Clustering of heterogeneous populations of networks

Jean-Gabriel Young [1,2,*] Alec Kirkley,[3,4,*] and M. E. J. Newman [3,5]

[1]*Department of Mathematics and Statistics, University of Vermont, Burlington, Vermont 05405, USA*
[2]*Vermont Complex Systems Center, University of Vermont, Burlington, Vermont 05405, USA*
[3]*Department of Physics, University of Michigan, Ann Arbor, Michigan 48109, USA*
[4]*School of Data Science, City University of Hong Kong, 999077, Hong Kong*
[5]*Center for the Study of Complex Systems, University of Michigan, Ann Arbor, Michigan 48109, USA*

Statistical methods for reconstructing networks from repeated measurements typically assume that all measurements are generated from the same underlying network structure. This need not be the case, however. People's social networks might be different on weekdays and weekends, for instance. Brain networks may differ between healthy patients and those with dementia or other conditions. Here we describe a Bayesian analysis framework for such data that allows for the fact that network measurements may be reflective of multiple possible structures. We define a finite mixture model of the measurement process and derive a Gibbs sampling procedure that samples exactly from the full posterior distribution of model parameters. The end result is a clustering of the measured networks into groups with similar structure. We demonstrate the method on both real and synthetic network populations.

I. INTRODUCTION

Many modern network analyses involve large corpora of networks defined on a constant set of nodes. Examples include repeated observations of proximity networks [1], longitudinal studies of social relationships among fixed groups of individuals [2], and measurements of neural connectivity across patients [3].

As pointed out in a number of recent studies, repeated measurements offer a unique opportunity to carry out robust network analyses [4–13]. With such large, rich data sets it is possible to extract information not available from a single network observation. For example, one can use repeated observations to infer a latent network structure despite the presence of measurement error—given multiple noisy realizations of the same network it becomes possible to estimate the network most likely to have generated the measurements. In the case of repeated observation of social interactions, for instance, one expects friends to interact more often than strangers, and repeated observations, properly analyzed, can thus reveal the network of friendships [6]. In neuroscience applications, one might focus on the identification of a characteristic brain connectivity pattern for a set of patients with a common condition, which is uncovered by combining the measurements made on all patients [3].

Implicit in these methods is the assumption that there is a single underlying network that determines all the observations, but this need not be the case. Patients undergoing brain scans might have different underlying conditions, for instance, leading to different measured connectivity patterns from one patient to another [14]. Likewise, acquaintances might interact differently based on the time of the day or location, leading, for example, to different social networks during working hours and outside of work.

Treating a population of observed networks as realizations of the same underlying graph might thus adversely affect our analysis of these systems, just as ignoring the multimodal nature of a distribution of numbers may lead to mischaracterization of a study sample. In some cases we know when the underlying network changes, such as when measurements are made on different days of the week, in which case the data can be divided up and processed separately using standard techniques. Often, however, the underlying generative processes are unknown, making it hard to split the sample manually. For example, if the brain activity of a patient is measured as they are put under anesthesia, then we know that the activity will undergo a change from one state to another but the transition point itself is unknown. It could be that the activity changes well before a visible loss of consciousness, or well after, so one cannot rely on direct observations to determine the change point; one needs to infer the transition from the networks. In this and other similar cases, one must simultaneously determine whether the data is best described by a single underlying network or many, what these networks are, and which measurements should be attributed to which underlying network.

In this paper we describe a framework for modeling heterogeneous populations of networks. We view each network in a population as a noisy realization of one underlying network out of multiple possibilities, which we call *modes*. We model the data as being generated from a superposition or finite mixture of graph distributions [15,16], which provides sufficient flexibility to accommodate highly heterogeneous populations.

---

*Equal contributors.

We demonstrate how our methodology allows us to simultaneously infer the underlying networks and cluster the observed networks such that each cluster of observations consists of noisy realizations of a single underlying network mode. Our framework also provides a natural means for selecting the number of modes that describe a given set of networks.

Previous work on statistical modeling of network measurements has mostly focused on the unimodal case, where it is assumed that a population of networks is best described by a single underlying network, of which the observations are noisy realizations [2,4,6,8,17–20]. Some recent approaches have explicitly incorporated multiple modes using latent space representations [21–24], exponential random graph models [25], or parametric network models [26]. Further afield are studies aiming to cluster the layers of multilayer networks (where layers can be categorical or temporal), for instance when trying to detect change points [27,28]—abrupt changes in sequences of network snapshots—or as a side-effect of pooling information across network layers when clustering their nodes [29,30]. Among these works the approach most closely related to our own is perhaps that of La Rosa *et al.* [31], who extend unimodal metric models of network populations [18,19] to the multimodal case with finite mixtures. In the unimodal case all possible networks are assigned a distance to the mode and networks closer to the mode are assumed more likely. The extension described by La Rosa *et al.* employs multiple modes and can thus more faithfully model diverse populations of networks. While mathematically elegant, however, this approach has some shortcomings. For example, the model is not easily estimated when the distance between networks is difficult to compute [32]. The approach of La Rosa *et al.* also does not differentiate between false-positive and false-negative rates, which may differ substantially and change the composition of the corresponding clusters of networks (see Appendix A).

This paper is organized as follows. First, we briefly review a previously described unimodal model of homogeneous populations of networks, then introduce our model for multimodal heterogeneous populations by building upon the unimodal case. We then discuss statistical estimation of the model using a Gibbs sampling procedure and demonstrate using synthetic data that we can recover model parameters when the level of noise in the measurements is sufficiently low. We further demonstrate our methods with an example application to a real-world network population from a longitudinal social network study.

II. THE MODEL

We consider an experiment or observational study in which N networks are measured on the same set of n nodes. The networks could record, for instance, connectivity patterns in brain scans of a cohort of N patients, in which a node represents a region of the brain and edges indicate when two regions are sufficiently connected in a given patient [33]. Or the population of networks could encode a set of relationships among a group of people such as students in a school with nodes representing the students and edges indicating when two students are within a certain physical distance of one another during a specified time interval [34]. We record the networks as a set of

N adjacency matrices $\mathscr{D} = \{\boldsymbol{D}^{(t)}\}_{t=1}^N$ indexed by $t = 1, \ldots, N$, where $\boldsymbol{D}^{(t)}$ is an $n \times n$ matrix with element $D_{ij}^{(t)} = 1$ if there is an edge between nodes i and j in network t, and $D_{ij}^{(t)} = 0$ otherwise. For simplicity of presentation we will assume the networks to be undirected so that $D_{ij}^{(t)} = D_{ji}^{(t)}$, but our methods are easily adapted to directed networks.

A. Homogeneous populations of networks

A variety of approaches have been proposed for analyzing multiple network observations of this kind when each observation is believed to be a noisy realization of a single underlying "ground truth" network—see, for example, Refs. [2,6,8,9,18,19]. As a starting point for our discussion we adopt the approach of Refs. [2,6] in which one defines a model to describe how the observed networks are related to the underlying ground truth. The particular model in this case has two parameters: $\alpha \in [0, 1]$ is the a true-positive rate for the edges and $\beta \in [0, 1]$ is the false-positive rate. In other words, if there is an edge connecting two nodes in the adjacency matrix $\boldsymbol{A}$ of the ground truth network, then that edge will be observed with independent probability α in a noisy realization $\boldsymbol{D}^{(t)}$, while an absent edge in $\boldsymbol{A}$ will be mistakenly observed with probability β. The probability of observing a complete network $\boldsymbol{D}^{(t)}$ under this model is then given by

$$P(\boldsymbol{D}^{(t)}|\boldsymbol{A}, \alpha, \beta) = \prod_{i<j} \left[\alpha^{D_{ij}^{(t)}}(1-\alpha)^{1-D_{ij}^{(t)}}\right]^{A_{ij}}$$
$$\times \left[\beta^{D_{ij}^{(t)}}(1-\beta)^{1-D_{ij}^{(t)}}\right]^{1-A_{ij}}, \quad (1)$$

where the product is over all (unordered) pairs of nodes. When the individual observed networks in $\mathscr{D}$ are independent of one another the likelihood of the complete population $\mathscr{D}$ is

$$P(\mathscr{D}|\boldsymbol{A}, \alpha, \beta) = \prod_{t=1}^N P(\boldsymbol{D}^{(t)}|\boldsymbol{A}, \alpha, \beta)$$
$$= \prod_{t=1}^N \prod_{i<j} \left[\alpha^{D_{ij}^{(t)}}(1-\alpha)^{1-D_{ij}^{(t)}}\right]^{A_{ij}}$$
$$\times \left[\beta^{D_{ij}^{(t)}}(1-\beta)^{1-D_{ij}^{(t)}}\right]^{1-A_{ij}}$$
$$= \prod_{i<j} \left[\alpha^{X_{ij}}(1-\alpha)^{N-X_{ij}}\right]^{A_{ij}}$$
$$\times \left[\beta^{X_{ij}}(1-\beta)^{N-X_{ij}}\right]^{1-A_{ij}}, \quad (2)$$

where $X_{ij} = \sum_t D_{ij}^{(t)}$ is the total number of times an edge is observed between node i and j in the N samples.

With Eq. (2) in hand, one can simulate observed networks or perform inference about the generative process and, for instance, estimate $\boldsymbol{A}$ from the data $\mathscr{D}$. Applications of this kind are studied at length in Refs. [2,6,8,12] among others. Our goal here is to extend the model to accommodate heterogeneous network populations, making it suitable for inference about a broader range of network data.

B. Heterogeneous populations of networks

Consider again our sample $\mathscr{D}$ of N networks sharing the same set of nodes. We can allow for heterogeneity in these

samples by letting the individual networks be noisy realizations of $K \ll N$ different underlying network modes $\mathscr{A} = (\boldsymbol{A}^{(1)}, ..., \boldsymbol{A}^{(K)})$, rather than just a single mode as before. In the context of network neuroscience, for example, we could repeatedly measure the brain of a single patient t undergoing a transition from the conscious to unconscious state, which would imply that $K = 2$. Alternatively, we might have a population of patients, some of whom have a neurological disorder such as Alzheimer's disease and some of whom do not.

Each sample t will be assigned to a network mode u so that $\boldsymbol{D}^{(t)}$ is a noisy realization of $\boldsymbol{A}^{(u)}$. We use a variable z_{tu} to encode this information thus:

$$z_{tu} = \begin{cases} 1 & \text{if } \boldsymbol{D}^{(t)} \text{ is a noisy realization of network } \boldsymbol{A}^{(u)}, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

With this notation we can denote the assignment of samples to modes as a $N \times K$ matrix $\boldsymbol{Z}$, whose rows correspond to samples and columns to reference networks. Each sample corresponds to precisely one reference network, so the rows of Z satisfy $\sum_{u=1}^{K} z_{tu} = 1$. Further, the column sums $N_u = \sum_{t=1}^{N} z_{tu}$ correspond to the number of samples in $\mathscr{D}$ generated from mode u.

To model the now heterogeneous population of networks we use the same generative process as before, Eq. (1), but with a set $\mathscr{A}$ of multiple underlying networks instead of just one. Since different modes may display different rates of measurement error, we allow each mode $u = 1, \ldots, K$ to have its own associated true- and false-positive rates α_u and β_u. For notational simplicity we henceforth denote the sets of parameters $\{\alpha_u\}_{u=1}^{K}$ and $\{\beta_u\}_{u=1}^{K}$, as well as other model parameters we will introduce shortly, collectively as $\boldsymbol{\theta}$.

The likelihood of this model is analogous to that of Eq. (2) and is given by

$$P(\mathscr{D}|\boldsymbol{Z}, \mathscr{A}, \boldsymbol{\theta}) = \prod_{t=1}^{N} \prod_{u=1}^{K} \left[\prod_{i<j} \left[\alpha_u^{D_{ij}^{(t)}} (1 - \alpha_u)^{1-D_{ij}^{(t)}} \right]^{A_{ij}^{(u)}} \right.$$
$$\left. \times \left[\beta_u^{D_{ij}^{(t)}} (1 - \beta_u)^{1-D_{ij}^{(t)}} \right]^{1-A_{ij}^{(u)}} \right]^{z_{tu}}. \quad (4)$$

The main difference is the product over modes u and the inclusion of the variable z_{tu} to encode the modes. Performing the product over network samples $t = 1, \ldots, N$, we can also write this as

$$P(\mathscr{D}|\boldsymbol{Z}, \mathscr{A}, \boldsymbol{\theta}) = \prod_{u=1}^{K} \prod_{i<j} \left[\alpha_u^{X_{ij}^u} (1 - \alpha_u)^{N_u - X_{ij}^u} \right]^{A_{ij}^{(u)}}$$
$$\times \left[\beta_u^{X_{ij}^u} (1 - \beta_u)^{N_u - X_{ij}^u} \right]^{1-A_{ij}^{(u)}}, \quad (5)$$

where $X_{ij}^u = \sum_{t=1}^{N} D_{ij}^{(t)} z_{tu}$ is the number of interactions observed between i and j across all noisy realizations of the underlying network u.

In most cases, however, the mode assignments $\boldsymbol{Z}$ will be unknown—if not, then we could just divide up the the networks into their groups and model them separately as disjoint homogeneous populations. We can eliminate $\boldsymbol{Z}$ by marginal-

izing over its possible values in Eq. (4) thus:

$$P(\mathscr{D}|\mathscr{A}, \boldsymbol{\theta}) = \sum_{\boldsymbol{Z}} P(\mathscr{D}, \boldsymbol{Z}|\mathscr{A}, \boldsymbol{\theta})$$
$$= \sum_{\boldsymbol{Z}} P(\mathscr{D}|\boldsymbol{Z}, \mathscr{A}, \boldsymbol{\theta}) P(\boldsymbol{Z}|\boldsymbol{\theta}), \quad (6)$$

where the sum is over all possible assignment matrices $\boldsymbol{Z}$ and $P(\boldsymbol{Z}|\boldsymbol{\theta})$ is a prior probability on the assignments. We choose the convenient categorical prior

$$P(\boldsymbol{Z}|\boldsymbol{\theta}) = \prod_{t=1}^{N} \prod_{u=1}^{K} \pi_u^{z_{tu}} \equiv \prod_u \pi_u^{N_u}, \quad (7)$$

with π_u the prior probability that a sample $\boldsymbol{D}^{(t)}$ is assigned to mode $\boldsymbol{A}^{(u)}$ (so that $\sum_{u=1}^{K} \pi_u = 1$), and the convention that $\boldsymbol{\theta}$ now additionally includes the parameters $\{\pi_u\}_{u=1}^{K}$. Substituting Eqs. (4) and (7) into Eq. (6), we then find that

$$P(\mathscr{D}|\mathscr{A}, \boldsymbol{\theta}) = \prod_{t=1}^{N} \sum_{u=1}^{K} \pi_u \alpha_u^{Y_{tu}^{11}} (1 - \alpha_u)^{Y_{tu}^{01}} \beta_u^{Y_{tu}^{10}} (1 - \beta_u)^{Y_{tu}^{00}},$$
$$(8)$$

where

$$Y_{tu}^{00} = \sum_{i<j} \left(1 - D_{ij}^{(t)}\right)\left(1 - A_{ij}^{(u)}\right), \quad Y_{tu}^{11} = \sum_{i<j} D_{ij}^{(t)} A_{ij}^{(u)}$$
$$Y_{tu}^{10} = \sum_{i<j} D_{ij}^{(t)} \left(1 - A_{ij}^{(u)}\right), \quad Y_{tu}^{01} = \sum_{i<j} \left(1 - D_{ij}^{(t)}\right) A_{ij}^{(u)}, \quad (9)$$

which we can think of as the elements of four matrices measuring agreement between samples and modes. (Refer to Table I in the Appendix for a summary of notations used in this paper.) For example, Y_{tu}^{00} counts the number of edges that are simultaneously absent in sample t and mode u and can be thought of as an entry of a matrix $\boldsymbol{Y}^{00}$ that records such numbers for all modes and samples.

The mixture model appearing in Eq. (8) is sufficiently flexible to account for complicated structure in populations of graphs, analogous to the flexibility seen in mixture models over distributions of numbers [16]. In Fig. 1 we show a population of small networks, generated from Eq. (8) with three modes ($K = 3$). As a visualization of the workings of the model we embed the networks it generates in a two-dimensional space (using multidimensional scaling [35] applied to Hamming distance) and then compute the density of network samples in that space, which is shown in the contour plot. From this plot one can see that the networks generated form clear clusters around the three representative modes, with the model introducing some noise.

III. ESTIMATION OF THE MODEL USING GIBBS SAMPLING

In a typical application of the model, we observe a population of networks $\mathscr{D}$ and we want to determine $\mathscr{A}$, $\boldsymbol{Z}$, and $\boldsymbol{\theta}$ assuming that $\mathscr{D}$ was generated from the model as described in Sec. II.[1] If we can successfully infer the parameters, then

---

[1]We may know some of these parameters, in which case the problem can be solved more easily.
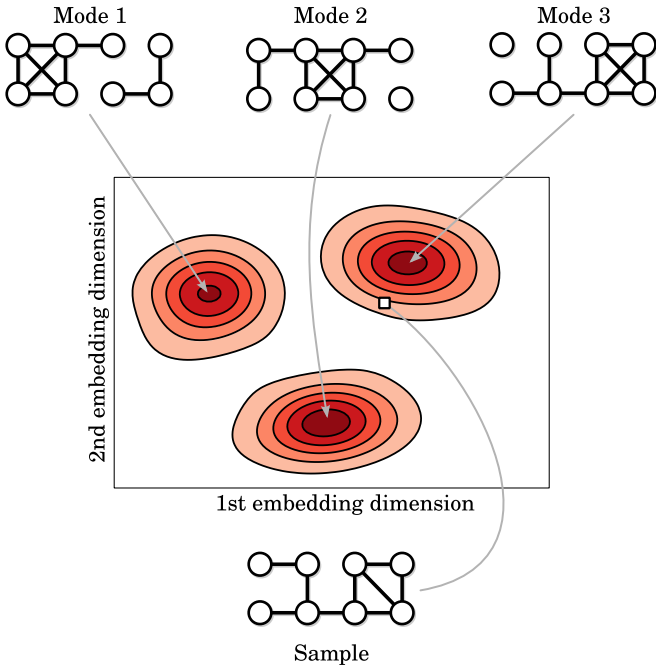
FIG. 1. Contour map of a population of networks generated from the heterogeneous model of Eq. 8, as embedded in a two-dimensional space using multidimensional scaling. Darker colors indicate a higher density of networks and the three peaks correspond to the modes used in the model, which are illustrated at the top. For this simple example, we draw networks from each of the three modes with identical probability and use the same true- and false-positive rates for all modes, namely, $\alpha_u = 0.9$ and $\beta_u = 0.1$ for $u = 1, 2, 3$. As illustrated at the bottom of the figure, a network resembling mode 3 sits close to that mode in the low-dimensional space.

we will have not only denoised the network samples $\mathscr{D}$ by finding each network's mode, but we will have also clustered the data into homogeneous classes of networks—two tasks of considerable scientific interest.

Here we adopt a Bayesian approach to the estimation problem, which allows us to draw directly from standard model-based clustering techniques [16]. The starting point for this approach is the posterior distribution for the quantities of interest:

$$P(\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta} | \mathscr{D}) = \frac{P(\mathscr{D} | \mathbf{Z}, \mathscr{A}, \boldsymbol{\theta}) P(\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta})}{P(\mathscr{D})}, \quad (10)$$

where $\boldsymbol{\theta}$ collectively denotes the rate parameters α and β as well as the mixture weights π, and $P(\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta})$ is a prior over all the unknown quantities. Note that the number K of reference networks does not appear in the equation; here we use a parametric approach and treat K as a known quantity to be handled separately. There are a number of more complex Bayesian nonparametric model alternatives one could consider that would permit Gibbs sampling and allow K to vary, including Dirichlet process mixture models [36], but we will not explore these options here. Instead, the procedure used to infer the posterior distribution in Eq. (10) can be run for multiple values of K and the optimal value can be chosen by identifying which value of K produced the samples with the highest probability. One can add a prior distribution on K to

penalize parametrizations with a greater number of clusters, thereby preventing uninformative solutions with $K \simeq N$ from being chosen. However, the prior probability $P(\mathbf{Z} | \boldsymbol{\theta})$ in Eq. (7) already penalizes solutions with $K \gg 1$, and this seems sufficient for making reasonable inference in practice—see the results in Sec. IV.

We have already discussed the likelihood in Eq. (10), which leaves us with the task of specifying the prior distribution $P(\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta})$ to complete the model. We use a prior that factorizes in the form

$$P(\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta}) = P(\mathbf{Z} | \boldsymbol{\theta}) P(\mathscr{A} | \boldsymbol{\theta}) P(\boldsymbol{\theta}). \quad (11)$$

The first probability on the right-hand side, $P(\mathbf{Z} | \boldsymbol{\theta})$, is specified in Eq. (7). It is the prior over assignments $\mathbf{Z}$ given the vector of group assignment probabilities π, which we have included in $\boldsymbol{\theta}$. For the other two probabilities we select the simplest priors possible. For the networks, we set

$$P(\mathscr{A} | \boldsymbol{\theta}) = \prod_{u=1}^{K} P(\mathbf{A}^{(u)}) = \prod_{u=1}^{K} \prod_{i<j} \rho^{A_{ij}^{(u)}} (1 - \rho)^{1 - A_{ij}^{(u)}}$$
$$= \rho^{M^*} (1 - \rho)^{K \binom{n}{2} - M^*}, \quad (12)$$

where $M^* = \sum_{i<j} \sum_u A_{ij}^{(u)}$ is the total number of edges in all reference networks and ρ, which we also now include in $\boldsymbol{\theta}$ for notational simplicity, is the prior probability of an edge being placed between each pair of nodes (i, j). For the parameters $\boldsymbol{\theta}$—i.e., $\{\alpha_u, \beta_u, \pi_u\}_{u=1}^{K}$ and ρ—we use uniform priors so that their overall contribution to the posterior distribution is a multiplicative constant. (One can instead opt for conjugate prior distributions for these quantities, which would be the beta and Dirichlet distributions in this case, without much mathematical complication; see Appendix B.)

With these priors in place, estimation of the heterogeneous network model in Sec. II amounts to either finding point estimates of the parameters in Eq. (10), or to computing averages over their distribution. Here, we employ a sampling algorithm that allows us to accomplish either of these tasks. The algorithm, described below, returns a series of samples $(\mathscr{A}_s, \mathbf{Z}_s, \boldsymbol{\theta}_s)_{s=1,\dots,S}$, each giving a possible choice of the network modes, assignment matrices, and parameters from which the networks $\mathscr{D}$ could have been generated. These samples can in turn be used to approximate expectations

$$E[f(\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta})] = \int \sum_{\mathbf{Z}, \mathscr{A}} f(\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta}) P(\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta} | \mathscr{D}) d\theta$$
$$\simeq \frac{1}{S} \sum_{s=1}^{S} f(\mathbf{Z}_s, \mathscr{A}_s, \boldsymbol{\theta}_s),$$

of arbitrary functions over the joint posterior distribution of the model, to find estimates of the model parameters, or to assess our uncertainty about the exact values of the parameters.

A. Gibbs sampling

We use a Gibbs sampling method to generate our samples [37]. The Gibbs sampler operates by cycling through the parameters $(\mathscr{A}, \mathbf{Z}, \boldsymbol{\theta})$ of the model and generating values for each set of parameters in turn, while conditioning on the

values of the remaining parameters, starting from random initial values. Specifically, we

(1) Initialize an iteration counter $\tau \leftarrow 0$ then draw random initial values $\mathscr{A}_0, \mathbf{Z}_0, \boldsymbol{\theta}_0$ of the parameters.

(2) Draw new modes $\mathscr{A}_{\tau+1} \sim P(\mathscr{A}|\mathscr{D}, \mathbf{Z}_\tau, \boldsymbol{\theta}_\tau)$.

(3) Draw new assignments $\mathbf{Z}_{\tau+1} \sim P(\mathbf{Z}|\mathscr{D}, \mathscr{A}_{\tau+1}, \boldsymbol{\theta}_\tau)$.

(4) Draw new parameters $\boldsymbol{\theta}_{\tau+1} \sim P(\boldsymbol{\theta}|\mathscr{D}, \mathscr{A}_{\tau+1}, \mathbf{Z}_{\tau+1})$.

(5) Increment τ and repeat from step 2.

Samples drawn from this random process at sufficiently large intervals will be distributed according to the joint posterior distribution of Eq. (10) [37].

The conditional probability distributions needed for Gibbs sampling can be calculated from the model's likelihood given in Eqs. (4) and (5) and from the priors of Eqs. (7) and (12). First, we have the probability of a given set of reference networks $\mathscr{A}$, conditioned on the data $\mathscr{D}$ and the values of $\mathbf{Z}$ and $\boldsymbol{\theta}$:

$$P(\mathscr{A}|\mathscr{D}, \mathbf{Z}, \boldsymbol{\theta}) = \frac{P(\mathscr{D}|\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta})P(\mathscr{A}|\boldsymbol{\theta})}{\sum_{\mathscr{A}} P(\mathscr{D}|\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta})P(\mathscr{A}|\boldsymbol{\theta})}, \qquad (13)$$

which we can transform into an explicit expression by substituting in the compact likelihood of Eq. (5) and our prior over $\mathscr{A}$. We find

$$P(\mathscr{A}|\mathscr{D}, \mathbf{Z}, \boldsymbol{\theta}) = \prod_{u=1}^{K} \prod_{i<j} \left(Q_{ij}^{u} \right)^{A_{ij}^{(u)}} \left(1 - Q_{ij}^{u} \right)^{1 - A_{ij}^{(u)}}, \qquad (14)$$

where

$$Q_{ij}^{u} = \left[1 + \frac{(1-\rho)}{\rho} \left(\frac{\beta_u}{\alpha_u} \right)^{X_{ij}^{u}} \left(\frac{1-\beta_u}{1-\alpha_u} \right)^{N_u - X_{ij}^{u}} \right]^{-1} \qquad (15)$$

is the probability that nodes i and j are connected in the uth reference network, when we know the cluster assignments and parameter values.

Likewise, the conditional probability of the cluster assignments $\mathbf{Z}$ can be calculated as

$$P(\mathbf{Z}|\mathscr{D}, \mathscr{A}, \boldsymbol{\theta}) = \frac{P(\mathscr{D}|\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta})P(\mathbf{Z}|\boldsymbol{\theta})}{\sum_{\mathbf{Z}} P(\mathscr{D}|\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta})P(\mathbf{Z}|\boldsymbol{\theta})}. \qquad (16)$$

We already have an expression for the denominator—it is the mixture likelihood of Eq. (8)—and the numerator can be calculated by combining the prior in Eq. (7) with another form of Eq. (4),

$$P(\mathscr{D}|\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta})$$

$$= \prod_{t=1}^{N} \prod_{u=1}^{K} \left[(\alpha_u)^{Y_{tu}^{11}} (1-\alpha_u)^{Y_{tu}^{01}} (\beta_u)^{Y_{tu}^{10}} (1-\beta_u)^{Y_{tu}^{00}} \right]^{z_{tu}}, \quad (17)$$

where the four $\mathbf{Y}$ matrices are the ones defined in Eq. (9). Putting everything together we find that

$$P(\mathbf{Z}|\mathscr{D}, \mathscr{A}, \boldsymbol{\theta}) = \prod_{t=1}^{N} \prod_{u=1}^{K} R_{tu}^{z_{tu}}, \qquad (18)$$

where

$$R_{tu} = \frac{\pi_u (\alpha_u)^{Y_{tu}^{11}} (1-\alpha_u)^{Y_{tu}^{01}} (\beta_u)^{Y_{tu}^{10}} (1-\beta_u)^{Y_{tu}^{00}}}{\sum_v \pi_v (\alpha_v)^{Y_{tv}^{11}} (1-\alpha_v)^{Y_{tv}^{01}} (\beta_v)^{Y_{tv}^{10}} (1-\beta_v)^{Y_{tv}^{00}}}$$

can be interpreted as the probability that sample t is a noisy version of mode u, conditioned on known values for the modes and parameters.

The update equation for the remaining parameters is given by

$$P(\boldsymbol{\theta}|\mathscr{D}, \mathbf{Z}, \mathscr{A}) = \frac{P(\mathscr{D}|\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta})P(\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta})}{\int P(\mathscr{D}|\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta})P(\mathbf{Z}, \mathscr{A}, \boldsymbol{\theta})d\boldsymbol{\theta}}. \qquad (19)$$

The integral appearing in the denominator has a closed-form solution, but we do not actually need to calculate it since our goal is to sample from the distribution, so we only need to know how Eq. (19) depends on $\boldsymbol{\theta}$, and the denominator is, by definition, not a function of $\boldsymbol{\theta}$. Upon substituting the likelihood and priors into Eq. (19) we find that the conditional distribution for $\boldsymbol{\theta}$ factorizes as

$$P(\boldsymbol{\theta}|\mathscr{D}, \mathbf{Z}, \mathscr{A}) \propto \rho^{M^*} (1-\rho)^{K\binom{N}{2} - M^*} \prod_{u=1}^{K} \pi_u^{N_u} \prod_{u=1}^{K} (\alpha_u)^{W_u^{11}}$$

$$\times (1-\alpha_u)^{W_u^{01}} (\beta_u)^{W_u^{10}} (1-\beta_u)^{W_u^{00}}, \quad (20)$$

where the matrices

$$W_u^{pq} = \sum_{t=1}^{N} Y_{tu}^{pq} z_{tu} \qquad (21)$$

quantify the edge agreement between the mode u and all the network samples in its corresponding cluster (see Table I for a summary).

Equations (14), (18), and (20) provide us with all the conditional distributions we need to implement the Gibbs sampler. As mentioned previously, we cycle through the variables $\mathscr{A}$, $\mathbf{Z}$, and $\boldsymbol{\theta}$, generating new samples of each from Eqs. (14), (18), and (20), respectively, while using the most recent samples of the other parameters as input.

The conditional distributions all allow for straightforward sampling. As Eq. (14) and (18) show, the edges of the modes and the cluster assignments are determined by independent categorical random variables. Equation (20) further shows that the parameters $\boldsymbol{\theta}$ are independent from one another (when we condition on $\mathbf{Z}$ and $\mathscr{A}$) and that they are governed by either a beta distribution (α, β, and ρ) or a Dirichlet distribution (π). Hence, generating updated parameters for the Gibbs sampler is straightforward, since all variables can be simulated with standard univariate sampling methods available in most statistical software packages.

B. Implementation

The Gibbs sampling approach allows us to sample from the full posterior distribution of Eq. (10), which is sufficient for estimating any expectation value of interest. In its naive form, however, the Gibbs sampler is quite slow. For example, generating new modes takes $O(K\binom{n}{2})$ steps which can become an issue for larger networks. Fortunately there are some computational tricks we can use to speed up the calculation substantially.

The first observation we make is that the variables $\mathbf{Y}$ are related, such that they need not be all computed every time

the modes are updated. Using Eqs. (9) one can show that

$$Y_{tu}^{10} = M_t - Y_{tu}^{11}, \tag{22}$$

$$Y_{tu}^{01} = M_u^* - Y_{tu}^{11}, \tag{23}$$

$$Y_{tu}^{00} = \binom{n}{2} - Y_{tu}^{10} - Y_{tu}^{01} - Y_{tu}^{11}, \tag{24}$$

where M_u^* is the number of edges in network mode u, and M_t is the number of edges in the sample t. By pre-computing M_t from the data $\mathcal{D}$ once and using the above relations we can recover all the matrices Y in terms of the edge agreements Y^{11} and the counts M_u^*. Furthermore, these latter quantities can be computed rapidly by traversing edge lists, which can be done in order $O(n)$ time for sparse networks with $O(n)$ edges, using, for example, hash tables to store the lists. Thus, updates of Y only take linear time.

A second set of observations allows us to accelerate the calculation by removing redundancies from the processing of network modes. First, we notice that, conditioned on the modes, the edges of $\mathcal{A}$ are independent identically distributed Bernoulli variables as shown in Eq. (14). Second, we observe that the probabilities Q_{ij}^u needed to generate these edges are identical for all node pairs (i, j) that occur a given number of times X_{ij}^u in networks belonging to cluster u, as shown by Eq. (15). Thus, the fundamental quantity needed to track and update the modes is, in effect, X_{ij}^u rather than the modes themselves.

To make use of these observations, we denote the set of edges that occur exactly l times in cluster u as

$$T_l^u \equiv \{(i, j) : X_{ij}^u = l\}, \tag{25}$$

and the number of unique values of l observed in cluster u for the current Gibbs sample as L_u. For each cluster u, we can then replace the $\binom{n}{2}$ Bernoulli trials needed to generate a sample by a fast two-step process. First, for each of the unique L_u values of X_{ij}^u, we generate the number E_l^u of edges that will appear in the mode out of $|T_l^u|$ independent trials with a probability of success of Q_l^u corresponding to the value of Q_{ij}^u for the edge $(i, j) \in T_l^u$. Then we choose E_l^u edges uniformly at random from the set T_l^u and add these edges to the mode adjacency matrix $A^{(u)}$, which we store using an edge list implemented with a hash table. We repeat this sampling procedure for all values $l \in L_u$ in the current cluster sample and repeat for all clusters u.

For a given cluster u, this two-step process can be carried out in $\sum_{l \in L_u} (1 + |T_l^u| Q_l^u)$ operations on average, which equals $K\binom{n}{2}$ only in the worst case when every edge occurs a different number of times l in every cluster. For a more typical population of $N \gg 1$ sparse networks with $O(n)$ edges, most pairs of nodes i, j will never be connected in any of the samples, and thus the sets T_0^u are typically much larger than the sets T_l^u for $l \geqslant 1$, which will have about n edges. As a result our alternative sampling scheme offers a drastic improvement in computational complexity, since we do not need to sample all $O(n^2)$ potential edge pairs in T_0^u and the low value of Q_0^u ensures that we will rarely sample many edges from this set.

With these implementation tricks, we are able to generate complete samples relatively quickly. For example, for the

collection of 418 networks on 96 nodes analyzed in Sec. IV B below, our implementation (which is written in Python) takes 83 milliseconds on average to generate a sample on a single 2.0 GHz processor, and a thousand samples usually suffice for computing accurate estimates.

IV. RESULTS

In this section we demonstrate the performance of our method using both synthetic (computer-generated) and real-world data sets.

A. Synthetic data

As a first demonstration of our approach and the proposed model, we test its ability to infer the parameters of a population of networks generated using the model itself. We use a two-mode configuration with the modes 1 and 3 shown in Fig. 1 as the ground-truth, and a symmetric population of networks with $\pi_1 = \pi_2 = \frac{1}{2}$, such that about half of the networks in the sample are noisy samples of mode 1 and the other half are samples of mode 3. We control the difficulty of the task by varying the true-positive rate α_u and the false-positive rate β_u, with

$$\beta_u = 1 - \alpha_u = p \leqslant 0.5 \tag{26}$$

identical for all clusters u. The parameter p is the probability that $D_{ij}^{(t)}$ is a false positive or a false negative—in other words, the probability of flipping an edge/nonedge in $A^{(u)}$. For each experiment we generate a population of $N = 20, 50, 100,$ or 200 networks, which we feed into our algorithm as input $\mathcal{D}$.

For every synthetic data set $\mathcal{D}$ we generate $S = 5\,000$ samples using the Gibbs sampling algorithm and from these samples compute point-estimates $\{\hat{Z}, \hat{\mathcal{A}}, \hat{\theta}\}$. For the modes $\hat{A} = (\hat{A}^{(1)}, \dots \hat{A}^{(K)})$ we compute the posterior probability that they exist,

$$\hat{A}_{ij}^{(u)} = \frac{1}{S} \sum_{s=1}^{S} [A_{ij}^{(u)}]_s, \tag{27}$$

where $[A_{ij}^{(u)}]_s$ is one sample of edge (i, j) of mode u, while for the parameters we use the posterior means directly,

$$\hat{\theta} = \frac{1}{S} \sum_{s=1}^{S} \theta^{(s)}. \tag{28}$$

We summarize the cluster labels $\hat{Z}$ as a cluster assignment $\hat{g} = \{g_t\}$, such that

$$\hat{g}_t = \mathrm{argmax}_u \left\{ \sum_{s=1}^{S} Z_{tu}^{(s)} : u \in [K] \right\}, \tag{29}$$

where $\hat{g}_t$ is the cluster label of network t that appears most frequently in the posterior marginal distribution.

Once we have these point estimates we quantify the quality of the reconstruction by comparing them against $\{Z_{\text{true}}, \mathcal{A}_{\text{true}}, \theta_{\text{true}}\}$, the parameters used to generate the synthetic networks in the first place. For the cluster labels g we compute the variation of information [38] between the estimates and the true values. For the parameters we compute the ℓ_1 distance between the parameter vectors θ and θ_{true}. And for the modes $\mathcal{A}$ we compute the total number

of missing and spurious edges across all modes (in other words the sum of the Hamming or ℓ_1 distances between the graphs in $\hat{\mathscr{A}}$ and $\mathscr{A}_{\text{true}}$).

We vary the sizes of our synthetic populations over $N = 20, 50, 100, 200$ to test how population size affects our reconstruction performance. For each size we generate 100 realizations of the N networks and take the average reconstruction quality over these synthetic populations (except for $N = 20$ where we use 500 realizations to smooth out inherent variations). The standard errors associated with the average reconstruction performances are shown as error bars in the plots. We also use a beta prior for the density ρ with $a^* = 1$, $b^* = 20$ (see Appendix B for details), which allows for more consistent estimates of the mode edge densities in the high noise region where $p \simeq 0.5$.

The results of these experiments are shown in Fig. 2. In Fig. 2(a) we show the reconstruction performance for the cluster assignments $\mathbf{Z}$, which degrades gradually as we increase the flip probability p and does not depend strongly on the number of networks N in the population. In Fig. 2(b) we show the analogous curves for the parameters $\boldsymbol{\theta}$, which also become gradually more difficult to recover as we increase the noise level, up to a certain point, but then become *easier* to recover. The reason for this is that in the completely noisy limit ($p = 0.5$), all the information about the mode networks and clusters is destroyed, and so we end up with completely randomized clusters and modes. But this means we are likely to end up with values of α_u and β_u near the correct value of 0.5, since this is the value for clusters with no structure. We are also likely to infer values of the π_u close to the correct value of 0.5, as this is the most likely size distribution of the clusters if they are chosen completely at random. As we approach this regime, therefore, we begin to see improvement in the performance for $\boldsymbol{\theta}$ due to these effects.

Of all the model variables, the modes $\mathscr{A}$ are the easiest to recover for low levels of noise, as demonstrated by the relatively long flat portions of the curves in Fig. 2(c). In this case, the modes are recovered near-perfectly for flip probabilities less than some transitional value which depends on N. Beyond this point the noise introduced into the model through the flip probability p begins to blur the clusters of networks and introduces errors in their recovery. As demonstrated by the ordering of the curves in all panels, the reconstruction generally becomes easier as we increase N, which is expected since each new network sample gives us more data about the latent structure. Reconstructing the partition becomes slightly more difficult for larger N since this gives us more opportunities for making errors. This effect is reflected in the small increase in partition distance in the regime near $p = 0.5$.

When there is little evidence for any specific modes in the data the posterior distribution over modes will be similar to the prior distribution. Hence, we can quantify the algorithm's certainty about the results using the Kullback-Leibler (KL) divergence between the prior and posterior distributions,

$$D = \sum_{u=1}^{K} \sum_{i<j} \left[Q_{ij}^u \log \frac{Q_{ij}^u}{\rho} + \left(1 - Q_{ij}^u\right) \log \frac{\left(1 - Q_{ij}^u\right)}{(1 - \rho)} \right],$$

(30)
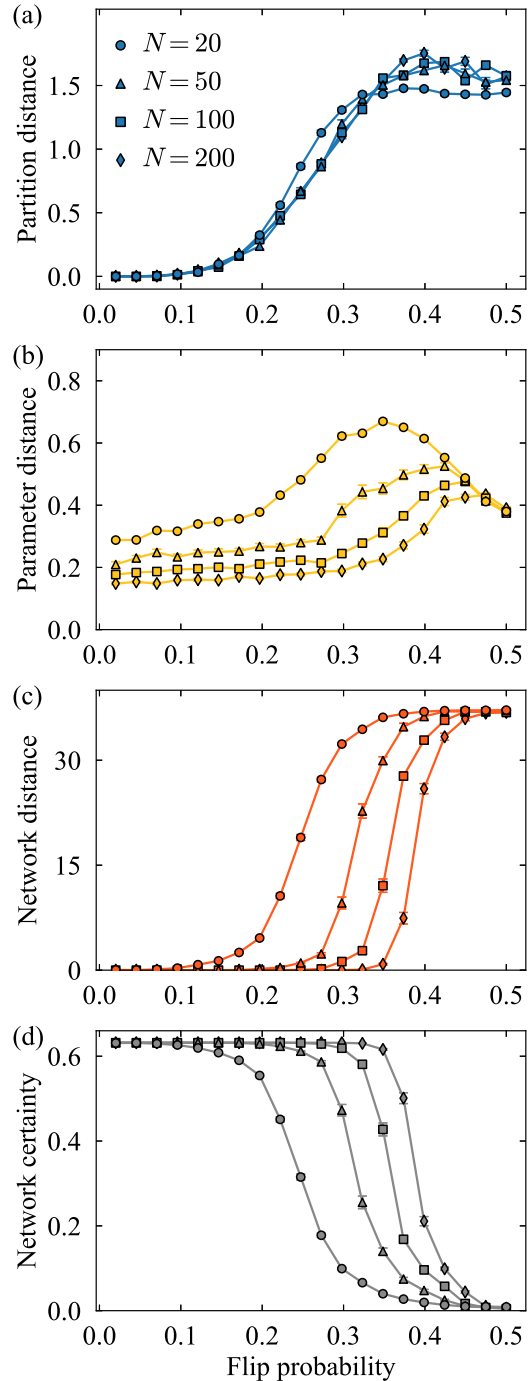
FIG. 2. Recovery performance for (a) the partitions $\mathbf{Z}$, (b) the model parameters $\boldsymbol{\theta}$, and (c) the modes $\mathscr{A}$, for a bimodal population of networks drawn from the modes 1 and 3 shown in Fig. 1 with $\pi_1 = \pi_2 = \frac{1}{2}$, across a range of flip probabilities (Eq. 26) and population sizes N. Panel (d) shows how much the model updates its certainty about the modes $\mathscr{A}$ using the data.

where Q_{ij}^u and ρ can be evaluated by averaging the posterior samples. This divergence is smallest when the posterior and prior distributions are most similar. As can be seen in Fig. 2(d), the KL divergence diminishes rapidly as we increase the flip probability and reaches zero at values of p that closely align with the loss of signal seen in Fig. 2(c). Thus, the method is able not only to recover modes in the low-flip-probability
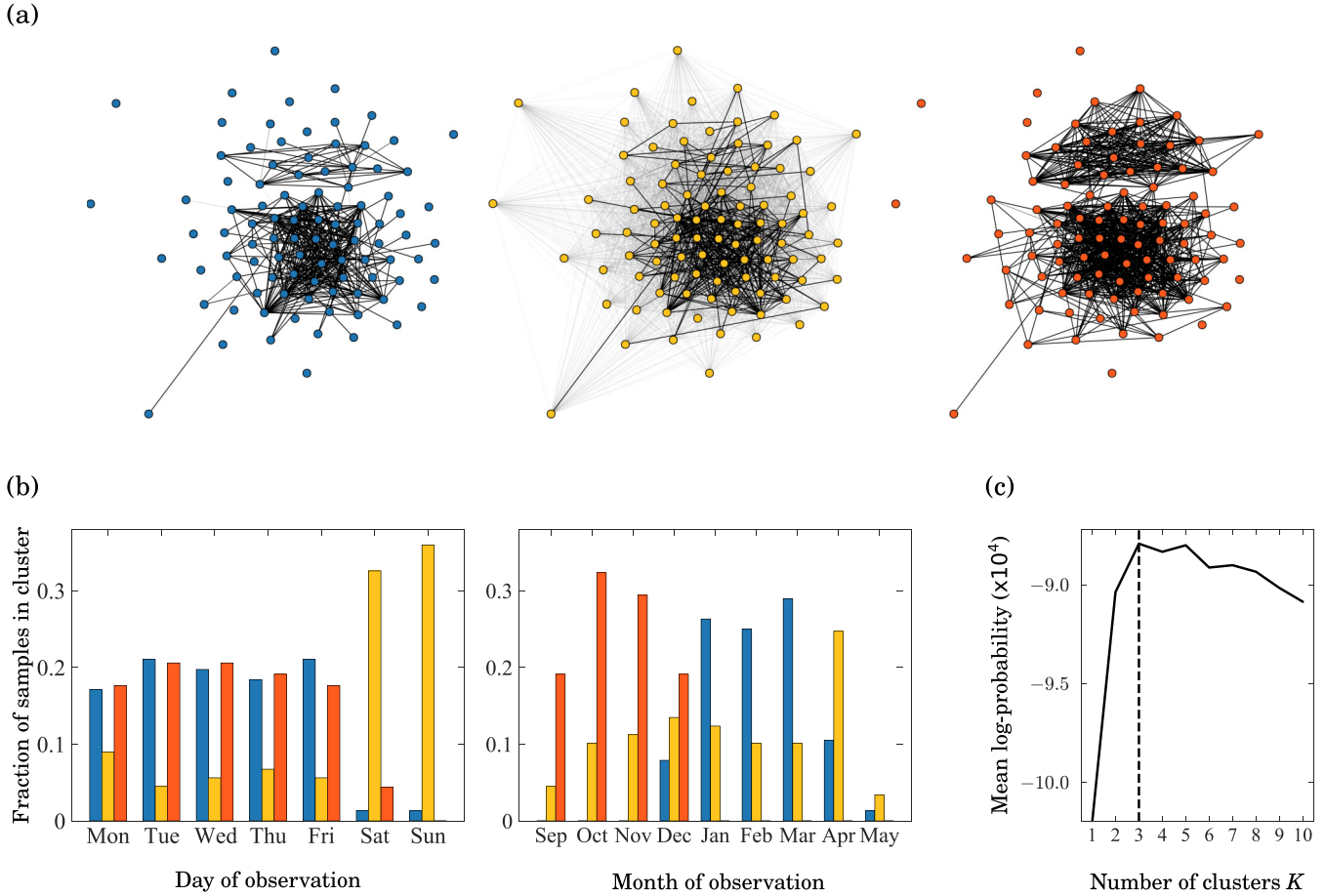
FIG. 3. Results for the reality mining proximity networks described in the text, in which proximity measurements were binned in day-long intervals. (a) Modal networks $\hat{\mathscr{A}}$ with edge widths proportional to the estimates $\hat{A}_{ij}^{(u)}$. (b) The fraction of networks in each cluster that were sampled on a particular day of the week (left) and during a particular month (right). (c) Average posterior log-probability [Eq. (10)] across Gibbs samples as a function of the number of clusters K.

regime, but is also able to tell us that there is no signal when p is large. Note that the true values of the parameters are not needed for computing the KL divergence, so this approach is applicable in experimental settings where the modes are unknown.

Taken together, the results of Fig. 2 tell us that our Gibbs sampler is capable of correctly inferring all of the model parameters so long as there is well-defined cluster structure in the data (something we can determine empirically with the KL divergence), and that inference becomes more reliable as the size of the network population grows. In the following section we demonstrate that our estimation procedure can also successfully cluster populations of real networks and identify multiple distinct modes under real-world conditions.

B. Social network

As our second example we study a network of physical proximity measurements among a group of college students and faculty in the "reality mining" study of Eagle and Pentland [39]. Over a nine-month period, study participants carried mobile phones equipped with special software that recorded when they were in close proximity with one another using Bluetooth radio technology. From these data we construct networks in which an edge between two participants

indicates that they were observed in proximity at least once on a given day. The result is one network for each of the 232 days of the study where at least one interaction happened, all with $n = 96$ nodes, and between 1 and 418 edges.

The results of applying our Gibbs sampling algorithm with $S = 1000$ samples to this population of networks are shown in Fig. 3. To select the number of clusters K, we run the algorithm at multiple values of K, identifying the value for which the average posterior probability is maximized as the optimal K. In Fig. 3(a) of the figure we display the three very distinct modes $\hat{\mathscr{A}}$ found for this population, with the edge widths proportional to the inferred values of $\hat{A}_{ij}^{(u)}$. In Fig. 3(b) we show histograms of the fractions of networks in each cluster that were sampled on a given day of the week (left) and during a given month (right). We can see clear separation in these histograms, indicating that the modes correspond roughly to weekends (yellow), weekdays during the fall semester (orange), and weekdays during the spring semester (blue). We see a clearly distinct structure in the networks during semester weekdays versus weekends, with a denser network observed in the first semester than the second. However, the weekend network mode contains a wider variety of edges with nonnegligible weight. These observations are consistent with a typical university course schedule in which

students associate primarily with classmates during classes but with a wider selection of acquaintances outside of work. In Fig. 3(c), we show the mean log-probability of the Gibbs samples as a function of the number of modes K, which can we see has a peak at $K = 3$ (the value chosen for our posterior estimates).

We note that the modes have significantly higher density than the individual observed networks with their corresponding clusters, as demonstrated by the high false negative rates $1 - \alpha_u$ and low false-positive rates β_u. This indicates that the set of likely edges in each mode is much larger than the number of interactions observed in any single time window. This is not surprising—most people do not see all of their acquaintances every day. Nonetheless, it emphasizes the ability of our method to infer a plausible set of "true" connections from very noisy individual network measurements, a task that would not be easy to do by clustering networks in a naive way.

V. CONCLUSIONS

In this paper we have described a method for statistical analysis of heterogeneous network populations, as seen in applications involving repeated or longitudinal measurements of a single network, or observation of a fixed set of nodes across multiple study subjects. We have proposed a generative model for heterogeneous networks in which networks divide into a number of clusters or modes and derived an exact Gibbs sampling procedure for sampling the resulting posterior distributions. We have demonstrated our model and estimation methods first on synthetic network data, finding that parameter recovery with the Gibbs sampler is possible as long as variability within the clusters of networks is not too high. We then analyze a real-world network population from a longitudinal proximity network study and find that this population is best described by three underlying mode networks rather than a single ground-truth network as is assumed in most network reconstruction approaches. Our model is capable of encompassing a greater variety of network data than such unimodal approaches and provides a natural framework for simultaneously clustering and denoising a set of networks.

The framework we describe could be extended in a number of ways. For the sake of simplicity we have here assumed that our networks are undirected, unweighted, and that that all edges in a given mode have identical error rates. There are other possibilities, however, including adaptations for directed or weighted edges, more complex noise models such as ones with individual error rates for each node or edge, or the use of a parametric model for the mode networks to allow for simultaneous inference of large-scale structure such as communities. Additionally, one could employ fast algorithms to fit the model without any sampling whatsoever, by inferring point estimates using methods such as k-means clustering after removing the dependence on $\boldsymbol{\theta}$ through profiling or marginalization. We leave exploration of these avenues for future work.

Note added. After this work was completed we learned of recent related work by Mantziou *et al.* A preprint can be found at Ref. [40].

APPENDIX A: METRIC INFERENCE

One approach to the problem of network clustering and denoising is to employ methods of *metric inference* [18], in which network samples live in a metric space and are clustered based on their distance in that space. It turns out that the simplest metric model of graphs, as used in Refs. [18,19,31], corresponds to a special case of the model analyzed in the present work. Here we demonstrate this correspondence.

Starting with the unimodal case, the likelihood of a single network sample under a generic metric model is given by [18]

$$P(\boldsymbol{D}^{(t)}|\boldsymbol{A}, \sigma) \propto f_\sigma[d(\boldsymbol{D}^{(t)}, \boldsymbol{A})], \tag{A1}$$

where σ is a parameter that controls an analog of the variance, $f_\sigma(\cdot)$ is a nonincreasing function, and $d(\cdot, \cdot)$ is a distance metric on the space of all labeled graphs on n nodes. This model is very flexible in principle but less so in practice, because computational issues place constraints on the possible choices of metric and the function f_σ. Most authors use an exponential function $f_\sigma(x) = e^{-x/\sigma}$ and the Hamming distance [18,19]

$$d_H(\boldsymbol{D}^{(t)}, \boldsymbol{A}) = \sum_{i<j} \left[\left(1 - D_{ij}^{(t)} \right) A_{ij} + D_{ij}^{(t)} (1 - A_{ij}) \right].$$

One can then use the properties of the exponential to derive the simple data likelihood

$$P(\mathscr{D}|\boldsymbol{A}, \sigma) = \prod_{t=1}^{N} \prod_{i<j} \left[\frac{e^{-\frac{1}{\sigma} D_{ij}^t}}{1 + e^{-\frac{1}{\sigma}}} \right]^{(1-A_{ij})} \left[\frac{e^{-\frac{1}{\sigma}(1-D_{ij}^t)}}{1 + e^{-\frac{1}{\sigma}}} \right]^{A_{ij}}$$

$$= \prod_{i<j} \left[\frac{e^{-\frac{1}{\sigma} X_{ij}}}{(1 + e^{-\frac{1}{\sigma}})^N} \right]^{(1-A_{ij})} \left[\frac{e^{-\frac{1}{\sigma}(N-X_{ij})}}{(1 + e^{-\frac{1}{\sigma}})^N} \right]^{A_{ij}}, \tag{A2}$$

which is straightforward to evaluate and simulate numerically.

Comparing with Eq. (2) of the main text, we see that this model is equivalent to our model with true- and false-positive rates if

$$\frac{e^{-\frac{1}{\sigma}(N-X_{ij})}}{(1 + e^{-\frac{1}{\sigma}})^N} = \alpha^{X_{ij}} (1 - \alpha)^{N - X_{ij}},$$

$$\frac{e^{-\frac{1}{\sigma} X_{ij}}}{(1 + e^{-\frac{1}{\sigma}})^N} = \beta^{X_{ij}} (1 - \beta)^{N - X_{ij}}. \tag{A3}$$

In particular, if we set $\alpha = 1 - \beta$, then we can map the two models directly, with

$$\sigma = \left[\log \frac{1 - \beta}{\beta} \right]^{-1}, \tag{A4}$$

so long as $\beta < 1/2$. In other words, the metric distance approach using the exponential function and Hamming distance

TABLE I. Variables used in the analysis and algorithm, as defined in terms of the basic quantities $\mathbf{Z}$, $\mathscr{D}$, and $\mathscr{A}$.

Quantity	Definition	Expression
N	Number of network samples	$N = \sum_{tu} z_{tu}$
N_u	Number of network samples in cluster for mode u	$N_u = \sum_t z_{tu}$
M_t	Edges in network sample t	$M_t = \sum_{ij} D_{ij}^{(t)}$
M_u^*	Edges in mode u	$M_u^* = \sum_{ij} A_{ij}^{(u)}$
M^*	Edges in all modes	$M^* = \sum_{iju} A_{ij}^{(u)}$
X_{ij}^u	Number of times i, j are connected in network samples from mode u	$X_{ij}^u = \sum_t D_{ij}^{(t)} z_{tu}$
Y_{tu}^{00}	Edges absent in sample t and absent in mode u ("true negatives")	$Y_{tu}^{00} = \sum_{ij} \left(1 - D_{ij}^{(t)}\right)\left(1 - A_{ij}^{(u)}\right)$
Y_{tu}^{10}	Edges present in sample t and absent in mode u ("false positives")	$Y_{tu}^{10} = \sum_{ij} D_{ij}^{(t)}\left(1 - A_{ij}^{(u)}\right)$
Y_{tu}^{01}	Edges absent in sample t and present in mode u ("false negatives")	$Y_{tu}^{01} = \sum_{ij} \left(1 - D_{ij}^{(t)}\right)A_{ij}^{(u)}$
Y_{tu}^{11}	Edges present in sample t and present in mode u ("true positives")	$Y_{tu}^{11} = \sum_{ij} D_{ij}^{(t)} A_{ij}^{(u)}$
W_u^{00}	Total true negatives for all samples in cluster for mode u	$W_u^{00} = \sum_{ijt} \left(1 - D_{ij}^{(t)}\right)\left(1 - A_{ij}^{(u)}\right) z_{tu}$
W_u^{10}	Total false positives for all samples in cluster for mode u	$W_u^{10} = \sum_{ijt} D_{ij}^{(t)}\left(1 - A_{ij}^{(u)}\right) z_{tu}$
W_u^{01}	Total false negatives for all samples in cluster for mode u	$W_u^{01} = \sum_{ijt} \left(1 - D_{ij}^{(t)}\right)A_{ij}^{(u)} z_{tu}$
W_u^{11}	Total true positives for all samples in cluster for mode u	$W_u^{11} = \sum_{ijt} D_{ij}^{(t)} A_{ij}^{(u)} z_{tu}$

[18,19] is equivalent to assuming a binomial model for edge measurements, but with a particular relationship between false-positive and true-positive rates, $\alpha = 1 - \beta$, which is generally unrealistic [7,12].

It is straightforward to verify that analogous results hold in the multimodal case. One finds that when $\alpha_u = 1 - \beta_u$, one can define σ_u analogously to Eq. (A4) to obtain a single parameter controlling the probability that existing edges from mode u are missed in the samples and that nonexistent edges are added. Thus, we conclude that metric models built with the exponential function and Hamming distance are less expressive than the generative process of network measurements considered here, and we recommend using the latter. If one does wish to use a metric model, then the algorithms presented in the main text can be used to fit it by setting $\alpha_u = 1 - \beta_u$ for all classes u.

APPENDIX B: CONJUGATE PRIORS

For ease of presentation, we have used flat priors for the parameters $\boldsymbol{\theta}$ in the main text, but one can easily replace them with more flexible priors thus:

$$P\big(\alpha_u | H_u^{11}, H_u^{01}\big) = \frac{\alpha_u^{H_u^{11}-1}(1 - \alpha_u)^{H_u^{01}-1}}{B\big(H_u^{11}, H_u^{01}\big)}, \qquad \text{(B1)}$$

$$P\big(\beta_u | H_u^{00}, H_u^{10}\big) = \frac{\beta_u^{H_u^{01}-1}(1 - \beta_u)^{H_u^{00}-1}}{B\big(H_u^{01}, H_u^{00}\big)}, \qquad \text{(B2)}$$

$$P(\rho | a^*, b^*) = \frac{\rho^{a^*-1}(1 - \rho)^{b^*-1}}{B(a^*, b^*)}, \qquad \text{(B3)}$$

$$P(\boldsymbol{\pi} | \boldsymbol{\gamma}) = \frac{\prod_{u=1}^K \pi_u^{\gamma_u-1}}{B(\boldsymbol{\gamma})}, \qquad \text{(B4)}$$

where $B(x, y)$ is the Euler beta function $B(x, y) = \Gamma(x)\Gamma(y)/\Gamma(x + y)$ and $B(\boldsymbol{\gamma})$ is its generalization $B(\boldsymbol{\gamma}) = \prod_{u=1}^K \Gamma(\gamma_u)/\Gamma(\sum_u \gamma_u)$, and where the parameters of these distributions (such as H_u^{11} and H_u^{01}) are new hyperparameters of the model.

Since the priors above are conjugate [41], our Gibbs sampling algorithm carries over with little modification. The only change needed is to the sampling distribution for the parameters:

$$P(\boldsymbol{\theta} | \mathscr{D}, \mathbf{Z}, \mathscr{A}) \propto \rho^{M^*+a^*}(1 - \rho)^{K\binom{N}{2}-M^*+b^*}$$

$$\times \prod_{u=1}^K \pi_u^{N_u+\gamma_u}$$

$$\times \prod_{u=1}^K (\alpha_u)^{W_u^{11}+H_u^{11}}(1 - \alpha_u)^{W_u^{01}+H_u^{01}}$$

$$\times \prod_{u=1}^K (\beta_u)^{W_u^{10}+H_u^{10}}(1 - \beta_u)^{W_u^{00}+H_u^{00}}. \quad \text{(B5)}$$

The parameters of the conjugate priors are analogous to the combinatorial quantities of Table I, and they thus act as pseudocounts of edges, missing edges, and so on.

[1] N. Eagle, A. S. Pentland, and D. Lazer, Inferring friendship network structure by using mobile phone data, Proc. Natl. Acad. Sci. USA **106**, 15274 (2009).

[2] C. T. Butts, Network inference, error, and informant (in)accuracy: A Bayesian approach, Soc. Netw. **25**, 103 (2003).

[3] O. Sporns, *Networks of the Brain* (MIT Press, Cambridge, MA, 2010).

[4] S. E. Fienberg, M. M. Meyer, and S. S. Wasserman, Statistical analysis of multiple sociometric relations, J. Am. Stat. Assoc. **80**, 51 (1985).

[5] C. E. Priebe, D. L. Sussman, M. Tang, and J. T. Vogelstein, Statistical inference on errorfully observed graphs, J. Comput. Graph. Stat. **24**, 930 (2015).

[6] M. E. J. Newman, Network structure from rich but noisy data, Nat. Phys. **14**, 542 (2018).

[7] M. E. J. Newman, Estimating network structure from unreliable measurements, Phys. Rev. E **98**, 062321 (2018).

[8] T. P. Peixoto, Reconstructing Networks with Unknown and Heterogeneous Errors, Phys. Rev. X **8**, 041011 (2018).

[9] C. M. Le, K. Levin, and E. Levina, Estimating a network from multiple noisy realizations, Electron. J. Stat. **12**, 4697 (2018).

[10] R. Tang *et al.*, Connectome smoothing via low-rank approximations, IEEE Trans. Med. Imaging. **38**, 1446 (2018).

[11] L. Wang, Z. Zhang, and D. Dunson, Common and individual structure of brain networks, Ann. Appl. Stat. **13**, 85 (2019).

[12] J.-G. Young, G. T. Cantwell, and M. E. J. Newman, Bayesian inference of network structure from unreliable data, J. Complex Netw. **8**, CNAA046 (2020).

[13] J.-G. Young, F. S. Valdovinos, and M. E. J. Newman, Reconstruction of plant–pollinator networks from observational data, Nat. Commun. **12**, 3911 (2021).

[14] D. S. Bassett, C. H. Xia, and T. D. Satterthwaite, Understanding the emergence of neuropsychiatric disorders with network neuroscience, Biol. Psychiatry: Cogn. Neurosci. Neuroimaging **3**, 742 (2018).

[15] D. M. Titterington, A. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*, (Wiley, New York, NY, 1985).

[16] G. J. McLachlan and D. Peel, *Finite Mixture Models* (Wiley, New York, NY, 2004).

[17] D. A. Kenny and L. La Voie, The social relations model, Adv. Exp. Soc. Psychol. **18**, 141 (1984).

[18] D. Banks and K. Carley, Metric inference for social networks, J. Classif. **11**, 121 (1994).

[19] S. Lunagómez, S. C. Olhede, and P. J. Wolfe, Modeling network populations via graph distances, J. Am. Stat. Assoc. **116**, 2023 (2020).

[20] N. Josephs, W. Li, and E. D. Kolaczyk, Network recovery from unlabeled noisy samples, arXiv:2104.14952 (2021).

[21] D. Durante, D. B. Dunson, and J. T. Vogelstein, Nonparametric Bayes modeling of populations of networks, J. Am. Stat. Assoc. **112**, 1516 (2017).

[22] A. M. Nielsen and D. Witten, The multiple random dot product graph model, arXiv:1811.12172 (2018).

[23] S. Wang, J. Arroyo, J. T. Vogelstein, and C. E. Priebe, Joint embedding of graphs, IEEE Trans. Pattern Anal. Mach. Intell. **43**, 1324 (2019).

[24] J. Arroyo, A. Athreya, J. Cape, G. Chen, C. E. Priebe, and J. T. Vogelstein, Inference for multiple heterogeneous networks with a common invariant subspace, J. Mach. Learn. Res. **22**, 1 (2021).

[25] F. Yin, W. Shen, and C. T. Butts, Finite mixtures of ERGMs for ensembles of networks, Bayesian Anal., 1 (2022).

[26] M. Signorelli and E. C. Wit, Model-based clustering for populations of networks, Stat. Model. **20**, 9 (2020).

[27] L. Peel and A. Clauset, Detecting change points in the large-scale structure of evolving networks, in *Proceedings of the 29th International Conference on Artificial Intelligence (AAAI)* (ACM, New York, NY, 2015), p. 2914.

[28] T. P. Peixoto and L. Gauvin, Change points, memory and epidemic spreading in temporal networks, Sci. Rep. **8**, 15511 (2018).

[29] T. P. Peixoto, Inferring the mesoscale structure of layered, edge-valued, and time-varying networks, Phys. Rev. E **92**, 042807 (2015).

[30] N. Stanley, S. Shai, D. Taylor, and P. J. Mucha, Clustering network layers with the strata multilayer stochastic block model, IEEE Trans. Netw. Sci. Eng. **3**, 95 (2016).

[31] P. S. La Rosa, T. L. Brooks, E. Deych, B. Shands, F. Prior, L. J. Larson-Prior, and W. D. Shannon, Gibbs distribution for statistical analysis of graphical data with a sample application to fcMRI brain images, Stat. Med. **35**, 566 (2016).

[32] J. Bento and S. Ioannidis, A family of tractable graph metrics, Appl. Netw. Sci. **4**, 107 (2019).

[33] D. S. Bassett and O. Sporns, Network neuroscience, Nat. Neurosci. **20**, 353 (2017).

[34] J. Stehlé, N. Voirin, A. Barrat, C. Cattuto, L. Isella, J.-F. Pinton, M. Quaggiotto, W. Van den Broeck, C. Régis, B. Lina *et al.*, High-resolution measurements of face-to-face contact patterns in a primary school, PLoS One **6**, e23176 (2011).

[35] J. B. Kruskal and M. Wish, Multidimensional scaling, in *Sage University Paper series on Quantitative Applications in Social Sciences, 07-011* (Sage Publications, Beverly Hills, CA, 1978).

[36] R. M. Neal, Markov chain sampling methods for Dirichlet process mixture models, J. Comput. Graph. Stat. **9**, 249 (2000).

[37] S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-6**, 721 (1984).

[38] M. Meilă, Comparing clusterings: An information based distance, J. Multivar. Anal. **98**, 873 (2007).

[39] N. Eagle and A. Pentland, Reality mining: Sensing complex social systems, J. Pers. Ubiquitous Comput. **10**, 255 (2006).

[40] A. Mantziou, S. Lunagómez, and R. Mitra, Bayesian model-based clustering for multiple network data, arXiv:2107.03431 (2021).

[41] A. Gelman, H. S. Stern, J. B. Carlin, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*, 3rd ed. (Chapman and Hall/CRC, New York, NY, 2013).