Bayesian Analysis (2023) **TBA**, Number TBA, pp. 1–27

Hierarchical Stochastic Block Model for Community Detection in Multiplex Networks*

Arash Amini†, Marina Paez‡ and Lizhen Lin§

Abstract. Multiplex networks have become increasingly more prevalent in many fields, and have emerged as a powerful tool for modeling the complexity of real networks. There is a critical need for developing inference models for multiplex networks that can take into account potential dependencies across different layers, particularly when the aim is community detection. We add to a limited literature by proposing a novel and efficient Bayesian model for community detection in multiplex networks. A key feature of our approach is the ability to model varying communities at different network layers. In contrast, many existing models assume the same communities for all layers. Moreover, our model automatically picks up the necessary number of communities at each layer (as validated by real data examples). This is appealing, since deciding the number of communities is a challenging aspect of community detection, and especially so in the multiplex setting, if one allows the communities to change across layers. Borrowing ideas from hierarchical Bayesian modeling, we use a hierarchical Dirichlet prior to model community labels across layers, allowing dependency in their structure. Given the community labels, a stochastic block model (SBM) is assumed for each layer. We develop an efficient slice sampler for sampling the posterior distribution of the community labels as well as the link probabilities between communities. In doing so, we address some unique challenges posed by coupling the complex likelihood of SBM with the hierarchical nature of the prior on the labels. An extensive empirical validation is performed on simulated and real data, demonstrating the superior performance of the model over single-layer alternatives, as well as the ability to uncover interesting structures in real networks.

Keywords: community detection, hierarchical stochastic block model (HSBM), multiplex networks, hierarchical Dirichlet process, random partition.

1 Introduction

Networks, which are used to model interactions among a set of entities, have emerged as one of the most powerful tools for modern data analysis. The last few decades have witnessed explosions in the development of models, theory, and algorithms for network analysis. Modern network data are often complex and heterogeneous. To model such heterogeneity, *multiplex networks* (Boccaletti et al., 2014; Kivelä et al., 2014), which

*The contribution of L. Lin was funded by NSF grants DMS 1654579, DMS 2113642 and a DARPA grant N66001-17-1-4041. A. Amini was partially supported by the NSF grant DMS-1945667.

†Department of Statistics, The University of California at Los Angeles, USA, aaamini@ucla.edu

‡Department of Statistical Methods, Federal University of Rio de Janeiro, Brazil, marina@im.ufrj.br

§Department of Applied and Computational Mathematics and Statistics, The University of Notre Dame, USA, lizhen.lin@nd.edu

 https://doi.org/10.1214/22-BA1355

also go by various other qualifiers (multiple-layer, multiple-slice, multi-array, multi-relational), have arisen as a useful representation of complex networks.

A multiplex network typically consists of a fixed set of nodes but multiple types of edges, often representing heterogeneous relationships as well as the dynamic nature of the edges. These networks have become increasingly prevalent in many applications. Typical examples include various types of dynamic networks (Berlingerio et al., 2013; Majdandzic et al., 2014) including temporal networks, dynamic social networks (Carley et al., 2009) and dynamic biological networks such as protein networks (Wang et al., 2014). An example of a multiplex social network is the Twitter network where different edge information is available such as "mention", "follow" and "retweet" (Greene and Cunningham, 2013). Another example is the co-authorship network where the authors are recorded on relationships such as "co-publishes", "convenes" and "cites" (Hmimida and Kanawati, 2015). Many other examples can be found in economics (Poledna et al., 2015), neuroscience (De Domenico et al., 2016), biology (Didier et al., 2015) and so on.

Due to the ubiquity of multiplex network data, there is a critical need for developing realistic statistical models that can handle their complex structures. There has already been growing efforts in extending static (or single-layered) statistical network models to the multiplex setting (Mucha et al., 2010). Many statistical metrics have already been extended from basic notions such as degree and node centrality (Bródka et al., 2011; Battiston et al., 2014) to the clustering coefficient and modularity (Cozzo et al., 2013; Blondel et al., 2008). Popular latent space models (Raftery et al., 2002) have also been extended to dynamic (Sarkar and Moore, 2005) and multiplex (Gollini and Murphy, 2016; Salter-Townshend and McCormick, 2017) networks. Based on such models, one can perform learning tasks such as link probability estimation or link prediction.

Another task that has been extensively studied in single-layer networks is community detection, that is, clustering nodes into groups or communities. There have already been a few works proposing models or algorithms for community detection in multiplex and multilayer networks (Mucha et al., 2010; Berlingerio et al., 2011; Kuncheva and Montana, 2015; Wilson et al., 2017; De Bacco et al., 2017; Bhattacharyya and Chatterjee, 2018). The general strategy adopted is to either transform a multiplex network into a single-layer and then apply the existing community detection algorithms, or extend a model from static networks to the multiplex setting. One of the key shortcomings of many existing methods is that communities across different layers are assumed to be the same, which is clearly restrictive and often unrealistic. Instead, there is interest in monitoring or exploring how the communities vary across different layers. Note that there is a literature on community detection for dynamic or temporal networks mainly through developing a dynamic stochastic block model or a dynamic spectral clustering algorithm, see e.g., Liu et al. (2018), Pensky and Zhang (2019) and Matias and Miele (2017), which typically require smoothing networks over time thus assuming networks are observed over a significant number of time points.

We propose a novel Bayesian community detection model, namely, the hierarchical stochastic block model (HSBM) for multiplex networks that is fundamentally different from the existing approaches. Specifically, we impose a random partition prior based on the hierarchical Dirichlet process (Teh et al., 2006) on communities (or partitions)

across different layers. Given these communities, a stochastic block model is assumed for each layer. One of the appealing features of our model is allowing the communities to vary across different layers of the network while being able to incorporate potential dependency across them. The hierarchical aspect of HSBM allows for effective borrowing of information or strength across different layers for improved estimation. Our approach has the added advantage of being able to handle a broad class of networks by allowing the number of nodes to vary, and not necessarily imposing the nodes to be fixed across layers. In addition, HSBM inherits the desirable property of hierarchical Dirichlet priors that allow for automatic and adaptive selection of the number of communities (for each layer) based on the data.

Our simulation study in Section 4 confirms that HSBM significantly outperforms a model that assumes independence of layers, especially when it comes to matching community labels across layers. In particular, the superior performance of our model over its independent single-layer counterpart is manifested by the significant improvement in the slicewise or aggregate NMI (Normalized Mutual Information) measures.

Although a hierarchical Dirichlet process is a natural choice for modeling dependent partition structures, extending the ideas from simple mixture models to community structured network models is not that straightforward. In particular, leveraging ideas from one of our recent work (Amini et al., 2019), we develop a new and efficient slice sampler for exact sampling of the posterior of HSBM for inference. We will discuss some of the technical difficulties in Remark 1.

The rest of the paper is organized as follow. Section 2 introduces the HSBM in details. Section 3 is devoted to describing a novel MCMC algorithm for inference of HSBM. Simulation study and real data analysis are presented in Sections 4 and 5. The code for all the experiments is available at Amini et al. (2021). We conclude with a discussion in Section 6.

2 Hierarchical stochastic block model (HSBM)

Consider a multiplex network with T layers (or T types of edge relations) and n_t nodes with labels in $[n_t] = \{1, \dots, n_t\}$ for each layer $t = 1, \dots, T$. Denote by $\boldsymbol{A}_t$ the adjacency matrix of the network at layer t, so that an observed multiplex network consists of the collection $\boldsymbol{A}_t \in \{0,1\}^{n_t \times n_t}$ for $t = 1, \dots, T$. We let $\boldsymbol{A}$ denote this collection and view it as a partial (or irregular) adjacency tensor. That is, $\boldsymbol{A} = (A_{tij}, t \in [T], i, j \in [n_t])$ and $\boldsymbol{A}_t = (A_{tij}, i, j \in [n_t])$ where $A_{tij} = 1$ if nodes i and j in layer t are connected. Our goal is to estimate the clustering or community structure of the nodes in each layer, given $\boldsymbol{A}$.

Specifically, to each node i, at each layer t, we assign a community label, encoded in a variable $\boldsymbol{Z} = (z_{ti}) \in \mathbb{N}^{T \times n_t}$, where $\mathbb{N} = \{1, 2, \dots, \}$ is the set of natural numbers. Let

$$\boldsymbol{\eta}_t = (\eta_{txy})_{x,y} \in [0,1]^{\mathbb{N} \times \mathbb{N}}, \quad \eta_{txy} \equiv \eta_t(x,y), \tag{2.1}$$

be a matrix of link probabilities between communities, indexed by $\mathbb{N}^2$, for $t = 1, \dots, T$. At times, we will use the equivalent notation $\eta_t(x,y) \equiv \eta_{txy}$ to increase readability.

For example, we interpret $\eta_{t12} = \eta_t(1,2)$ as the probability of an edge being formed between a node from community 1 and a node from community 2 at layer t. Note that we have assumed that the total number of community labels is infinite. However, for a given adjacency matrix $\boldsymbol{A}_t$ observed at layer t, the number of community labels is finite, unknown, and will be denoted by K_t.

For each layer t, we model the distribution of the adjacency matrix $\boldsymbol{A}_t \in \{0,1\}^{n_t \times n_t}$ as a stochastic block model (SBM) with membership vector $\boldsymbol{z}_t = (z_{ti}) \in \mathbb{N}^{n_t}$, and edge probability matrix $\boldsymbol{\eta}_t$, that is,

$$\boldsymbol{A}_t \mid \boldsymbol{z}_t, \boldsymbol{\eta}_t, \sim \mathsf{SBM}(\boldsymbol{z}_t; \boldsymbol{\eta}_t) \iff A_{tij} \overset{\text{iid}}{\sim} \mathsf{Ber}\big(\, \eta_t(z_{ti}, z_{tj}) \,\big), \quad 1 \le i < j \le n_t. \tag{2.2}$$

In an SBM, the link probability between nodes i and j is uniquely determined by which communities these nodes belong to. In our notation, at a layer t, the link probability between nodes i and j is given by $\eta_t(z_{ti}, z_{tj})$. Note that our SBM notation is slightly different from the traditional stochastic block model where the set of community labels is random. In writing $\mathsf{SBM}(\boldsymbol{z}; \boldsymbol{\eta})$, we assume that $\boldsymbol{z}$ is given and nonrandom.

If there is belief or prior information that the community structure in one layer of the network is independent of the others, independent stochastic block models (SBM) could be assumed for the $\boldsymbol{A}_t$'s. This assumption is, however, too restrictive when we are dealing with networks of different kinds of relations but among the same set of nodes; or with different sets of nodes but similar types of relations. The other extreme is to assume that all layers in the network share the same partition—meaning that $\boldsymbol{z}_t = \boldsymbol{z}$ for all t and the $\boldsymbol{A}_t$'s are conditionally independent given $\boldsymbol{z}$.

We believe, however, that a model that can incorporate various dependencies between these two extremes is more appropriate in many applications. In other words, it may be desirable to allow some change in the partition structure between layers, but also impose some kind of dependence among them. Here, we propose a model that achieves this goal by allowing the community structures at various layers to be different but dependent, using a hierarchical specification for the distribution of the partitions.

2.1 Hierarchical community prior

Before stating the prior on the labels, let us introduce a simplification, namely, that $\boldsymbol{\eta}_t$ does not vary by layer. Therefore, we drop index t, and denote the matrix of link probabilities by $\boldsymbol{\eta}$ and its elements by $\eta_{xy} \equiv \eta(x,y)$. This assumption is not necessarily restrictive since, as will become clear shortly, our model allows for an infinite number of communities. By imposing this restriction, we are simply stating that cluster i at a certain layer corresponds to the same cluster i at every other layer. We then assume independent Beta prior distributions for each element of $\boldsymbol{\eta}$, given by:

$$\boldsymbol{\eta} = (\eta_{xy}) \overset{\text{iid}}{\sim} \mathsf{Beta}(\alpha_\eta, \beta_\eta), \tag{2.3}$$

where Beta stands for the usual Beta distribution.

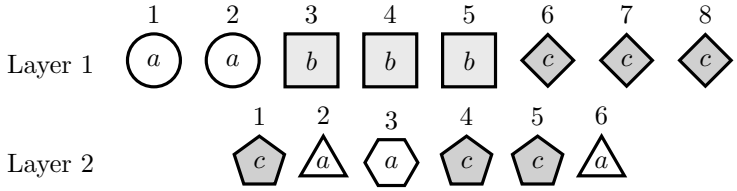


Figure 1: Toy example of the nodes of a two-layer network. The numbers are the indices of the nodes in each layer. The shape of a node represents its group and the letter symbol inside represents the community it belongs to.

Our key idea is to impose a dependent random partition prior on the membership labels at different layers, i.e., $\boldsymbol{z}_t, t \in [T]$. We adopt the prior based on a hierarchical Dirichlet process (HDP).

We assume that the reader is familiar with the HDP and its various interpretations, and in particular, the Chinese Restaurant Franchise process (CRF); see for example (Teh et al., 2006). In the CRF interpretation, each layer t of the network corresponds to a restaurant, and nodes are gathered around *tables*, or *groups*, in each restaurant. Let g_{ti} denote the group of node i in restaurant (i.e., layer) t. All the nodes in the same group g, at the same layer t, share the same dish (i.e., community) which is denoted by k_{tg}. Thus, given all the groups $\boldsymbol{g}_t = (g_{ti})_i$ and *group-communities* $\boldsymbol{k}_t = (k_{tg})_g$, the label of node i (at layer t) is uniquely determined:

$$z_{ti} \,|\, \boldsymbol{g}_t, \boldsymbol{k}_t = k_{t,g_{ti}}. \tag{2.4}$$

To help the reader understand the notation of the label part of the model, we present a simple toy example:

Example 2.1. *Consider a two-layer network with $n_1 = 8$ and $n_2 = 6$ nodes. At each layer (restaurant), the nodes are presented in a line, and are numbered from left to right, as illustrated in Figure 1. The shape of each node represents the group (table) it belongs to and the fill, as well as its label $\in \{a, b, c\}$, represents the community (dish). For example, in layer 1, nodes 1 and 2 are grouped together, represented by a circle, and similarly for $\{3, 4, 5\}$ and $\{6, 7, 8\}$ represented by square and diamond, respectively. In this first layer, each group is associated with a different community. In layer 2, we have the following groups: $\{1, 4, 5\}$, $\{2, 6\}$ and $\{3\}$, represented by a pentagon, triangle and hexagon, respectively. Also, note that the groups represented by a triangle and a hexagon share the same community (dish). In general, groups in the same layer or in two different layers can be linked via their group-community assignment, which is encoded by k_{tg}. For example, the circular group in the first layer shares the same community (dish) with the triangular and hexagonal groups in the second layer. This information is carried through the definition of k_{tg}: According to Figure 1,*

$$k_{1,\bigcirc} = k_{2,\triangle} = k_{2,\hexagon} = a.$$

This, in turn, implies that all the nodes in these groups (nodes $\{1,2\}$ *in layer 1 and nodes* $\{2,6\} \cup \{3\}$ *in layer 2) share the same community, that is, we have*

$$z_{11} = z_{12} = z_{22} = z_{26} = z_{23} = a.$$

As an example of how (2.4) *is used to determine the node community labels, we have* $g_{26} = \vartriangle$ *and* $k_{2,\vartriangle} = a$*, hence* $z_{26} = k_{2,g_{26}} = a$*.*

A prior on z_{ti} can be obtained via priors on k_{tg} and g_{ti}. To impose dependence among layers, we assume that group-communities $\boldsymbol{k}_t$, across layers t, are drawn from the same prior:

$$k_{tg} \mid \boldsymbol{\pi} \sim \boldsymbol{\pi}, \quad g \in \mathbb{N}, \tag{2.5}$$

$$g_{ti} \mid \boldsymbol{\gamma}_t \sim \boldsymbol{\gamma}_t, \quad i = 1, \ldots, n_t, \tag{2.6}$$

where $\boldsymbol{\gamma}_t, t = 1, \ldots, T$ are categorical distributions with infinite categories, with the weight of each category representing the fraction of the nodes in layer t that would end up in group g (eventually).

To complete the prior specification, we impose

$$\boldsymbol{\pi} \mid \gamma_0 \sim \mathsf{GEM}(\gamma_0), \quad \boldsymbol{\pi} = (\pi_k) \tag{2.7}$$

$$\boldsymbol{\gamma}_t \mid \alpha_0 \sim \mathsf{GEM}(\alpha_0), \quad \boldsymbol{\gamma}_t = (\gamma_{tg}), \tag{2.8}$$

where GEM stands for Griffiths, Engen and McCloskey (Picard and Pitman, 2006); it is a distribution for a random measure on $\mathbb{N}$ which has the well-known stick-breaking construction (Sethuraman, 1994). Equations (2.7), (2.8), (2.5), (2.6) and (2.4) together specify our hierarchical label prior on the node labels $(\boldsymbol{z}_t)_{t=1}^T$.

The hierarchical label prior above, together with prior on $\boldsymbol{\eta}$ in (2.3) and the SBM of (2.2) provide the full specification of the HSBM.

Remark. It turns out that the label prior described above is exactly the label prior implicit in the well-known Hierarchical Dirichlet Process of Teh et al. (2006). Since this equivalence is not central to the model we present, we do not go into further details here and refer the interested reader to Amini et al. (2019).

2.2 Joint distribution

The joint density for the label part of HSBM, that is, equations (2.4)–(2.7), can be expressed as:

$$p(\boldsymbol{g}, \boldsymbol{k}, \boldsymbol{\gamma}', \boldsymbol{\pi}') = \prod_{t=1}^{T} \Big[p(\boldsymbol{g}_t|\boldsymbol{\gamma}_t)\, p(\boldsymbol{\gamma}'_t)\, p(\boldsymbol{k}_t|\boldsymbol{\pi}) \Big] p(\boldsymbol{\pi}'), \tag{2.9}$$

where $p(\boldsymbol{g}_t|\boldsymbol{\gamma}_t) = \prod_{i=1}^{n_t} \gamma_{t,g_{ti}}$ and $p(\boldsymbol{k}_t|\boldsymbol{\pi}) = \prod_{g=1}^{\infty} \pi_{k_{tg}}$. The new variables $\boldsymbol{\gamma}'_t$ and $\boldsymbol{\pi}'$ are related to $\boldsymbol{\gamma}_t$ and $\boldsymbol{\pi}$ via the stick-breaking construction for the GEM distribution (Sethuraman, 1994; Ishwaran and James, 2001). The idea behind this construction

is to imagine a stick with length 1, which will be successively broken into smaller pieces. Let $F : [0,1]^{\mathbb{N}} \to [0,1]^{\mathbb{N}}$ be given by

$$[F(\boldsymbol{x})]_1 := x_1, \quad [F(\boldsymbol{x})]_j := x_j \prod_{\ell=1}^{j-1}(1-x_\ell), \tag{2.10}$$

where $\boldsymbol{x} = (x_j, j \in \mathbb{N})$. Then, $[F(\boldsymbol{x})]_j$ is the length of the piece broken at iteration j after successive fractions $x_1, x_2, \ldots, x_j$ are broken off. Denote

$$b_{\alpha,\beta}(x) := x^{\alpha-1}(1-x)^{\beta-1}$$

which is the density for $\mathsf{Beta}(\alpha,\beta)$ up to a normalization constant. Both $\boldsymbol{\pi}$ and $\boldsymbol{\gamma}_j$ above have stick-breaking representations of this form:

$$\begin{aligned} \gamma'_{tg} &\sim b_{1,\alpha_0}(\cdot), \quad & \pi'_k &\sim b_{1,\gamma_0}(\cdot), \\ \boldsymbol{\gamma}_t &= F(\boldsymbol{\gamma}'_t), & \boldsymbol{\pi} &= F(\boldsymbol{\pi}'), \end{aligned}$$

where $\boldsymbol{\gamma}'_t = (\gamma'_{tg})$ and $\boldsymbol{\pi}' = (\pi'_k)$.

Adding the network part to the label part of the model, we obtain the full joint density of HSBM:

$$\begin{aligned} &p(\boldsymbol{A}, \boldsymbol{\eta}, \boldsymbol{g}, \boldsymbol{k}, \boldsymbol{\gamma}', \boldsymbol{\pi}') = p(\boldsymbol{A} \mid \boldsymbol{g}, \boldsymbol{k}, \boldsymbol{\eta}) \cdot p(\boldsymbol{g}, \boldsymbol{k}, \boldsymbol{\gamma}', \boldsymbol{\pi}') \cdot p(\boldsymbol{\eta}) \\ &= \prod_{t=1}^{T} \left(\prod_{1 \le i < j \le n_t} L\Big(\eta(k_{t,g_{ti}}, k_{t,g_{tj}}); A_{tij}\Big) \prod_{i=1}^{n_t} \gamma_{t,g_{ti}} \prod_{g=1}^{\infty} b_{1,\alpha_0}(\gamma'_{tg}) \prod_{g=1}^{\infty} \pi_{k_{tg}} \right) \times \\ &\quad \prod_{k=1}^{\infty} b_{1,\gamma_0}(\pi'_k) \prod_{1 \le k \le \ell < \infty} b_{\alpha_\eta,\beta_\eta}(\eta_{k\ell}), \end{aligned} \tag{2.11}$$

where

$$L(p;a) := p^a(1-p)^{1-a}$$

is the Bernoulli likelihood Note that we have used the alternative notation $\eta(x,y) = \eta_{xy}$ for readability. In the next section, we derive a novel MCMC algorithm for sampling the posterior distribution of our model.

3 Slice sampling for HSBM

We propose a slice sampler for HSBM, based on a slice sampling algorithm we recently developed for HDP (Amini et al., 2019). Recall that in slice sampling from a density $f(x)$, we introduce the nonnegative variable u, and look at the joint density $g(x,u) = 1\{0 \le u \le f(x)\}$ whose marginal over x is $f(x)$. Then, we perform Gibbs sampling on the joint g. In the end, we only keep samples of x and discard those of u. This idea has been employed in Kalli et al. (2011) to sample from the classical DP mixture and extended in Amini et al. (2019) to sample HDPs.

In order to perform the slice sampling for HSBM, we introduce (independent) variables $\boldsymbol{u} = (u_{ti})$ and $\boldsymbol{v} = (v_{tg})$ so that the *augmented joint density* for (2.9) is

$$p(\boldsymbol{g}, \boldsymbol{k}, \boldsymbol{\gamma}', \boldsymbol{\pi}', \boldsymbol{u}, \boldsymbol{v}) = p(\boldsymbol{g}, \boldsymbol{u} \mid \boldsymbol{\gamma}) \cdot p(\boldsymbol{\gamma}') \cdot p(\boldsymbol{k}, \boldsymbol{v} \mid \boldsymbol{\pi}) \cdot p(\boldsymbol{\pi}'),$$

where for example

$$p(\boldsymbol{g}, \boldsymbol{u} \mid \boldsymbol{\gamma}) = \prod_{t=1}^{T} \prod_{i=1}^{n_t} 1\{0 \le u_{ti} \le \gamma_{t,g_{ti}}\},$$

and similarly for $p(\boldsymbol{k}, \boldsymbol{v} \mid \boldsymbol{\pi})$. Note that marginalizing out $\boldsymbol{u}$ from $p(\boldsymbol{g}, \boldsymbol{u} \mid \boldsymbol{\gamma})$ gives back $p(\boldsymbol{g} \mid \boldsymbol{\gamma}) = \prod_t \prod_i \gamma_{t,g_{ti}}$ as before. The full augmented joint density is now

$$\begin{aligned} p(\boldsymbol{A}, \boldsymbol{\eta}, \boldsymbol{g}, \boldsymbol{k}, \boldsymbol{\gamma}', \boldsymbol{\pi}', \boldsymbol{u}, \boldsymbol{v}) &= p(\boldsymbol{A} \mid \boldsymbol{g}, \boldsymbol{k}, \boldsymbol{\eta}) \cdot p(\boldsymbol{g}, \boldsymbol{k}, \boldsymbol{\gamma}', \boldsymbol{\pi}', \boldsymbol{u}, \boldsymbol{v}) \cdot p(\boldsymbol{\eta}) \\ &= \prod_{t=1}^{T} \Bigg(\prod_{1 \le i < j \le n_t} L\Big(\eta(k_{t,g_{ti}}, k_{t,g_{tj}}); A_{tij}\Big) \prod_{i=1}^{n_t} 1\{u_{ti} \le \gamma_{t,g_{ti}}\} \times \\ &\qquad \prod_{g=1}^{\infty} b_{1,\alpha_0}(\gamma'_{tg}) \prod_{g=1}^{\infty} 1\{v_{tg} \le \pi_{k_{tg}}\} \Bigg) \prod_{k=1}^{\infty} b_{1,\gamma_0}(\pi'_k) \prod_{1 \le k \le \ell < \infty} b_{\alpha_\eta, \beta_\eta}(\eta_{k\ell}), \end{aligned} \tag{3.1}$$

with the support understood to be restricted to $\boldsymbol{u} \ge 0$ and $\boldsymbol{v} \ge 0$. We then perform block Gibbs sampling on the augmented density. Note that marginalizing variables $(\boldsymbol{u}_t)$ and $(\boldsymbol{v}_t)$ out, we get back original joint density (2.11). The idea is to sample $(\boldsymbol{\gamma}', \boldsymbol{u})$ jointly given the rest of the variables, and similarly for $(\boldsymbol{\pi}', \boldsymbol{v})$. The updates for variables $\boldsymbol{u}, \boldsymbol{\gamma}', \boldsymbol{v}$ and $\boldsymbol{\pi}'$ are similar to those in Amini et al. (2019). However, the updates for the underlying latent groups $\boldsymbol{g}$ and group-communities $\boldsymbol{k}$ require some care due to the coupling introduced by the SBM likelihood. As can be seen from the derivation below, these updates will be quite nontrivial in the case of SBM relative to the case where the data follows a simple mixture model given the partition.

3.1 Sampling $(\boldsymbol{u}, \boldsymbol{\gamma}')$

First, we sample $(\boldsymbol{u} \mid \boldsymbol{\gamma}', \Theta_{-\boldsymbol{u}\boldsymbol{\gamma}'})$, where $\Theta_{-\boldsymbol{u}\boldsymbol{\gamma}'}$ denotes all variables except $\boldsymbol{u}$ and $\boldsymbol{\gamma}'$. This density factorizes and coordinate posteriors are $p(u_{ti} \mid \boldsymbol{\gamma}', \Theta_{-\boldsymbol{u}\boldsymbol{\gamma}'}) \propto 1\{u_{ti} \le \gamma_{t,g_{ti}}\}$, that is

$$u_{ti} \mid \boldsymbol{\gamma}', \Theta_{-\boldsymbol{u}\boldsymbol{\gamma}'} \;\sim\; \mathsf{Unif}(0, \gamma_{t,g_{ti}}).$$

Next, we sample from $(\boldsymbol{\gamma}' \mid \Theta_{-\boldsymbol{u}\boldsymbol{\gamma}'})$. To do this, we first marginalize out $\boldsymbol{u}$ in (3.1) which gives back (2.11). The corresponding posterior is, thus, proportional to (2.11) viewed only as a function of $\boldsymbol{\gamma}'$. The posterior factorizes over t and g and we have (Amini et al., 2019, Lemma 1)

$$\gamma'_{tg} \mid \Theta_{-\boldsymbol{u}\boldsymbol{\gamma}'} \sim \mathsf{Beta}\left(n_g(\boldsymbol{g}_t) + 1, n_{>g}(\boldsymbol{g}_t) + \alpha_0\right), \tag{3.2}$$

where $n_g(\boldsymbol{g}_t) = |\{i : \; g_{ti} = g\}|$ and $n_{>g}(\boldsymbol{g}_t) = |\{i : \; g_{ti} > g\}|$.

3.2 Sampling $(\boldsymbol{v}, \boldsymbol{\pi}')$

First, we sample $(\boldsymbol{v} \mid \boldsymbol{\pi}', \Theta_{-\boldsymbol{v}\boldsymbol{\pi}'})$ which factorizes and coordinate posteriors are $p(v_{tg} \mid \boldsymbol{\pi}', \Theta_{-\boldsymbol{v}\boldsymbol{\pi}'}) \propto 1\{v_{tg} \leq \beta_{k_{tg}}\}$, that is

$$v_{tg} \mid \boldsymbol{\pi}', \Theta_{-\boldsymbol{v}\boldsymbol{\pi}'} \ \sim \ \mathsf{Unif}(0, \pi_{k_{tg}}).$$

Next, we sample from $(\boldsymbol{\pi}' \mid \Theta_{-\boldsymbol{v}\boldsymbol{\pi}'})$. As in the case of $\boldsymbol{\gamma}'$, we first marginalize $\boldsymbol{v}$ which leads to the usual block Gibbs sampler updates: The posterior factorizes over k, and

$$\pi'_k \mid \Theta_{-\boldsymbol{v}\boldsymbol{\pi}'} \ \sim \ \mathsf{Beta}\big(n_k(\boldsymbol{k}) + 1, n_{>k}(\boldsymbol{k}) + \gamma_0\big), \tag{3.3}$$

where $n_k(\boldsymbol{k}) = |\{(t, g) : \ k_{tg} = k\}|$ and similarly for $n_{>k}(\boldsymbol{k})$.

3.3 Sampling $\boldsymbol{g}$

This posterior factorizes over t (but not over i). From (3.1), we have

$$\mathbb{P}(g_{ti} = g \mid \boldsymbol{g}_{-ti}, \Theta_{-\boldsymbol{g}}) \ \propto \prod_{j \in [n_t] \setminus \{i\}} L\big(\eta(k_{tg}, k_{t,g_{tj}}); A_{tij}\big) 1\{u_{ti} \leq \gamma_{tg}\}. \tag{3.4}$$

Let $G_{ti} := \sup\{g : \ u_{ti} \leq \gamma_{tg}\}$. According to the above equation, g_{ti} given everything else will be distributed as

$$g_{ti} \mid \cdots \ \sim \ \big(\, \rho_{ti}(g) \,\big)_{g \, \in \, [G_{ti}]},$$

where, using $k_{t,g_{tj}} = z_{tj}$ and $L(p; a) = p^a (1-p)^{1-a}$,

$$\begin{aligned} \rho_{ti}(g) &:= \prod_{j \in [n_t] \setminus \{i\}} L\big(\eta(k_{tg}, z_{tj}); A_{tij}\big) = \prod_{j \in [n_t] \setminus \{i\}} \prod_{\ell} \Big[L\big(\eta(k_{tg}, \ell); A_{tij}\big) \Big]^{1\{z_{tj} = \ell\}} \\ &= \prod_{\ell} \eta(k_{tg}, \ell)^{\tau_{ti\ell}} [1 - \eta(k_{tg}, \ell)]^{m_{ti\ell} - \tau_{ti\ell}}, \end{aligned}$$

with

$$\tau_{ti\ell} := \sum_{j \in [n_t] \setminus \{i\}} A_{tij} 1\{z_{tj} = \ell\}, \quad m_{ti\ell} := \sum_{j \in [n_t] \setminus \{i\}} 1\{z_{tj} = \ell\}. \tag{3.5}$$

3.4 Sampling $\boldsymbol{k}$

This posterior also factorizes over t (but not over g). First, note that since we are conditioning on $\boldsymbol{g}$, we can simplify as

$$\begin{aligned} \prod_{1 \leq i < j \leq n_t} L\big(\eta(k_{t,g_{ti}}, k_{t,g_{tj}}); A_{tij}\big) &= \prod_{1 \leq i < j \leq n_t} \prod_{g,g'=1}^{\infty} \big[L\big(\eta(k_{tg}, k_{tg'}); A_{tij}\big) \big]^{1\{g_{ti} = g, \, g_{tj} = g'\}} \\ &= \prod_{g,g'=1}^{\infty} h_{tgg'}(k_{tg}, k_{tg'}) \end{aligned}$$

where $h_{tgg'}(k,\ell) := \eta(k,\ell)^{\xi_{tgg'}}[1-\eta(k,\ell)]^{O_{tgg'}-\xi_{tgg'}}$,

$$\xi_{tgg'} := \sum_{i,j} A_{tij} 1\{g_{ti}=g,\, g_{tj}=g'\}, \quad O_{tgg'} := \sum_{i,j} 1\{g_{ti}=g,\, g_{tj}=g'\}, \tag{3.6}$$

and the summations are over $1 \le i < j \le n_t$. Then, the posterior of $\boldsymbol{k} \mid \cdots$ factorizes over t, and for any fixed t,

$$p(\boldsymbol{k}_t \mid \boldsymbol{k}_{-t}, \Theta_{-\boldsymbol{k}}) \;\propto\; \prod_{g,g'=1}^{\infty} h_{tgg'}(k_{tg}, k_{tg'}) \prod_{g=1}^{\infty} 1\{v_{tg} \le \pi_{k_{tg}}\}. \tag{3.7}$$

Note that $\xi_{tgg'}$ is not symmetric in g and g', because of the condition $i<j$ in the summation. Letting $\widetilde{h}_{tgg'}(k,\ell) := h_{tgg'}(k,\ell)\, h_{tg'g}(\ell,k)$, it follows that for any fixed t and g:

$$\mathbb{P}(k_{tg}=k \mid \boldsymbol{k}_{-tg}, \Theta_{-\boldsymbol{k}}) \;\propto\; 1\{v_{tg} \le \pi_k\}\, h_{tgg}(k,k) \prod_{g':\, g'\neq g} \widetilde{h}_{tgg'}(k, k_{tg'}). \tag{3.8}$$

By the symmetry of $\eta(k,\ell)$ in its arguments, $h_{tgg'}(k,\ell)$ is also symmetric in (k,ℓ), and

$$\widetilde{h}_{tgg'}(k,\ell) = \eta(k,\ell)^{\widetilde{\xi}_{tgg'}}[1-\eta(k,\ell)]^{\widetilde{O}_{tgg'}-\widetilde{\xi}_{tgg'}} \tag{3.9}$$

where $\widetilde{\xi}_{tgg'} = \xi_{tgg'} + \xi_{tg'g}$ and $\widetilde{O}_{tgg'} = O_{tgg'} + O_{tg'g}$. That is,

$$\widetilde{\xi}_{tgg'} := \sum_{1\le i\neq j\le n_t} A_{tij} 1\{g_{ti}=g,\, g_{tj}=g'\}, \quad \widetilde{O}_{tgg'} := \sum_{1\le i\neq j\le n_t} 1\{g_{ti}=g,\, g_{tj}=g'\}.$$

Let us simplify the last factor in (3.8) further. Using (3.9), we have

$$\begin{aligned}\prod_{g':\, g'\neq g} \widetilde{h}_{tgg'}(k,k_{tg'}) &= \prod_{\ell} \prod_{g':\, g'\neq g} \big[\widetilde{h}_{tgg'}(k,\ell)\big]^{1\{k_{tg'}=\ell\}} \\ &= \prod_{\ell} \eta(k,\ell)^{\zeta_{tg\ell}} \big[1-\eta(k,\ell)\big]^{R_{tg\ell}-\zeta_{tg\ell}},\end{aligned}$$

where

$$\zeta_{tg\ell} := \sum_{g':\, g'\neq g} \widetilde{\xi}_{tgg'} 1\{k_{tg'}=\ell\}, \quad R_{tg\ell} := \sum_{g':\, g'\neq g} \widetilde{O}_{tgg'} 1\{k_{tg'}=\ell\}. \tag{3.10}$$

According to the above, k_{tg} given everything else will be distributed as

$$k_{tg} \mid \cdots \;\sim\; \big(\delta_{tg}(k)\big)_{k\,\in\,[K_{tg}]},$$

where $K_{tg} := \sup\{k :\, v_{tg} \le \pi_k\}$ and

$$\delta_{tg}(k) = \eta_{kk}^{\xi_{tgg}}(1-\eta_{kk})^{O_{tgg}-\xi_{tgg}} \prod_{\ell} \eta_{k\ell}^{\zeta_{tg\ell}}[1-\eta_{k\ell}]^{R_{tg\ell}-\zeta_{tg\ell}}.$$

3.5 Sampling η

Recalling that $z_{ti} = k_{t,g_{ti}}$, the relevant part of (3.1) is

$$\Big[\prod_{t=1}^{T}\prod_{1\le i<j\le n_t} L\big(\eta(z_{ti},z_{tj});A_{tij}\big)\Big]\prod_{1\le k\le \ell<\infty} b_{\alpha_\eta,\beta_\eta}(\eta_{k\ell}).$$

Using the fact that $\eta(k,\ell)=\eta_{k\ell}$ is symmetric in its two arguments (that is, we treat $\eta_{k\ell}$ and $\eta_{\ell k}$ as the same variable), we have, for $k\le\ell$,

$$p(\eta_{k\ell}\mid\cdots)\ \propto\ \eta_{k\ell}^{\lambda_{k\ell}}(1-\eta_{k\ell})^{N_{k\ell}-\lambda_{k\ell}}\, b_{\alpha_\eta,\beta_\eta}(\eta_{k\ell}),$$

where

$$\lambda_{k\ell}=\sum_{t=1}^{T}\sum_{i,j}A_{tij}1\{z_{ti}=k,z_{tj}=\ell\},\quad N_{k\ell}=\sum_{t=1}^{T}\sum_{i,j}1\{z_{ti}=k,z_{tj}=\ell\},\tag{3.11}$$

and the (i,j) summations are over $1\le i<j\le n_t$ for $k=\ell$ and over $1\le i\ne j\le n_t$ for $k\ne\ell$. We conclude that for $k\le\ell$,

$$\eta_{k\ell}\mid\cdots\ \sim\ \mathsf{Beta}\big(\,\lambda_{k\ell}+\alpha_\eta,\ N_{k\ell}-\lambda_{k\ell}+\beta_\eta\,\big).$$

Remark 1. Comparing with the updates in Amini et al. (2019), we observe that, under HSBM, the updates for $\boldsymbol{k}$ and $\boldsymbol{g}$ (which is equivalent to $\boldsymbol{t}$ in Amini et al. (2019)) and $\boldsymbol{\eta}$ (which is equivalent to parameters of $\boldsymbol{f}$ in Amini et al. (2019)) are much more complex. For the usual HDP, the data are assumed to follow a simple mixture model where observations are independent given the labels, and each, only depends on its own label. This causes the posterior for $\boldsymbol{k}$, $\boldsymbol{g}$ and the parameters of the mixture components to factorize over their coordinates. In contrast, the SBM likelihood makes each observation A_{tij} dependent on two labels z_{ti} and z_{tj}. This causes the posterior for $\boldsymbol{k}$, $\boldsymbol{g}$ and $\boldsymbol{\eta}$ to remain coupled under HSBM. Nevertheless, the Gibbs sampling scheme above allows us to effectively sample from these coupled multivariate posteriors.

3.6 Computational speedup

Let us discuss some ideas that lead to the implementation of a fast sampler. We consider three ideas: Spare matrix computations, parallel versus sequential label updates and truncation versus slice sampling.

Many of the key computations for the slice sampler can be sped up for sparse networks, using fast sparse matrix-vector operations. Consider for example the computation of $\boldsymbol{\tau}_t=(\tau_{i\ell})$. This can be done by defining the operator "row-compress$(A,\boldsymbol{z})$" that takes a $A\in\mathbb{R}^{n\times n}$ and a label vector $\boldsymbol{z}\in[K]^n$ and compresses each row of matrix A by summing over entries having the same label according to $\boldsymbol{z}$, producing an $n\times K$ matrix. Then, we will have

$$\boldsymbol{\tau}_t=\text{row-compress}(\boldsymbol{A}_t,\boldsymbol{z}_t).\tag{3.12}$$

Let M_t be the number of nonzero elements of $\boldsymbol{A}_t \in \mathbb{R}^{n_t \times n_t}$. The above operator can be implemented in $O(M_t)$ operation by iterating only over the nonzero entries of $\boldsymbol{A}_t$ once. When $\boldsymbol{A}_t$ is sparse, this allows for a significant computational saving relative to the naive approach which takes $O(n_t^2)$, since in the sparse case, usually $M_t = O(n_t)$.

The operation (3.12) is thus extremely fast if implemented *in parallel*, i.e., the entire matrix $\boldsymbol{\tau}_t$ is computed all at once. To have a valid Gibbs sampler, however, one needs to perform row compression one row at a time in a *sequential* manner, since once one element of $\boldsymbol{z}_t$ is updated, the row compression for subsequent rows will be affected. The sequential row compression can be implemented with the same complexity of $O(M_t)$, but cannot benefit from parallelism, hence will potentially be slower than the parallel version. We refer to the two versions of the sampler as HSBM-par and HSBM-seq, respectively, based on whether the label updates are done in parallel or sequential.

To summarize, HSBM-par is doing an approximation of a valid Gibbs sampler (not an exact Gibbs sampler). HSBM-seq computes the current row compression, samples the corresponding label and then uses this new label when computing the compression for the next row. HSBM-par, however, computes the row compressions all in parallel and samples the labels all in parallel, using the current snapshot the labels. HSBM-par does not take into account the effect of sequentially sampling of labels on subsequent row-compressions. In practice, we have found that HSBM-par performs as good as HSBM-seq and will be the default implementation used in simulations.

Similarly, computing $\widetilde{\boldsymbol{\xi}}_t = (\widetilde{\xi}_{tgg'})$ and $\boldsymbol{\lambda} = (\lambda_{k\ell})$ which are the key computations in updating $\boldsymbol{k}$ and $\boldsymbol{\eta}$, can be done extremely fast. To see this, consider the operator "block-compress$(A, \boldsymbol{z})$" that returns the block compression of A according to labels $\boldsymbol{z}$, by summing all the entries of A over blocks having the same row-column label pair. If $\boldsymbol{z} \in [K]^n$, the output will be a $K \times K$ matrix and it can be computed by traversing only the nonzero entries of A once. We have $\widetilde{\boldsymbol{\xi}}_t = \text{block-compress}(\boldsymbol{A}_t, \boldsymbol{g}_t)$ and $\boldsymbol{\lambda} = \sum_t \text{block-compress}(\boldsymbol{A}_t, \boldsymbol{z}_t)$, ignoring the minor modifications needed depending on whether diagonal entries need to be accounted for or not. Both of these calculations can be done in $O(\sum_t M_t)$ operations.

Truncation sampler Finally, it is possible to speed up the convergence by switching to a *truncation sampler*: Instead of introducing auxiliary variables $\boldsymbol{u}$ and $\boldsymbol{v}$ to truncate the infinite measures automatically, one can truncate them at a fixed sufficiently large index. That is, we sample the (approximate) joint density

$$p(\boldsymbol{g}, \boldsymbol{k}, \boldsymbol{\gamma}', \boldsymbol{\pi}') \approx \prod_{t=1}^{T} \left(\prod_{i=1}^{n_t} \gamma_{t,g_{ti}} \prod_{g=1}^{G} b_{1,\alpha_0}(\gamma'_{tg}) \prod_{g=1}^{G} \pi_{k_{tg}} \right) \prod_{k=1}^{K} b_{1,\gamma_0}(\pi'_k). \tag{3.13}$$

where K and G are pre-specified large integers. The resulting Gibbs sampler is almost identical to the slice sampler with minor modifications. In particular, in the $\boldsymbol{g}$-update, we need to replace $\rho_{ti}(g)$ with $\rho_{ti}(g)\gamma_{t,g}$ and in the $\boldsymbol{k}$-update, $\delta_{tg}(k)$ with $\delta_{tg}(k)\pi_k$. Otherwise, everything else remains the same. Empirically, we have found that the truncation sampler mixes faster than the slice sampler, hence will be our default choice in the simulations.

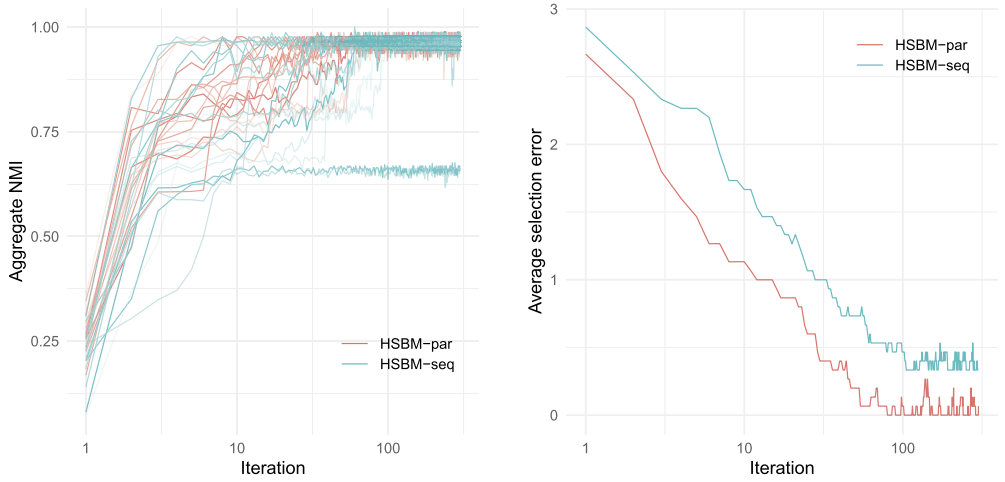


Figure 2: The convergence speed of the sampler(s) for the personality-friendship network with $T = 5$ layers and $n_t = 200$ nodes per layer. (a) Aggregate NMI relative to the true labels (b) Average error in estimating the true number of clusters.

Figure 2 illustrates the convergence speed of the sampler following these implementation choices. Figure 2(a) shows the aggregate NMI (Section 4.4) versus iteration for 15 realizations of the HSBM-par and HSBM-seq chains. The underlying network is a multilayer personality-friendship network (Section 4.1) with $T = 5$ layers and $n_t = 200$ nodes per layer. Figure 2(a) is the absolute deviation between the estimated total number of communities and the truth (i.e., 3 communities), averaged over the 15 realizations. Both plots indicate fast mixing, with convergence achieved under 50 iterations for most realizations. We refer to Section 4 for more details on the simulation setup. The code for the sampler(s) is available at Amini et al. (2021).

4 Simulation study

We consider networks generated from a multilayer SBM with a single connectivity matrix, but potentially varying labels across layers. We start with an example to illustrate how such assumptions naturally arise in some applications.

4.1 Multilayer personality-friendship network

Consider, as an illustration, the group of young women who run, every year, for the Miss World title. Each country holds a preliminary competition to choose their delegate to the main race. After meeting each other in the competition, some of the girls tend to bond and become friends at a social networking website. Suppose that we are interested

	Extrovert	Ambivert	Introvert
Extrovert	90%	75%	50%
Ambivert	75%	60%	25%
Introvert	50%	25%	10%

Table 1: Hypothetical probability of friendship between personality types.

in identifying groups of competitors by their personalty types: extrovert, introvert or ambivert (a balance between the former two). Let us also suppose that around 40% of the female population can be classed as extrovert, 35% introvert, and 25% is in the ambivert group. Assume that the distribution of personality types among the contestants in the Miss World competition is similar to that of the general population.

It is natural to assume that extroverts have a higher chance than introverts to form bonds with other contestants. For this experiment, we consider the probabilities of friendship between and within groups listed in Table 1. We consider each year of the competition as a layer and the competitors represent the nodes. Alternatively, since each country is represented by a single competitor, we can consider nodes as representing different countries. The membership group of each node (i.e., its personality type) varies from layer to layer since the same country is represented by different girls in different years. However, the connection probability, i.e., the $\boldsymbol{\eta}$ matrix in our notation, remains the same across layers, assuming that there is a fundamental pattern of connections among personality types rooted in human nature and invariant across years. This example thus naturally conforms to our setup of fixed $\boldsymbol{\eta}$ and potentially variable $\boldsymbol{z}_t$ over $t = 1, \ldots, T$.

4.2 Markovian labels and random connectivity

In order the systematically study the performance of HSBM and its competitors, we introduce additional degrees of freedom in generating simulated networks:

1. In addition to the $\boldsymbol{\eta}$ introduced in Table 1, we also consider a random symmetric connectivity matrix with entries on and above diagonal generated i.i.d. from $\mathsf{Unif}(0.1, 0.9)$. In the experiments where the random connectivity matrix is used, all layers share the same $\boldsymbol{\eta}$, however, different replications of the experiment use different randomly generated $\boldsymbol{\eta}$. Note that we do not restrict $\boldsymbol{\eta}$ to be assortative and the resulting random ensemble captures the full complexity possible in an SBM connectivity matrix.
2. We generate the labels $\boldsymbol{z}_t = (z_{ti}), t = 1, \ldots, T$ based on the following Markovian process: For simplicity, we assume that all layers have the same number of nodes n. The elements of $\boldsymbol{z}_1$ are generated i.i.d. from a Categorical distribution with probability vector $\boldsymbol{\pi}_0$, that is, $z_{1i} \sim \mathsf{Cat}(\boldsymbol{\pi}_0), i = 1, \ldots, n$. Subsequent labels are held fixed at previous layer value with probability $1 - \tau$ or randomly generated from $\mathsf{Cat}(\boldsymbol{\pi}_0)$, that is,

$$z_{ti} \sim (1 - \tau)\delta_{z_{t-1,i}} + \tau\, \mathsf{Cat}(\boldsymbol{\pi}_0)$$

for $t = 2, \ldots, T$ and $i = 1, \ldots, n$, where δ_x is the point-mass measure at x. We refer to τ as the transition probability.

3. We vary the number of layers, T, say from 2 to 12.

4.3 Competing methods

We compare the performance of HSBM with a wide variety of methods. The first natural competitor is to apply DP-SBM separately to each layer. Here, DP-SBM is a SBM with a DP prior on its labels. It can be thought of as HSBM with a single layer and with $\boldsymbol{w}_1$ set equal to $\boldsymbol{\pi}$. This model is essentially the same as the one considered in Mørup and Schmidt (2012).

We also compare with various spectral approaches: SC-sliced that applies spectral clustering separately to each slice; SC-avg that applies spectral clustering to the average adjacency matrix $\bar{A} = \frac{1}{T}\sum_{t=1}^{T} A_t$; SC-ba, the debiased spectral approach of Lei and Lin (2020); SC-omni, the omnibus spectral approach of Levin et al. (2017). In addition, we consider two versions of the PisCES algorithm (Liu et al., 2018) that solves an optimization problem that smooths out spectral projection matrices across time. PisCES implements the version described in Liu et al. (2018). PisCES-sh is a modification we made where we use the same initialization to start the k-means clustering algorithm across different layers. Without the shared initialization, there is no guarantee of a matching between clusters obtained by k-means applied to the spectral representation of different layers.

HSBM, DP-SBM, SC-sliced, SC-omni and PisCES naturally produce labels for all layers. SC-avg and SC-ba are designed to produce a single set of labels for all layers; their underlying assumption is that the labels remain the same across layers. For these two methods, we repeat the estimated label vector for all layers to obtain a multilayer label estimate.

4.4 Measuring performance

We measure the accuracy of the estimated cluster labels using the normalized mutual information (NMI), a well-known measure of similarity between two cluster assignments. NMI takes values in $[0, 1]$ where 1 corresponds to a perfect match. A random assignment against the truth is guaranteed to map to NMI ≈ 0. The NMI penalizes mismatch quite aggressively. In our setting, an NMI ≈ 0.5 corresponds to a roughly 90% match.

In the multilayer setting, we can compute at least two NMIs: (1) The *slicewise NMI* where one takes the average of NMIs computed separately for each layer, and (2) the *aggregate NMI* where we consider the labels for all the layers together and compute a single NMI between competing label assignments. Here, we focus mostly on the aggregate NMI, since achieving a high aggregate NMI is more challenging, requiring consistency both within and across layers.

For the HSBM and DP-SBM the estimated labels used in the NMI calculations will be the MAP estimates, calculated as detailed below.

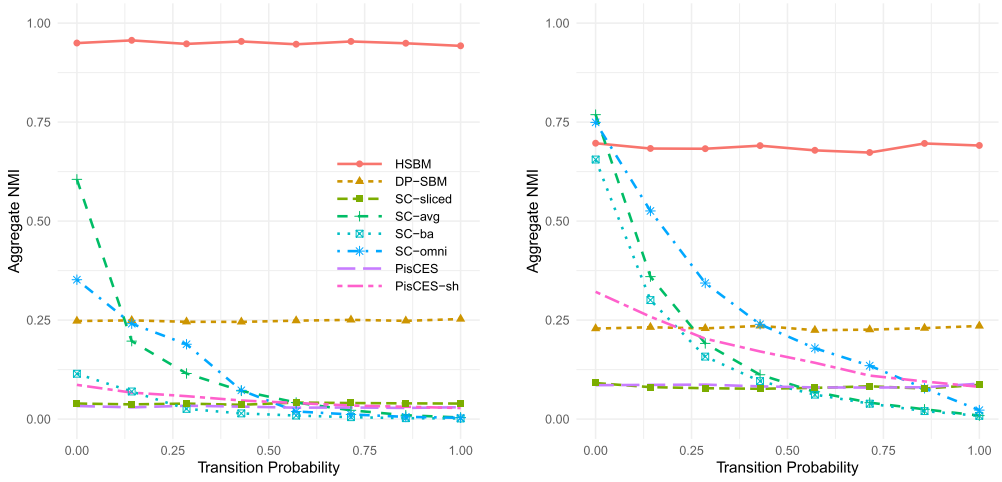


Figure 3: The performance of various methods for multilayer SBM networks with varying transition probability. (a) Personality network connectivity. (b) Random connectivity.

MAP estimate For HSBM, we compute the maximum a posteriori (MAP) label assignment by finding, for each node, the label which is most likely according to the posterior: $\text{argmax}_k \, \mathbb{P}(z_{ti} = k \mid \cdots)$. Associated with the MAP estimate, there is a *confidence* which is the value of the posterior probability. To compute the MAP estimate we use the posterior estimate given by $\frac{1}{N-N_0} \sum_{j=N_0+1}^{N} 1(\hat{z}_{ti}^{(j)} = k)$ where $\hat{\boldsymbol{z}}^{(j)} = (\hat{z}_{it}^{(j)})$ is the label assignment at MCMC iteration j, N is the total number of iterations and N_0 the length of the burn-in. Here, we ignore the potential mismatch between $\hat{\boldsymbol{z}}^{(j)}$ and $\hat{\boldsymbol{z}}^{(j')}$ due to the potential label-switching. In practice, all labels after MCMC convergence are coming from a single mode of the posterior as can be verified by computing the NMI between consecutive samples $\boldsymbol{z}^{(j)}$ and $\boldsymbol{z}^{(j+1)}$.

4.5 Results

We ran the HSBM and DP-SBM samplers for $N = 100$ iterations with a burn-in of $N_0 = N/2$. We consider two main settings, one where we fix the number of layers at $T = 5$ and change the label transition probability τ, and one where we fix the transition probability at $\tau = 0.25$ and vary the number of layers T. In both cases, there are $n_t = 200$ nodes in each layer. The results are shown in Figures 3 and 4 respectively. In each case, the expected aggregate NMI is computed by averaging over 500 replications. For each of the two settings, we have considered both a fixed $\boldsymbol{\eta}$, namely the personality-friendship connectivity of Table 1, and a random $\boldsymbol{\eta}$ generated as discussed in Section 4.2. Table 2 provides numerical values for the mean and the standard deviation of the aggregate and slicewise NMIs in a typical setting.

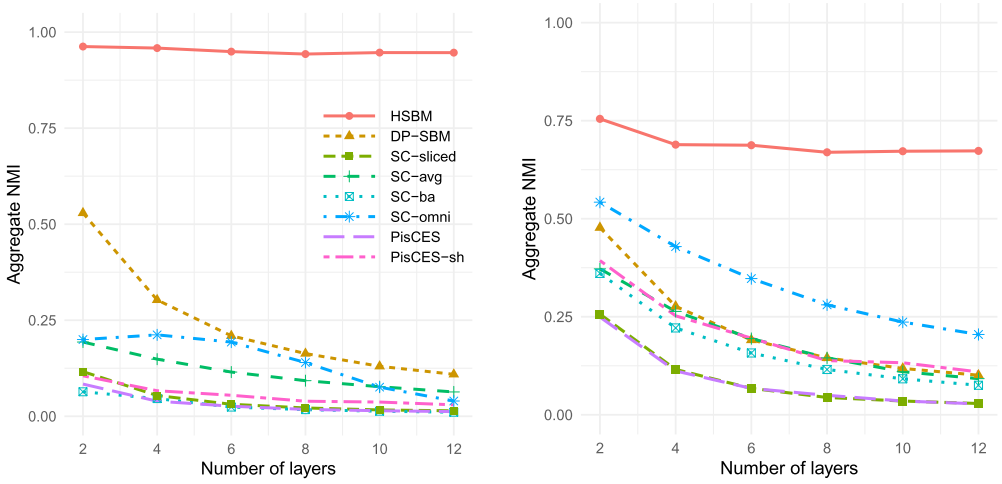


Figure 4: The performance of various methods for multilayer SBM networks with varying number of layers. (a) Personality network connectivity. (b) Random connectivity.

Method	Agg. NMI		Slice. NMI		Runtime (s)	
	mean	s.d.	mean	s.d.	mean	s.d.
HSBM	0.68	0.26	0.87	0.21	0.39	0.13
DP-SBM	0.22	0.07	0.83	0.17	1.15	0.34
SC-omni	0.18	0.09	0.33	0.11	0.28	0.14
PisCES-sh	0.14	0.15	0.56	0.28	0.18	0.08
PisCES	0.08	0.07	0.59	0.29	0.19	0.08
SC-sliced	0.08	0.08	0.56	0.30	0.10	0.05
SC-avg	0.07	0.03	0.11	0.04	0.05	0.02
SC-ba	0.06	0.02	0.12	0.03	0.15	0.06

Table 2: Mean and standard deviation for aggregate and slicewise NMIs as well as runtimes for a typical experiment. The setting is that of random $\boldsymbol{\eta}$ with Markov labels having transition probability $\tau \approx 0.57$ and $T = 5$ layers, corresponding to roughly the midpoint of the horizontal axis in Figure 3.

The results clearly show the superior performance of HSBM relative to the competing methods. In particular, Figure 3 shows that the performance of HSBM remains almost constant as one varies the transition probability (τ). Most spectral methods in contrast deteriorate as τ is increased. Note that at $\tau = 1$, the labels across layers are completely independent. Nevertheless the aggregate NMI for HSBM remains high even in this case. What allows HSBM to match the clusters correctly across layers is the shared connectivity matrix $\boldsymbol{\eta}$. The personality-friendship connectivity (Figure 3(a)) is especially

challenging for spectral methods, mainly due to the fact the connectivity matrix is nearly rank-deficient and non-assortative. In contrast, HSBM performs almost perfectly in this case, due to the very different connectivity patterns across clusters.

A similar qualitative behavior is observed as one varies T in Figure 4. The performance of most methods deteriorates as T is increased while that of HSBM remains roughly the same. Interestingly, in both experiments, SC-omni is overall the most performant among the spectral methods.

Remark. Note that aggregate NMI is used as a measure of performance in most of our figures. For a method to have a high aggregate NMI, it has to also properly match the communities across layers. This is the main reason why most of the SC methods fail when looking at the aggregate NMI. For example, the SC-sliced method, which is applied to each layer separately, does not have any such capability.

SC-ba method is designed for the same nodes having exactly the same communities across all layers (with only the connectivity matrix changing across layers). Thus, it fails to have high aggregate NMI in multiplex networks with varying community structures over time.

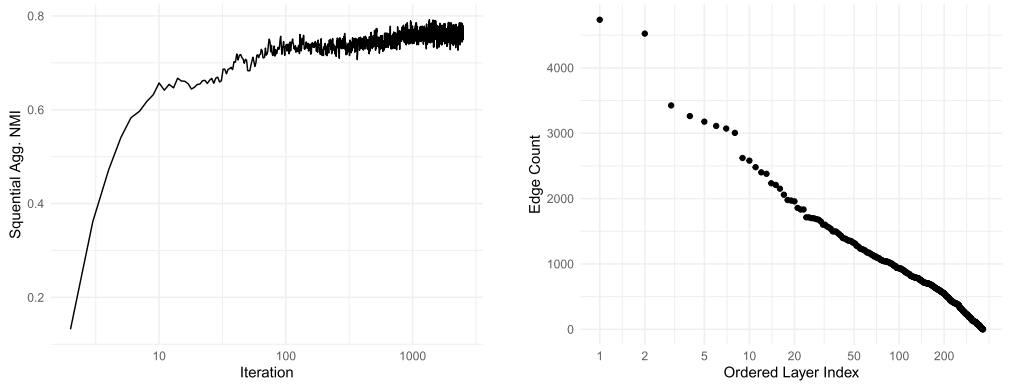
SC-omni and PiCES allow for variable community across layers and the SC-omni is doing quite well in terms of the aggregate NMI if the transition probability is not quite high. See for example Figure 3(b).

If we only want to measure how the method performs in each layer separately, slicewise NMI is a more appropriate measure. This measure for example is reported in Table 2 and one can verify that it is quite high for SC-sliced. On the other hand, slicewise NMI ignores the multiplex nature of the network, and that is why we focused on aggregate NMI which is a much more natural measure if one believes shared communities exist across layers.

5 Real data analysis: FAO Trade Network

In this section, we illustrate the performance of our model and algorithm on one real data example. We consider the multilayer FAO trade network provided by De Domenico et al. (2015). The data contains trade connections between 214 countries, treated as nodes, across 314 product categories considered as layers. We sorted the layers according to their total edge count, and chose the 20 most dense layers. Figure 5(b) shows the distribution of the edge counts, suggesting there is a natural cut-off at about 20 layers. We then selected the nodes that had a degree greater than 20 in the *sum network*, obtained by summing all the adjacency matrices. After filtering, we were left with a multilayer network on $n = 172$ nodes and 20 layers.

We then ran the HSBM sampler for 2500 iterations with a burn-in of 1250 and obtained the MAP labels. Figure 5(a) shows the sequential NMI plot for the sampler, obtained by computing the aggregate NMI between consecutive labels across the iterations of the chain. The plot suggests that by iteration 1000 the chain is already well-mixed.



(a) (b)

Figure 5: (a) Convergence on FAO network: Sequential Aggregate NMI versus iteration. (b) Edge count distribution for layers of FAO network.

The estimated labels uncover a rich community structure. Figure 6 shows the community assignment across some of the layers. The nodes are laid out using a force-directed algorithm that puts highly connected nodes closer together. That the communities inferred by HSBM align with the physical proximity in these layouts is an indication of the quality of the inferred communities.

There are a total of 15 estimated communities. Figure 7 shows the (MAP) estimated connectivity matrix $\boldsymbol{\eta}$ and Figure 8 shows the distribution of the 15 communities across the 20 layers. Community 9 which is the most frequent across a majority of the layers corresponds to countries that sparsely trade, as evidenced by the corresponding row (and column) in $\boldsymbol{\eta}$.

The connectivity matrix is not assortative–as evidenced, for example, by the near loop in Figure 7 among communities 10–14. This figure also suggests that communities 8 and 10 are rather interesting. Figure 9 shows the full community assignment for all the countries that are assigned communities 8 or 10 in at least one layer. The figure shows the complex nature of the inferred communities, with countries allowed to change communities across layers. Nevertheless, the countries that belong to group 10 across the majority of layers tend to form the tightly knit group {Germany, USA, Netherlands, UK, Italy, France, China, China (mainland)}. This is a reasonable group since these countries are known to have extensive trade relationships with each other. Interestingly, some of these countries behave like group 8 countries in some layers. Spain also has an interesting position, with equal assignments to group 10 and 8, respectively. Looking at the layers, group 8 seem to have something to do with the pattern of trade in crude materials and group 5 to the pattern of trade in green coffee. Perhaps the best way to interpret the inferred groups (or communities) is to note that they correspond to patterns of trade, hence a country can exhibit multiple of these patterns across different product.

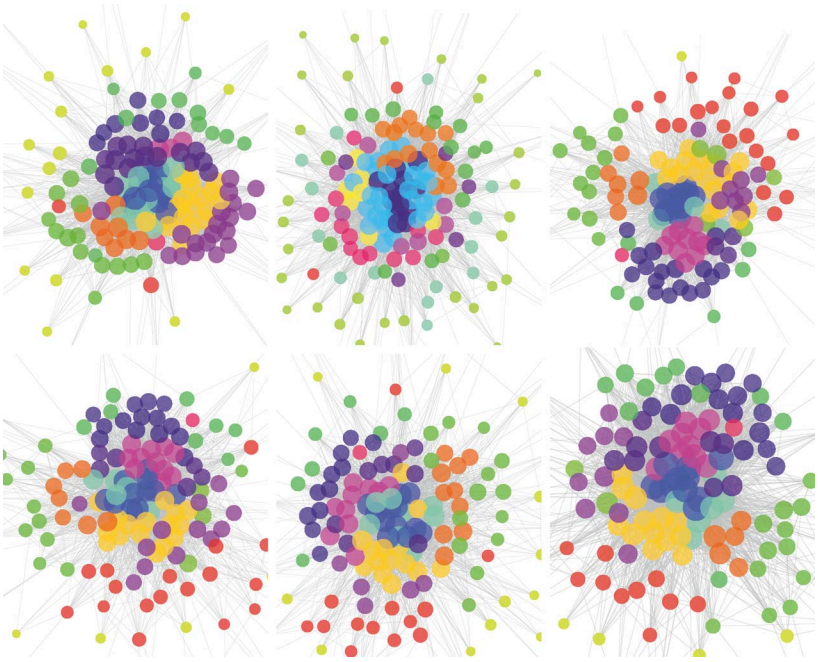


Figure 6: FAO trade network: Examples of the community assignments. Layers shown are (from top left): Food perp. nes., Crude materials, Pastry, Suger confectionery, Fruit perp. nes., Chocolate products nes. The layouts are generated by the Fruchterman–Reingold (FR) algorithm, applied separately to each layer. FR positions the nodes according to forces exerted along the edges, resulting in spatial proximity being correlated with network connectivity. The size of a node reflects its degree. More precisely, $s_i \propto \log(d_i + 3)$ where s_i and d_i are the size and degree of node i, respectively.

Since under HSBM, the same country can be assigned a different label for each layer, to visualize the community assignments better, we consider the following reduction: For each label, we collect all the nodes that have that label across at least 8 out of 20 layers. Figure 10 shows the word cloud of major groups obtained by this procedure. The size and color of a country indicate the frequency of the label across layers. For example, in Group 6, Canada has been assigned label 6 for 13 out of 20 layers, more than any other country in that group. Switzerland, Australia and Belgium were all assigned label 6 in 11 layers. In addition to the groups shown in the figure, we have three singleton groups: Chile, UAE and "unspecified".

The groups in Figure 10 make intuitive sense for the most part, with some of the groups roughly corresponding to geographical regions. The analysis, however, reveals many unintuitive connections, with seemingly unrelated countries grouped together. Examples include: Macao and Rwanda, Iraq and Guinea, Guam and Grenada, Fiji and Uganda, Bahrain and Mauritius, and so on.

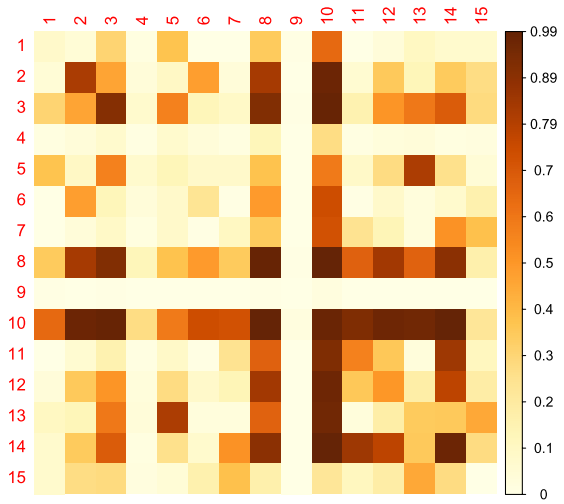


Figure 7: FAO network: The estimated connectivity matrix.

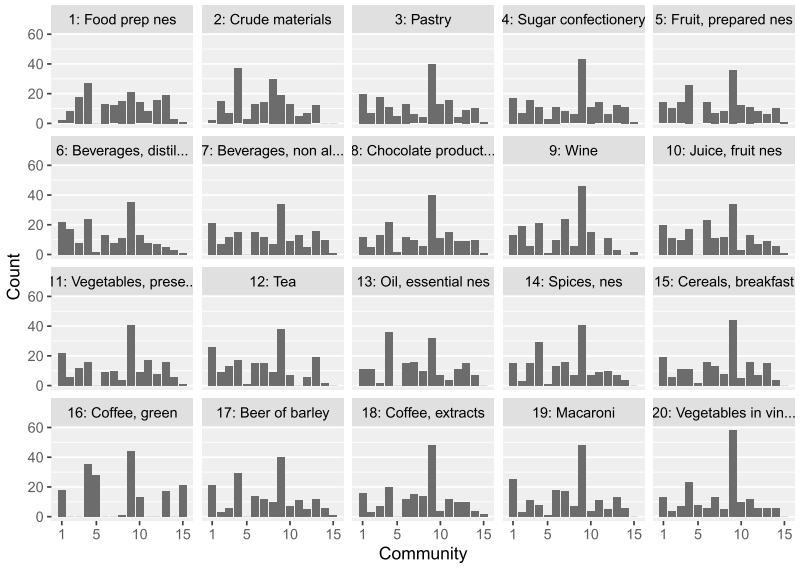


Figure 8: FAO network: The distribution of the 15 estimated communities across layers.

To verify these connections and the overall quality of clustering, we consider the following quantitative evaluation metric: For each pair of nodes, we compute the normalized Hamming distance for their corresponding rows in the adjacency matrices, and

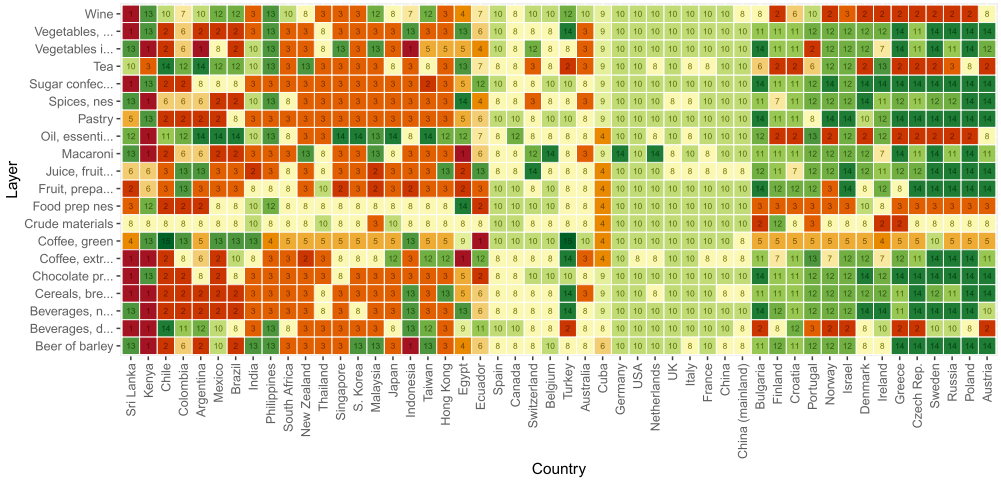


Figure 9: FAO network: Community assignments across layers for the countries that are assigned to one of the communities 8 or 10 in at least one layer.

then take the average over layers. That is, we consider the following distance

$$d(i,j) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{n_t}H(A_{ti*}, A_{tj*}),$$

with $H(\cdot,\cdot)$ denoting the Hamming distance. We refer to $d(i,j)$ as the Average Normalized Hamming (ANH) distance and note that it measures how dissimilar the connection patterns of two nodes are across layers. We expect nodes that are grouped together to have a lower value of $d(i,j)$ relative to a randomly selected pair.

Figure 11(a) shows plots of the distributions of $d(i,j)$ when the pairs are selected randomly within the groups versus the case were they are selected completely at random. The figure shows clearly that the ANH is much lower on average for within-group pairs relative to random pairs. This confirms that HSBM has uncovered groups based on trading patterns. The median for "Within" and "Random" distributions are 0.027 and 0.21, respectively. The unintuitive pairs mentioned earlier have ANH much lower than the average for random pairs. For example, the ANH for Macao–Rwanda, Iraq–Guinea and Guam–Grenada, Fiji–Uganda and Bahrain–Mauritius pairs are 0.074, 0.019, 0.0088, 0.14 and 0.14 respectively. This shows that these seemingly unrelated countries have similar trading pattern across multiple food product categories.

In Figure 11(a) the ANH is computed in-sample, that is, using the same 20 most dense layers used for fitting the HSBM. Figure 11(b) shows the corresponding plot when ANH is computed out-of-sample, using the remaining 344 layers. These layers are generally sparser, as reflected in the absolute value of ANH, but the plot shows a similar separation among the two distributions.

Figure 10: FAO trade network: Word cloud of groups based on frequency of assignment across layers.

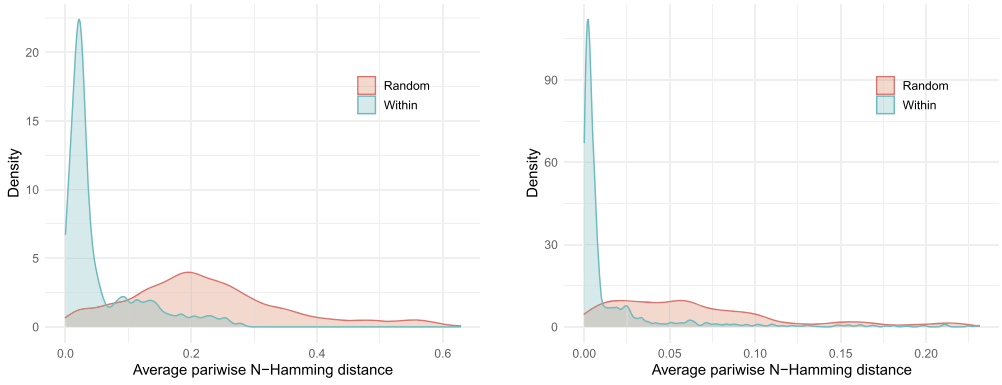


Figure 11: FAO trade network: Average Normalized Hamming (ANH) distance between pairs of nodes. The two distributions correspond to random selection of pairs, versus (random) selection within estimated groups. (a) Calculated based on the in-sample layers. (b) Calculated based on out-of-sample layers.

6 Discussion

In this work, we proposed a novel Bayesian model for community detection in multiplex networks by adopting the well-known HDP as a prior distribution for community assignments. Under the random partition prior, a block model is assumed. This model facilitates flexible modeling of community structure as well as link probabilities with its ability of incorporating potential dependency and borrowing strength among networks from different layers. For posterior inference of HSBM, we develop an efficient slicer sampler. The principles behind the slice sampler can be applied to developing sampling algorithms for many other models. Future work will be focused on developing models for community detection in networks with covariates, and for inference of network-valued objects.

References

Amini, A., Paez, M., and Lin, L. (2021). "The `hsbm` R package: Hierarchical Stochastic Block Model." URL https://github.com/aaamini/hsbm 3, 13

Amini, A. A., Paez, M., Lin, L., and Razaee, Z. S. (2019). "Exact slice sampler for Hierarchical Dirichlet Processes." *arXiv e-prints*, arXiv:1903.08829. 3, 6, 7, 8, 11

Battiston, F., Nicosia, V., and Latora, V. (2014). "Structural measures for multiplex networks." *Physical Review E*, 89: 032804. URL https://link.aps.org/doi/10.1103/PhysRevE.89.032804 2

Berlingerio, M., Coscia, M., and Giannotti, F. (2011). "Finding and Characterizing Communities in Multidimensional Networks." In *2011 International Conference on Advances in Social Networks Analysis and Mining*, 490–494. 2

Berlingerio, M., Coscia, M., Giannotti, F., Monreale, A., and Pedreschi, D. (2013). "Evolving Networks: Eras and Turning Points." *Intelligent Data Analysis*, 17(1): 27–48. 2

Bhattacharyya, S. and Chatterjee, S. (2018). "Spectral clustering for multiple sparse networks: I." *arXiv preprint arXiv:*1805.10594. 2

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). "Fast unfolding of communities in large networks." *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10): P10008. URL http://stacks.iop.org/1742-5468/2008/i=10/a=P10008 2

Boccaletti, S., Bianconi, G., Criado, R., [del Genio], C., Gómez-Gardeñes, J., Romance, M., Sendiña-Nadal, I., Wang, Z., and Zanin, M. (2014). "The structure and dynamics of multilayer networks." *Physics Reports*, 544(1): 1–122. The structure and dynamics of multilayer networks. MR3270140. doi: https://doi.org/10.1016/j.physrep.2014.07.001. 1

Bródka, P., Skibicki, K., Kazienko, P., and Musiał, K. (2011). "A degree centrality in multi-layered social network." In *2011 International Conference on Computational Aspects of Social Networks (CASoN)*, 237–242. 2

Carley, K. M., Martin, M. K., and Hirshman, B. R. (2009). "The Etiology of Social Change." *Topics in Cognitive Science*, 1(4): 621–650. MR3382220. doi: https://doi.org/10.1126/science.aac6076. 2

Cozzo, E., Kivelä, M., De Domenico, M., Solé, A., Arenas, A., Gómez, S., Porter, M. A., and Moreno, Y. (2013). "Clustering coefficients in multiplex networks." *arXiv preprint arXiv:*1307.6780. URL https://cds.cern.ch/record/1564815 2

De Bacco, C., Power, E. A., Larremore, D. B., and Moore, C. (2017). "Community detection, link prediction, and layer interdependence in multilayer networks." *Physical Review E*, 95(4): 042317. 2

De Domenico, M., Nicosia, V., Arenas, A., and Latora, V. (2015). "Structural reducibility of multilayer networks." *Nature communications*, 6(1): 1–9. 18

De Domenico, M., Sasai, S., and Arenas, A. (2016). "Mapping Multiplex Hubs in Human Functional Brain Networks." *Frontiers in Neuroscience*, 10: 326. URL http://journal.frontiersin.org/article/10.3389/fnins.2016.00326 2

Didier, G., Brun, C., and Baudot, A. (2015). "Identifying communities from multiplex biological networks." *PeerJ*, 3: e1525. 2

Gollini, I. and Murphy, T. B. (2016). "Joint Modeling of Multiple Network Views." *Journal of Computational and Graphical Statistics*, 25(1): 246–265. MR3474046. doi: https://doi.org/10.1080/10618600.2014.978006. 2

Greene, D. and Cunningham, P. (2013). "Producing a Unified Graph Representation from Multiple Social Network Views." In *Proceedings of the 5th Annual ACM Web Science Conference*, WebSci '13, 118–121. New York, NY, USA: ACM. URL http://doi.acm.org/10.1145/2464464.2464471 2

Hmimida, M. and Kanawati, R. (2015). "Community detection in multiplex networks: A seed-centric approach." *Networks and Heterogeneous Media*, 10(1): 71–85. MR3359512. doi: https://doi.org/10.3934/nhm.2015.10.71. 2

Ishwaran, H. and James, L. (2001). "Gibbs Sampling Methods for Stick-Breaking Priors." *Journal of the American Statistical Association*, 96(453): 161–173. MR1952729. doi: https://doi.org/10.1198/016214501750332758. 6

Kalli, M., Griffin, J. E., and Walker, S. G. (2011). "Slice sampling mixture models." *Statistics and Computing*, 21(1): 93–105. MR2746606. doi: https://doi.org/10.1007/s11222-009-9150-y. 7

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., and Porter, M. A. (2014). "Multilayer networks." *Journal of Complex Networks*, 2(3): 203–271. URL https://doi.org/10.1093/comnet/cnu016 1

Kuncheva, Z. and Montana, G. (2015). "Community Detection in Multiplex Networks Using Locally Adaptive Random Walks." In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, ASONAM '15, 1308–1315. New York, NY, USA: ACM. URL http://doi.acm.org/10.1145/2808797.2808852 2

Lei, J. and Lin, K. Z. (2020). "Bias-adjusted spectral clustering in multi-layer stochastic block models." *arXiv preprint arXiv:*2003.08222. 15

Levin, K., Athreya, A., Tang, M., Lyzinski, V., Park, Y., and Priebe, C. E. (2017). "A central limit theorem for an omnibus embedding of multiple random graphs and implications for multiscale network inference." *arXiv preprint arXiv:*1705.09355. 15

Liu, F., Choi, D., Xie, L., and Roeder, K. (2018). "Global spectral clustering in dynamic networks." *Proceedings of the National Academy of Sciences*, 115(5): 927–932. MR3763702. doi: https://doi.org/10.1073/pnas.1718449115. 2, 15

Majdandzic, A., Podobnik, B., Buldyrev, S. V., Kenett, D. Y., Havlin, S., and Eugene Stanley, H. (2014). "Spontaneous recovery in dynamical networks." *Nature Physics*, 10: 34–38. 2

Matias, C. and Miele, V. (2017). "Statistical clustering of temporal networks through a dynamic stochastic block model." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(4): 1119–1141. MR3689311. doi: https://doi.org/10.1111/rssb.12200. 2

Mørup, M. and Schmidt, M. N. (2012). "Bayesian Community Detection." *Neural Computation*, 24(9): 2434–2456. MR2986776. doi: https://doi.org/10.1162/NECO_a_00314. 15

Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., and Onnela, J.-P. (2010). "Community structure in time-dependent, multiscale, and multiplex networks." *Science*, 328(5980): 876–878. MR2662590. doi: https://doi.org/10.1126/science.1184819. 2

Pensky, M. and Zhang, T. (2019). "Spectral clustering in the dynamic stochastic block model." *Electronic Journal of Statistics*, 13(1): 678–709. MR3914178. doi: https://doi.org/10.1214/19-ejs1533. 2

Picard, J. and Pitman, J. (2006). *Combinatorial Stochastic Processes: Ecole d'Eté de Probabilités de Saint-Flour XXXII - 2002*. Lecture Notes in Mathematics. Springer Berlin Heidelberg. URL https://books.google.rw/books?id=6qFTR4PZE4AC MR2245368. 6

Poledna, S., Molina-Borboa, J. L., Martínez-Jaramillo, S., Van Der Leij, M., and Thurner, S. (2015). "The multi-layer network nature of systemic risk and its implications for the costs of financial crises." *Journal of Financial Stability*, 20: 70–81. 2

Raftery, A. E., Handcock, M. S., and Hoff, P. D. (2002). "Latent space approaches to social network analysis." *Journal of the American Statistical Association*, 15: 460. MR1951262. doi: https://doi.org/10.1198/016214502388618906. 2

Salter-Townshend, M. and McCormick, T. H. (2017). "Latent space models for multiview network data." *Annals of Applied Statistics*, 11(3): 1217–1244. MR3709558. doi: https://doi.org/10.1214/16-AOAS955. 2

Sarkar, P. and Moore, A. (2005). "Dynamic Social Network Analysis using Latent Space Models." In *Advances in Neural Information Processing Systems*. 2

Sethuraman, J. (1994). "A constructive definition of Dirichlet priors." *Statistica Sinica*, 4: 639–650. MR1309433. 6

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). "Hierarchical Dirichlet Processes." *Journal of the American Statistical Association*, 101(476): 1566–1581. MR2279480. doi: https://doi.org/10.1198/016214506000000302. 2, 5, 6

Wang, J., Peng, X., Peng, W., and Wu, F.-X. (2014). "Dynamic protein interaction network construction and applications." *PROTEOMICS*, 14(4-5): 338–352. URL http://dx.doi.org/10.1002/pmic.201300257 2

Wilson, J. D., Palowitch, J., Bhamidi, S., and Nobel, A. B. (2017). "Community Extraction in Multilayer Networks with Heterogeneous Community Structure." *Journal of Machine Learning Research*, 18(1): 5458–5506. URL http://dl.acm.org/citation.cfm?id=3122009.3208030 MR3763783. 2