma = -3$, minimum degree $k_{min} = 3$ and maximum degree $k_{max} = 30$. We varied the affinity matrix of the block model according to a mixing parameter μ: if $\mu = 0$ all edges lie within communities, and if $\mu = 1$ edges are assigned regardless of the community structure.

We define our algorithm's accuracy on these benchmarks in terms of how well the inferred membership vectors match the ground-truth distribution of memberships across layers. That is, we hope that, after normalizing so that $\sum_k \bar{u}_{ik} = 1$, each u_{ik} is close to the fraction of layers in which node i belongs to group k, which we denote $\bar{u}_{ik}^0$. We quantify the similarity between these two distributions using two measures. The first is their inner

product, often called the *cosine similarity* (CS), averaged over all the nodes:

$$\text{CS} = \frac{1}{N} \sum_{i=1}^{N} \frac{\bar{u}_i^0 \cdot \bar{u}_i}{|\bar{u}_i^0||\bar{u}_i|}, \tag{11}$$

where in the denominator $|\bar{u}|$ denotes the Euclidean norm. Here $\text{CS} = 1$ corresponds to perfect accuracy. The second is the L_1 error between the two distributions, also known as their statistical distance or total variation distance, averaged over all the nodes:

$$L_1 = \frac{1}{2N} \sum_{i=1}^{N} \left\| \bar{u}_i^0 - \bar{u}_i \right\|_1 = \frac{1}{2N} \sum_{i=1}^{N} \sum_{k=1}^{K} \left| \bar{u}_{ik}^0 - \bar{u}_{ik} \right|. \tag{12}$$

The factor of $1/2$ is used so that this distance ranges from 0 for identical distributions to 1 for distributions with disjoint support. In both measures, we give ourselves the freedom to permute the groups, so that the inferred groups of our model correspond to the groups of the benchmark. Thus we maximize the cosine similarity CS, and minimize the L_1 error, over all $K!$ permutations of the K groups.

For comparison, we use two other algorithms that infer overlapping multilayer partitions. The first is the restricted diagonal version of our model, which only allows diagonal affinity matrices $w^{(\alpha)}$; as discussed above, this is equivalent to the Poisson version of PARAFAC tensor factorization [34] and in the undirected case $u = v$ this corresponds to assume an assortative network structure. The second algorithm is a fully Bayesian Poisson tensor factorization (BPTF) [13]. The main differences between these two models are the prior information and the optimization approach. The former considers a uniform prior and calculates point estimates of the parameters using an iterative algorithm similar to ours. The BPTF algorithm instead assumes Gamma-distributed parameters and updates the parameters of these distributions instead of the point estimates; in the end it uses the geometric mean of these distributions as its estimate of the parameters.

In Table I we report the best results in terms of cosine similarity and L_1 error obtained by three algorithms: MULTITENSOR, its diagonal special case (or Poisson PARAFAC), and BPTF. By varying the layer interdependence p and the mixing parameter μ, we range over cases where the community structure is relatively easy to infer to those where it is much harder. Specifically, inference is easier when p is large, so that the layers are strongly correlated, and μ is small, so that most links are within communities. For each pair (μ, p) we generated 50 independent benchmark networks, and for each network and each algorithm we performed 50 independent runs with independently random initial conditions, taking the fixed point with highest likelihood. We defined convergence numerically by testing whether $\mathcal{L}(\Theta, \rho)$ has not improved by more than 0.1 for 10 iterations.

In every case our algorithm achieves the highest cosine similarity, and in the majority of the cases the smallest L_1 error, indicated in boldface. All three algorithms

perform poorly in the hard regime where one of the two parameters introduces a high level of stochasticity in the network, i.e. when either $\mu = 0.5$ or $p = 0.5$. In the other cases our algorithm is significantly better according to both measures.

The benchmarks of [8] are assortative in every layer, and the diagonal and BPTF algorithms work fairly well. To illustrate the greater flexibility of our algorithm, we also generated synthetic networks with different kinds of structure in different layers.. Specifically, we generated layers whose structures are assortative ($w_{11} = w_{22} > w_{12} = w_{21}$), disassortative ($w_{11} = w_{22} < w_{12} = w_{21}$), core-periphery ($w_{11} > w_{12} = w_{21} > w_{22}$), and directed with a bias from the first group to the second one ($w_{12} > w_{11} = w_{22} > w_{21}$).

We considered three types of networks, all having $K = 2$ groups and $N = 300$ nodes, but with different numbers and kinds of layers (see Table II). In each one the groups are of equal size, with un-mixed group memberships (i.e., $u_i = v_i = (0, 1)$ or $(1, 0)$). The first type of network has $L = 2$ layers, one assortative and one disassortative; the second network has $L = 4$, with two assortative and two disassortative layers; the last one has $L = 4$, with one layer each with assortative, disassortative, core-periphery, and biased directed structure.

We generated 10 independent samples of each of these types of network and calculated the CS and the L_1 norm between the inferred membership and the ground truth using the maximum-likelihood fixed point over 10 runs of each algorithm with different random initial conditions. As shown in Table III, MULTITENSOR achieves significantly greater performance than the diagonal or BPTF algorithms in all three cases, due to its flexibility in modeling mixtures of these different types of structure.

V. LEARNING LAYER INTERDEPENDENCE VIA LINK PREDICTION

Despite the fact that a network may have multiple layers, there is no guarantee that the structure of one layer is related to the structure of any other. In fact, depending on the context, it may even be desirable that two layers are entirely uncorrelated, since then they reveal different kinds of information about the underlying structure. The *layer interdependence problem* consists of identifying which sets of layers are structurally related, and quantifying the strengths of those interrelationships.

We are not the first to provide a solution to the layer interdependence problem. One intuitive approach is to independently infer the community structure of each layer and then simply compute pairwise correlations between the community partitions of each layer [40]. While this approach is straightforward, it is unable to use the information in some layers to assist with inference of structure in other layers, since every layer is treated independently. Another approach has been to cluster the inferred affinity matrices, analogous to our $w^{(\alpha)}$, during the parameter in-

$\mu = 0.0$	$p = 0.5$				$p = 0.8$				$p = 0.9$			
	CS	σ	L_1	σ	CS	σ	L_1	σ	CS	σ	L_1	σ
MULTITENSOR	**0.66**	0.06	0.58	0.07	**0.93**	0.07	**0.19**	0.09	**0.99**	0.01	**0.07**	0.01
Diagonal	0.65	0.05	0.60	0.06	0.84	0.06	0.31	0.08	0.97	0.04	0.10	0.05
BPTF	**0.66**	0.06	**0.57**	0.06	0.89	0.07	0.24	0.08	0.96	0.03	0.10	0.04

$\mu = 0.1$	$p = 0.5$				$p = 0.8$				$p = 0.9$			
	CS	σ	L_1	σ	CS	σ	L_1	σ	CS	σ	L_1	σ
MULTITENSOR	**0.63**	0.05	0.62	0.05	**0.92**	0.06	**0.22**	0.08	**0.98**	0.01	**0.11**	0.01
Diagonal	**0.63**	0.04	0.62	0.04	0.87	0.06	0.29	0.07	0.94	0.07	0.17	0.08
BPTF	0.62	0.06	**0.61**	0.05	0.84	0.07	0.32	0.08	0.93	0.06	0.18	0.07

$\mu = 0.5$	$p = 0.5$				$p = 0.8$				$p = 0.9$			
	CS	σ	L_1	σ	CS	σ	L_1	σ	CS	σ	L_1	σ
MULTITENSOR	**0.55**	0.03	0.70	0.02	**0.55**	0.05	**0.67**	0.04	**0.59**	0.07	0.59	0.07
Diagonal	**0.55**	0.03	0.71	0.02	0.53	0.03	0.69	0.03	0.58	0.07	0.63	0.06
BPTF	0.52	0.03	**0.69**	0.03	0.49	0.05	0.68	0.04	**0.59**	0.07	**0.58**	0.06

TABLE I. Performance in detecting overlapping partitions on synthetic networks, using our MULTITENSOR algorithm, the diagonal special case which corresponds to the Poisson version of PARAFAC tensor decomposition [34], and Bayesian Poisson Tensor Factorization [13]. Benchmark networks were generated with the model of [8] with interdependence p and mixing parameter μ: the community structure is stronger when p is large and μ is small. For each (μ, p) pair we measure the cosine similarity (CS) and the average L_1 error between the planted and inferred structure averaged over 50 independently generated benchmark networks. For each network, we run each network 50 times with independently random initializations, and use the parameters given by the highest-likelihood fixed point. Good performance corresponds to high cosine similarly and low L_1 error, and the best performance for each pair of parameter values is indicated by boldface. The errors σ are the standard deviation over the 50 benchmarks.

G	K	L	$\langle E \rangle$	Structure
1	2	2	1980	{assort, disassort}
2	2	4	3960,2250	{assort, disassort, core-per, dir. disassort}
3	2	4	1980	{2 assort, 2 disassort}

TABLE II. Description of synthetic network structures. $\langle E \rangle$ is the average number of edges per layer, all networks have $N = 300$ nodes. For networks $G = 1, 3$ we used affinity matrices W^a and W^d for the assortative and disassortative layers respectively, with entries $w^a_{11} = w^a_{22} = w^d_{12} = w^d_{21} = 0.04$, $w^a_{12} = w^a_{21} = w^d_{11} = w^d_{22} = 0.004$ so that $\langle E \rangle = 1980$; for $G = 2$ the first two layers (1 assortative and 1 disassortative) have affinity matrices W^a and W^b with entries $w^a_{11} = w^a_{22} = w^b_{12} = w^b_{21} = 0.08$, $w^a_{12} = w^a_{21} = w^b_{11} = w^b_{22} = 0.008$ so that $\langle E \rangle = 3960$; the third and fourth (1 core-periphery and 1 directed disassortative) have affinity matrices W^c and W^d with entries $w^c_{11} = w^d_{12} = 0.08$, $w^c_{12} = w^c_{21} = w^d_{11} = w^d_{22} = 0.008$ and $w^c_{22} = w^d_{21} = 0.004$ so that $\langle E \rangle = 2250$.

ference and estimation procedures, thus gathering layers into "strata" [18].

However, while we also explore clustering the $w^{(\alpha)}$ below, neither of these methods captures the general kind of independence we are interested in. For instance, if two layers use the same group labels for the vertices, but one layer is assortative while the other is disassortative, then knowing one of them is very helpful in predicting the other. In this sense they are closely related, even though they are statistically very different, and indeed

anticorrelated with each other.

Our proposal to capture this more general kind of interdependence is based on the idea that two layers are interdependent if and only if the structure of one layer provides meaningful knowledge about the structure of the other. Specifically, by performing link prediction in one layer with or without information about another layer, we quantify the extent to which these two layers are related.

Since our model is generative, it naturally includes a framework to predict links given partially-observed data: simply use the known data to estimate the parameters, and then use these estimated parameters to compute $M_{ij}^{(\alpha)}$, i.e., the expected number of links of type α between each pair of nodes i, j. We then rank each missing entry in the adjacency tensor according to $M_{ij}^{(\alpha)}$. The method succeeds to the extent that true missing links are given a higher estimate of $M_{ij}^{(\alpha)}$ than false ones. We follow [41] in defining the accuracy as the area under the receiver-operator curve or AUC [42]. This is the probability that a random true positive is ranked above a random true negative; thus the AUC is 1 for perfect prediction, and $1/2$ for chance.

To test our ability to predict layer α, we perform experiments with 5-fold cross validation. That is, we hold out 20% of its adjacency matrix $A^{(\alpha)}$, hiding those entries from the algorithm. We infer the model parameters using the remaining 80% as a training dataset, with or without knowledge of other layers of the network, and measure

	$G = 1$				$G = 2$				$G = 3$			
	CS	σ	L_1	σ	CS	σ	L_1	σ	CS	σ	L_1	σ
MULTITENSOR	**0.984**	0.008	**0.06**	0.01	**0.990**	0.001	**0.058**	0.003	**0.989**	0.001	**0.056**	0.002
Diagonal	0.61	0.01	0.49	0.01	0.91	0.01	0.22	0.01	0.63	0.02	0.48	0.01
BPTF	0.60	0.02	0.48	0.01	0.90	0.03	0.21	0.05	0.63	0.03	0.48	0.01

TABLE III. Cosine similarity and L_1 norm for mixed structure synthetic networks. CS as in Eq. (11) and L_1 norm as in Eq. (12) are calculated between membership vectors $\bar{u}$, $\bar{v}$ of the inferred partitions and the ground truth [$\bar{u}_0 = \bar{v}_0 = (1,0)$, $(0,1)$ for nodes in group 1 and 2, respectively]. Results are averages and standard deviations of the results on the two memberships over 10 networks sampled from each network type G, as described in Table II. The best performance is indicated in boldface.

the inferred model's accuracy on the held-out entries in $A^{(\alpha)}$. The independence of the training and test datasets makes cross-validation a robust method against overfitting. Note that holding out 20% of a layer does not mean removing 20% of the nodes or 20% of the links, but rather hiding 20% of the entries of its adjacency matrix, including both zeros and ones. This means that we just the accuracy of our link prediction algorithm on both links and non-links.

The final AUC is the average obtained over the 5 folds, each of which holds out a different subset of 20%. Clearly this AUC depends both on the layer α we are trying to predict, and on what set of other layers we give the algorithm access to.

Given this framework, we can define the pairwise interdependence between two layers α, β as follows. We perform link prediction on layer α, with 20% of the entries of $A^{(\alpha)}$ hidden; but we do this first by giving the algorithm access only to the rest of $A^{(\alpha)}$, and then by giving it access to all of $A^{(\beta)}$ as well. We then measure the difference in the AUC between these two experiments, determining how much knowledge about layer β helps us predict layer α. We call this the *two-layer AUC*. Similarly, the *three-layer AUC* tells us how much knowledge about two layers β and γ help us predict α, and so on. Notice that if different layers have independent structure, without common underlying communities, then including one in the training set much actually decreases our ability to predict the other, causing the AUC to go down.

Computing all ℓ-*layer AUCs* would require us to try $\binom{L}{\ell}$ subsets of the layers, which becomes computationally expensive as ℓ increases. To avoid this computational bottleneck, we use a greedy bottom-up procedure in which we add one layer at a time to the training dataset, whichever one most increases the AUC, until as many layers as desired have been added. This allows us to find a small set of layers which together make it possible to predict links accurately. While this greedy procedure is not guaranteed to find the best possible subset of a given size, it is computationally efficient.

Alternatively, if the goal is to decide which layers are less informative in predicting the others, in order for instance to compress information by discarding less informative layers, we can use a top-down procedure which starts with all L layers, then iteratively removes whichever one decreases the AUC the least, until a small

informative subset of layers remains. We did not pursue this here.

In addition to this link prediction approach, we also cluster the affinity matrices $w^{(\alpha)}$ in a way similar to [18]. That is, we treat the inferred $w^{(\alpha)}$ as K^2-dimensional vectors, and cluster them in K^2-dimensional space using the k-means algorithm. Results for both notions of layer interdependence are shown in the following section.

Network	K	MULTITENSOR	Diagonal	BPTF
Teṉpaṭṭi Village	4	**0.89**	0.88	0.87
Aḻakāpuram Vill.	6	**0.93**	0.92	0.91
Malaria	3	**0.83**	0.82	0.82
Malaria	5	**0.86**	0.85	0.85
Malaria	8	**0.88**	**0.88**	**0.88**

TABLE IV. AUC for link prediction on real networks using our MULTITENSOR model, the diagonal/PARAFAC algorithm and Bayesian Poisson Tensor Factorization (BPTF) [13]. Here we look at the entire dataset at once, and define the AUC as the probability a random link is ranked above a random non-link. Results correspond to the maximum likelihood fixed point over 100 runs of each algorithm with random initial conditions. The best performance for each network is indicated by boldface. All three algorithms perform quite well when shown all layers at once, although MULTITENSOR's performance is the best by a small margin.

VI. LINK PREDICTION AND LAYER INTERDEPENDENCE IN REAL NETWORKS

To demonstrate our MULTITENSOR model and algorithm beyond synthetic data, we apply it to two real-world multilayer networks. In one network, the MULTITENSOR model finds that many layers are interdependent, revealing a shared community structure among them. However, in the other network, the models finds the layers to be independent, concluding that there exists no shared structure among them. Together, these two different scenarios illustrate the concrete use of our method in both positive result and negative result scenarios, both of which are likely to arise when analyzing real-world data.
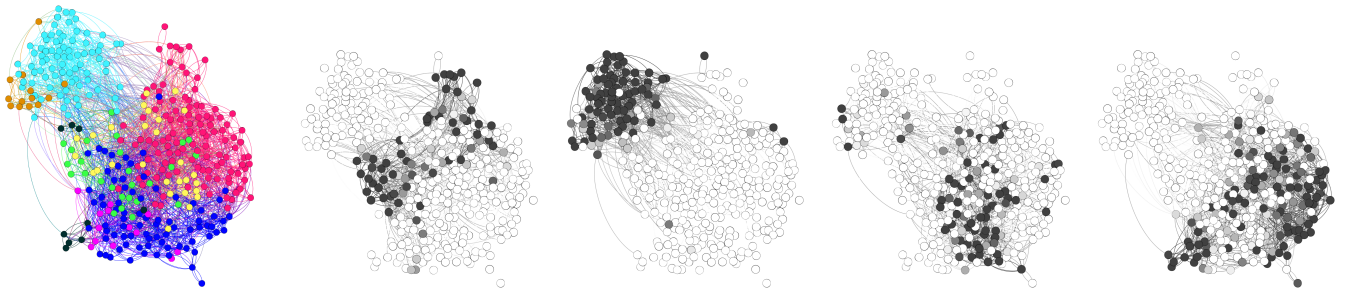
FIG. 1. Teṉpaṭṭi Village community partition. On the left we show the division by caste membership. To the right we show the membership in each of the 4 communities for each node (each figure represents one community), with color ranging from white if the normalized out-going membership $u_{ik} = 0$ to black if $u_{ik} = 1$. Values in between denote overlapping membership (grey). The fact that caste membership partially overlaps with the communities identified by our algorithm suggests a relationship between topological structure and caste, a topic that will be investigated in a future paper.
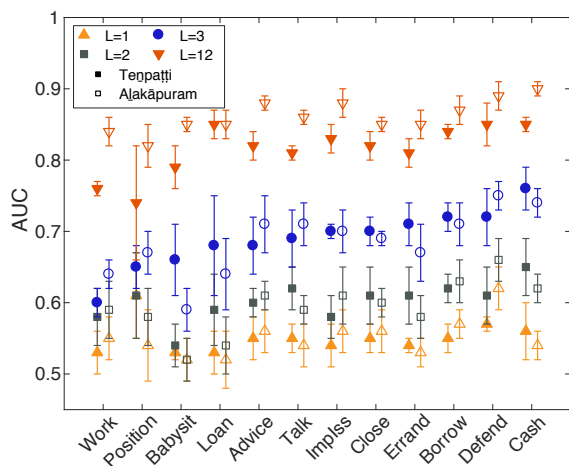
FIG. 2. Layer interdependence in the Indian social support networks. On the x-axis are the layers' labels used in the test dataset, and the y-axis shows the AUC obtained through the cross-validation schemes for measuring layer interdependence. Bold lines are for Teṉpaṭṭi Village, dashed for Aḻakāpuram Village. $L = 1$ refers to single-layer AUC, where the algorithm is only given access to that layer. $L = 2, 3, 12$ show the increase in the AUC for that layer when the algorithm is given access to L layers; for $L = 2$ and $L = 3$ we choose the best set of $L-1$ additional layers using the greedy procedure described in the previous section.

FIG. 3. Layer interdependence in the malaria network. Each of the 9 layers corresponds to a so-called "highly variable region" (HVR) of the malaria parasite genes, indicated on the x-axis, and the y-axis shows the AUC obtained through the cross-validation schemes for measuring layer interdependence. $L = 1$ refers to single-layer AUC, where the algorithm is only given access to that layer. $L = 2, 3, 9$ show the increase in the AUC for that layer when the algorithm is given access to L layers; for $L = 2$ and $L = 3$ we choose the best set of $L-1$ additional layers using the greedy procedure described in the previous section. Points and error bars are the average and standard deviation over the 5 folds of cross-validation. Unlike the social support networks, we see that the accuracy of predicting one layer actually decreases when we include others in the training set, indicating that the different layers have independent structure.

First, we analyze social support networks from two villages in the Indian state of Tamil Nadu, which we call by the pseudonyms [43] "Teṉpaṭṭi" and "Aḻakāpuram" [44, 45]. As part of a survey questionnaire, village residents were asked to name those individuals who provided them with 12 different types of support, ranging from lending them household items to helping them navigate government bureaucracy. The resulting directed networks have $N = 362$ and $N = 420$ nodes, respectively. Each type of support corresponds to a layer in these networks, giving each of them $L = 12$ layers, with average degrees ranging from 2.0 to 4.4.
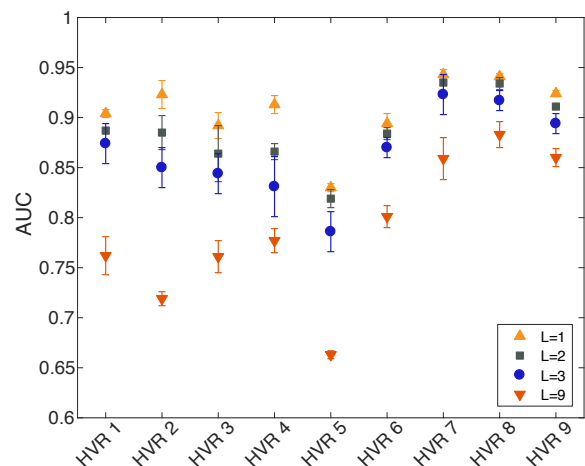
Second, we analyze the patterns of shared genetic substrings among a set of malaria parasite virulence genes [40]. Each of the $N = 307$ nodes represents a single gene, and an edge connects two genes if they share a substring of significant length. Due to the fact that the same set of genes was analyzed at nine different genetic loci (i.e., locations on the genes themselves) which are
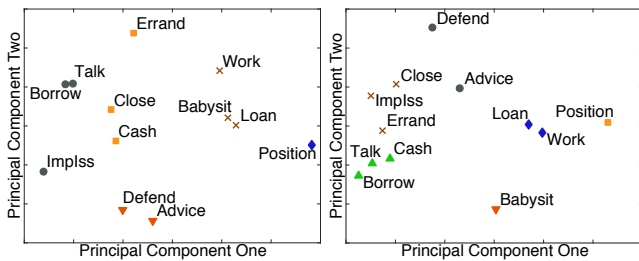
FIG. 4. Clusters of the affinity matrices in the layers of the Indian village networks, for Teṉpaṭṭi on the left and Alakāpuram on the right. Cluster labels were obtained using the k-means algorithm, treating each $w^{(\alpha)}$ as a K^2-dimensional vector, and we use PCA to visualize them in two dimensions.

called "highly variable regions" (HVRs), this undirected network has $L = 9$ layers, with average degrees ranging from 5.1 to 76.4.

The scientifically interesting questions for both networks revolve around the mechanisms driving edge formation. Hypothesized factors include kinship and caste in the Indian social support networks, and upstream promotor sequence or parasite origin in the malaria genetic networks. However, addressing these questions is beyond the scope of this paper, where we instead wish to evaluate the effectiveness of our algorithm.

One option would be to use our algorithm to cluster the nodes, and compare the resulting group assignments with metadata such as gender, caste, or geographical location. Indeed, in Figure 1 we show the community assignment for Teṉpaṭṭi predicted by our model, and compare it with the division of individuals into castes. Although the figure suggests that the partition might be correlated with caste membership, we do not expect this to be the only type of metadata correlated with the community structure, and we do not consider this correlation to be a good measure of accuracy. Here we focus instead on link prediction, and in particular on the extent to which knowledge of some layers helps us predict links in others, as described in the previous section.

As for the synthetic networks, our MULTITENSOR algorithm, the Diagonal/PARAFAC algorithm, and the BPTF algorithm each provide a framework for link prediction. Table IV reports the AUC over each entire network and for each algorithm. The algorithms' performance are roughly similar, although our algorithm has slightly higher performance. This suggests that these networks are primarily assortative; this is certainly true of the malaria network, since it is defined in terms of similarity.

To measure layer interdependence we implemented the method described in the previous section, where we attempt to predict the adjacency matrix of a given layer with 20% of its entries held out, and give the algorithm access to a subset of other layers as part of its training dataset. Interestingly, we obtain opposite results in these

two cases.

For the social networks, we find that increasing the number of layers in the training dataset does indeed improve link prediction, with a performance that increases monotonically with the number of additional layers. In Figure 2 we show that the AUC for each layer as a function of the number of layers the algorithm is given access to. We found that the best number of groups for link prediction was $K = 4$ for the first village and $K = 6$ for the second one.

Many layers viewed on their own ($L = 1$) are difficult to predict, with AUCs just above 0.5, i.e., only slightly better than chance. By giving the algorithm access to one more layer ($L = 2$) the AUC typically improves by only about 0.05. However, if we give it access to two additional layers ($L = 3$) the AUC improves significantly for almost all of the layers, and this is even more true when we give it access to the entire dataset. (For $L = 2, 3$ we use the greedy procedure to choose which $L - 1$ layers to add to the training dataset.)

Thus in these social networks, the MULTITENSOR algorithm is able to usefully apply knowledge from some layers to others. Interestingly, we also see consistency between the two villages with regard to which layers are the hardest to predict, and which layers are the most helpful to include in the training dataset. In particular, the ImpIss layer ("Who do you discuss important matters with?") is helpful in predicting many layers, while Position, Work, Loan, and Babysit are much less so, and in some cases even decrease the AUC.

We can compare this with the clustering of the L affinity matrices we obtained using standard clustering algorithms, in a spirit similar to [18]. In Figure 4 we use Principal Component Analysis [46] to visualize the L matrices $w^{(\alpha)}$, projecting them along two principal directions in K^2-dimensional space, and we give them cluster labels using the k-means algorithm [47]. Indeed we see that Position, Work, Loan, and Babysit are farther from the others, suggesting that these layers are structurally quite different from the others; note also in Figure 2 that these layers are among the hardest to predict. In contrast, ImpIss is closer to the other layers, at least for the second village, consistent with the fact that it often helps predict other layers. We also find for $L = 2$, that the Borrow layer is the most helpful when predicting the Talk layer in both villages, which is consistent with the fact that these two layers are clustered close together.

In contrast, for the malaria network we find that the best performance is obtained when no other layer is added to the dataset, meaning that prediction actually worsens monotonically as we increase the number of added layers, as shown in Figure 3. This seems to corroborate past findings [40] in an important way. Specifically, the standing hypothesis about these genes is that they are maximally diverse in order to most effectively evade the immune system. If there were correlations between loci, which we would see here as the ability of one layer to help in the link prediction of another layer, then this

would diminish these genes' overall diversity. This would diminish the amount of "immune evasion space" that is spanned by the parasites, and would therefore result in an overall fitness decrease for the parasites.

VII. CONCLUSIONS

We have proposed a generative model for multi-layer networks that extends and generalizes the mixed-membership stochastic block model. It assumes that the layers share a common community structure, but allows links in different layers to be correlated with the community memberships in different ways, such as assortative, disassortative, core-periphery, or hierarchical structure, or arbitrary mixtures thereof. It explicitly allows the communities to overlap, and can be applied to networks with directed, undirected, or integer-weighted links. We showed that it can be fit to large datasets using a scalable expectation-maximization algorithm, whose running time per iteration is linear in the total size of the dataset and which converges quickly in practice. Due to its ability to describe a wide variety of graph structures, it performs well on synthetic and real data, in terms of both community detection and link prediction.

In addition to performing community detection, the methods in this paper naturally incorporate a framework for link prediction, which we use as a quantitative definition of interdependency between the network's layers. Namely, we measure how much knowledge of one layer, or a set of layers, improves the accuracy of link prediction in another layer. This measure is quite general, and goes beyond approaches that cluster layers into strata with similar parameters (e.g. [18]); for instance, if two layers both depend strongly on the underlying communities, they will be interdependent in this sense even if one is assortative and the other is disassortative, making their affinity matrices very different. In addition to providing hints about causal or structural relationships between the layers, this notion of interdependence may be useful to choosing weights for multilayer versions of common network measures, such as eigenvector centrality [16] and modularity [7]. The same link prediction and cross-validation framework used to quantify layer interdependence can also be used to identify and avoid overfitting.

Beyond establishing high performance on synthetic datasets, we also applied our methods to two real-world datasets. We found patterns of interdependence between layers of social networks from two Indian villages, indicating correlations between different kinds of social ties, and confirmed that these patterns are largely consistent between the two villages. In contrast, when we applied our methods to a multilayer network of sequence sharing among malaria's virulence genes, we found that the layers were essentially unrelated. This suggests that similarities at different loci of the amino acid sequences are evolving under uncorrelated constraints, rigorously confirming a result based on independent analyses of each layer [40]. In both cases, our MULTITENSOR approach provided information not revealed in previous studies of these datasets, and proved to be useful in identifying not only the presence of meaningful structure, but its absence as well.

The solution we provide for the layer interdependence problem may find application beyond the analysis of extant datasets. Because our method can be used to aggregate layers into clusters, or to compress a dataset by identifying especially relevant or redundant layers, it can direct experimentalists or field researchers in learning which data to collect or prioritize. For example, if two layers of a social network are found by our methods to be redundant during a pilot study, the redundant layer need not be collected at scale. Particularly in cases where data collection is labor intensive, expensive, or generally difficult, robust solutions to the layer interdependence problem can help maximize the impact of studies constrained by limited resources in the laboratory or the field. On the other hand, when layers are found to be independent of each other, our methods provide justification for comprehensive data collection of the relevant layers.

[1] M. Costanzo, A. Baryshnikova, J. Bellay, Y. Kim, E. D. Spear, C. S. Sevier, H. Ding, J. L. Y. Koh, K. Toufighi, S. Mostafavi, *et al.*, science **327**, 425 (2010).

[2] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly, IEEE Control Systems **21**, 11 (2001).

[3] L. M. Verbrugge, Social Forces **57**, 1286 (1979).

[4] S. Wasserman and K. Faust, *Social network analysis: Methods and applications*, Vol. 8 (Cambridge university press, 1994).

[5] D. Taylor, R. S. Caceres, and P. J. Mucha, arXiv preprint arXiv:1609.04376 (2016).

[6] D. Taylor, S. Shai, N. Stanley, and P. J. Mucha, Physical Review Letters **116**, 228301 (2016).

[7] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J. P. Onnela, Science **328**, 876 (2010).

[8] M. Bazzi, L. G. Jeub, A. Arenas, S. D. Howison, and M. A. Porter, arXiv preprint arXiv:1608.06196 (2016).

[9] S. Paul and Y. Chen, arXiv preprint arXiv:1608.00623 (2016).

[10] S. Paul, Y. Chen, *et al.*, Electronic Journal of Statistics **10**, 3807 (2016).

[11] T. P. Peixoto, Physical Review E **92**, 042807 (2015).

[12] A. Schein, J. Paisley, D. Blei, and H. Wallach, in *Proceedings of the NIPS 2014 Workshop on Networks: From Graphs to Rich Data* (2014).

[13] A. Schein, J. Paisley, D. M. Blei, and H. Wallach, in *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, 2015) pp. 1045–1054.

[14] A. Schein, M. Zhou, D. M. Blei, and H. Wallach, Proceedings of the 33rd International Conference on Machine Learning **48** (2016).

[15] T. Valles-Catala, F. A. Massucci, R. Guimera, and M. Sales-Pardo, Physical Review X **6**, 011036 (2016).

[16] L. Solá, M. Romance, R. Criado, J. Flores, A. G. del Amo, and S. Boccaletti, Chaos **23**, 033131 (2013).

[17] M. De Domenico, V. Nicosia, A. Arenas, and V. Latora, Nature Communications **6**, 6864 (2015).

[18] N. Stanley, S. Shai, D. Taylor, and P. Mucha, IEEE Transactions on Network Science and Engineering **3**, 95 (2016).

[19] M. E. J. Newman and E. A. Leicht, Proceedings of the National Academy of Sciences USA **104**, 9564 (2007).

[20] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, Journal of Machine Learning Research **9**, 1981 (2008).

[21] J. J. Ramasco and M. Mungan, Physical Review E **77**, 036122 (2008).

[22] B. Ball, B. Karrer, and M. E. J. Newman, Physical Review E **84**, 036103 (2011).

[23] P. K. Gopalan and D. M. Blei, Proceedings of the National Academy of Sciences USA **110**, 14534 (2013).

[24] J. Yang, J. McAuley, and J. Leskovec, in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining* (ACM, 2014) pp. 323–332.

[25] M. Zhou, in *AISTATS*, Vol. 38 (San Diego, CA, USA, 2015) pp. 1135–1143.

[26] T. Hofmann, in *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence* (Morgan Kauf-mann Publishers Inc., 1999) pp. 289–296.

[27] D. M. Blei, A. Y. Ng, and M. I. Jordan, Journal of Machine Learning research **3**, 993 (2003).

[28] T. G. Kolda and B. W. Bader, SIAM Review **51**, 455 (2009).

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

[30] J. Canny, in *Proceedings of the 27th annual international ACM SIGIR conference on Research and Development in Information Retrieval* (ACM, 2004) pp. 122–129.

[31] D. B. Dunson and A. H. Herring, Biostatistics **6**, 11 (2005).

[32] A. T. Cemgil, Computational Intelligence and Neuroscience **2009**, 785152 (2009).

[33] P. Gopalan, J. M. Hofman, and D. M. Blei, in *Proceedings of the 31-st Conference on Uncertainty in Artificial Intelligence* (Amsterdam, Netherlands, AUAI Press, 2015) pp. 122–129.

[34] E. C. Chi and T. G. Kolda, SIAM Journal on Matrix Analysis and Applications **33**, 1272 (2012).

[35] L. R. Tucker, Psychometrika **31**, 279 (1966).

[36] R. A. Harshman, UCLA Working Papers in Phonetics **16**, 1 (1970).

[37] J. D. Carroll and J.-J. Chang, Psychometrika **35**, 283 (1970).

[38] Code available at github.com/cdebacco/MultiTensor.

[39] B. Karrer and M. E. J. Newman, Physical Review E **83**, 016107 (2011).

[40] D. B. Larremore, A. Clauset, and C. O. Buckee, PLoS Computational Biology **9**, e1003268 (2013).

[41] A. Clauset, C. Moore, and M. E. J. Newman, Nature **453**, 98 (2008).

[42] J. A. Hanley and B. J. McNeil, Radiology **143**, 29 (1982).

[43] Pseudonyms for villages are used for privacy reasons.

[44] E. A. Power, *Building bigness: Religious practice and social support in rural South India*, Doctoral Dissertation, Stanford University, Stanford, CA (2015).

[45] E. A. Power, Nature Human Behaviour **1**, 0057 (2017).

[46] I. Jolliffe, *Principal component analysis* (Wiley Online Library, 2002).

[47] J. MacQueen *et al.*, Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability **1**, 281 (1967).