Thinned random measures for sparse graphs with overlapping communities

Federica Zoe Ricci
Department of Statistics
University of California, Irvine, CA, USA
fzricci@uci.edu

Michele Guindani
Department of Biostatistics
University of California, Los Angeles, CA, USA
mguindani@g.ucla.edu

Erik B. Sudderth
Departments of Computer Science and Statistics
University of California, Irvine, CA, USA
sudderth@uci.edu

Abstract

Network models for exchangeable arrays, including most stochastic block models, generate dense graphs with a limited ability to capture many characteristics of real-world social and biological networks. A class of models based on completely random measures like the generalized gamma process (GGP) have recently addressed some of these limitations. We propose a framework for thinning edges from realizations of GGP random graphs that models observed links via nodes' overall propensity to interact, as well as the similarity of node memberships within a large set of latent communities. Our formulation allows us to learn the number of communities from data, and enables efficient Monte Carlo methods that scale linearly with the number of observed edges, and thus (unlike dense block models) sub-quadratically with the number of entities or nodes. We compare to alternative models for both dense and sparse networks, and demonstrate effective recovery of latent community structure for real-world networks with thousands of nodes.

1 Introduction

Given observations of (often binary) relationships Y_{ij} between pairs of nodes or entities (i, j), many relational models [1] seek to uncover an underlying set of communities. Classic stochastic blockmodels [2] generalize mixture models for clustering non-relational data by assigning each entity to one of K communities (clusters). The *infinite relational model* (IRM) [3] instead uses a Dirichlet process prior [4] to partition entities into single communities. While the IRM allows the number of communities to be inferred from data, later work has shown that real-world social networks are better captured by models which allow nodes to participate in multiple communities [5], including applications of the *hierarchical Dirichlet process* (HDP) [6] to relational data [7]. Node relationships may also be modeled by shared features [8, 9] learned via the *Indian Buffet Process* [10], by a combination of node and interaction factors [11], or by proximity in a latent space [12, 13].

There is an extensive literature on descriptive statistics of biological and social networks [14, 15] including degree distributions, path distances and "small world" phenomena [16], community structures and modularity, and notions of centrality and causality. In particular, sparsity is a ubiquitous phenomenon in real-world networks [14, 15]: as network size grows, the number of edges grows more slowly than the quadratic number of node pairs. However, the IRM and HDP relational models (and a large literature of related models [1]) generate *dense* graphs where the number of edges scales quadratically with the number of nodes. In fact, a classic representation theorem [17, 18] shows

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

that any generative model that regards graphs as exchangeable adjacency matrices, meaning arrays whose distribution is invariant to permutations of node indices, generates dense graphs. Most existing probabilistic network models, which generate an adjacency matrix by sampling edges from independent Bernoulli distributions given latent node-specific parameters, have this limitation. Another important property of many real-world networks is assortative mixing [19, 20, 21], that is the presence of more popular or sociable nodes to which other nodes are more likely to connect. Models with degree-correction mechanisms [22, 23, 24] account for this phenomenon.

By representing graphs as a latent process (a completely random measure), Caron and Fox [25] showed that it is possible to formulate generative models that capture the sparsity of real-world networks as well as assortative mixing. Related models, including certain infinite limits of graphs called *graphons*, have been studied by several authors [26, 27]. However, these models mostly produce homogeneous graphs with sparsity and heavy-tailed degree distributions, but lacking the community structure of real networks. Two notable exceptions are work by Herlau et al. [28] and Todeschini et al. [29] that augment random-measure models [25] with latent community structure. Alternative approaches to sparse networks regard edges, rather than nodes, as the core of the generative process [30, 31, 32]. Another approach to simultaneously capture local density and global sparsity models graphs as a collection of cliques [33].

In this paper, we propose a novel random graph model that efficiently thins edges from sparse homogeneous graphs to reveal community structure while maintaining sparsity. Unlike [28], we allow entities to have membership in multiple communities. Unlike [29], we allow completely flexible specification of the hierarchical Bayesian prior on latent community memberships; importantly, this enables the network-specific learning of the appropriate number of latent communities. We further develop an efficient Monte Carlo inference algorithm that, unlike nearly all dense block models, scales linearly with the number of observed edges and thus sub-quadratically with the number of entities. Experiments show recovery of communities for networks with thousands of nodes.

2 Background: Stochastic Blockmodels for Dense and Sparse Networks

An undirected binary network with N nodes and E edges may be represented by a symmetric $N \times N$ adjacency matrix Y. If there is an edge (link) between nodes $i \neq j$ then $Y_{ij} = 1$, otherwise $Y_{ij} = 0$.

2.1 Mixed Membership Stochastic Blockmodels

Stochastic blockmodels (SBMs) [2] assume that each node belongs to one of K latent communities, and that the probability of an edge depends on how strongly their communities are connected. The community $c_i \in \{1, \dots, K\}$ of node i follows a categorical distribution $c_i \overset{\text{ind}}{\sim} \text{Cat}(\beta)$, where $\beta = (\beta_1, \dots, \beta_K)$ so that β_k is the frequency of community k. Edges are sampled independently as $Y_{ij} \overset{\text{ind}}{\sim} \text{Bernoulli}(\eta_{c_i c_j})$, where $\eta_{k\ell} = \eta_{\ell k}$ is the probability of an edge between nodes in communities k and ℓ. These interaction probabilities are often assigned conjugate beta priors, $\eta_{k\ell} \overset{\text{ind}}{\sim} \text{Beta}(\tau_a, \tau_b)$.

Mixed membership stochastic blockmodels (MMSBs) [5] extend SBMs to allow nodes to be members of multiple communities. Let $\pi_i = (\pi_{i1}, \dots, \pi_{iK})$ denote a K-dimensional probability vector representing the strength of affiliation of node i to each of K communities. For every pair of nodes (i, j), the communities governing their interaction are sampled as $c_{ij} \overset{\text{ind}}{\sim} \text{Cat}(\pi_i)$, $c_{ji} \overset{\text{ind}}{\sim} \text{Cat}(\pi_j)$. Then, like standard SBMs, edges are sampled independently as $Y_{ij} \overset{\text{ind}}{\sim} \text{Bernoulli}(\eta_{c_{ij} c_{ji}})$. Community memberships are typically assigned a hierarchical prior, such as

$$\pi_i \mid \beta \overset{\text{ind}}{\sim} \text{Dirichlet}(\zeta\beta_1, \dots, \zeta\beta_K), \qquad \beta \sim \text{Dirichlet}\left(\frac{\gamma}{K}, \dots, \frac{\gamma}{K}\right). \tag{1}$$

The concentration parameter ζ controls the polarization of π_i and its variation around β, with smaller values of ζ inducing more polarized community memberships that place significant probability on only a few communities, and larger values of ζ inducing community memberships that differ very little from β. When fitting this model to data, we can treat K as an upper bound to the number of communities (that is, larger than the maximum number of communities that we expect to be necessary to model the network), fix $\gamma \ll K$ and let ζ be small. In this way, the hierarchical Dirichlet formulation allows network-specific learning of the number of communities by favoring sparse

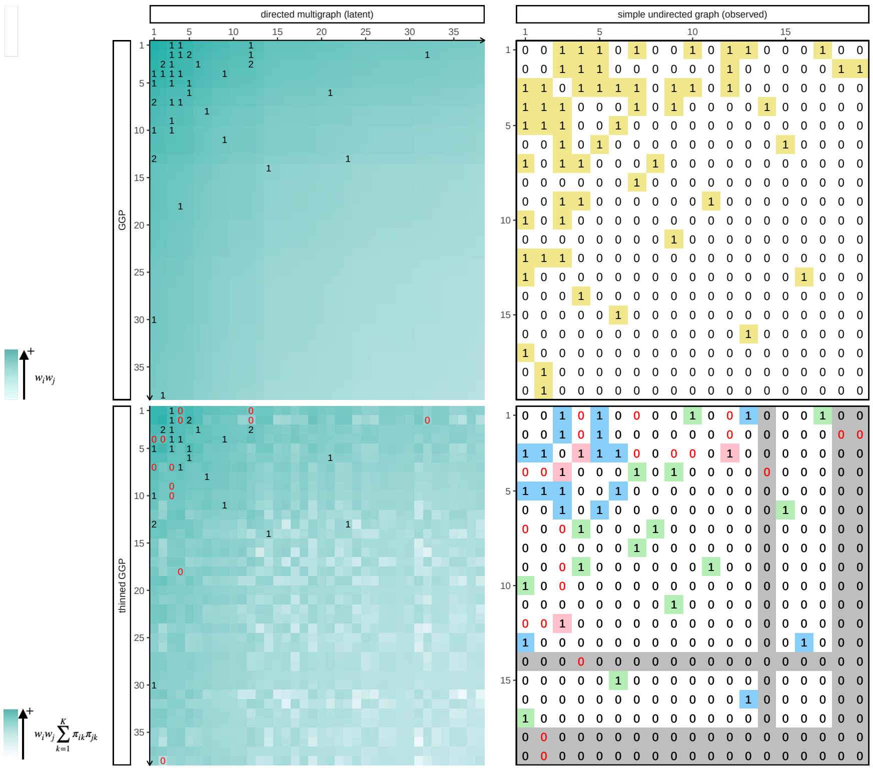


Figure 1: Generation of an adjacency matrix from the GGP model of Caron and Fox [25] (top) and via thinning via our proposed model with $K = 3$ communities (bottom). Rows and columns correspond to nodes in all plots, ordered by decreasing sociability w_i. *Left:* Latent directed multigraphs underlying the observed graphs on the right. The intensity of a cell is proportional to the Poisson rate: $w_i w_j$ for the GGP, and $w_i w_j \sum_{k=1}^{K} \pi_{ik}\pi_{jk}$ for the thinned GGP. Red zeros indicate thinned edges. *Right:* Binary adjacency matrix of the observed graph for the GGP (top) and thinned GGP (bottom) models. For the thinned GGP, edges are colored according to the community that generated it. Gray cells mark nodes whose edges in the latent multigraph were all thinned, and are thus observed under the GGP but unobserved under the thinned GGP. GGP hyperparameters were set as $\sigma = 0.1$, $\tau = 1$, $\alpha = 10$, and community memberships were sampled given $\beta = (1/3, 1/3, 1/3)$, $\zeta = 1$.

community frequency vectors $\beta = (\beta_1, \ldots, \beta_K)$ with some elements $\beta_h \approx 0$, $h \in \{1, \ldots, K\}$. The hierarchical Dirichlet process [6, 7] is the limit of this prior as K approaches infinity [34].

2.2 Sparse Network Models via Completely Random Measures

For any fixed community frequencies β and interaction probabilities η, the SBM and MMSB may only generate *dense* graphs where the number of edges scales quadratically with the number of nodes [18]. In contrast, many real-world networks appear to be sparse [15]. Heuristics are often used to fit (mixed membership) SBMs to large but sparse networks, such as fixing (rather than learning) $\eta_{k\ell} = \varepsilon \approx 0$ for $k \neq \ell$; for example Kim et al. [7] fix $\varepsilon = e^{-30}$. We seek to build models that can capture mixed memberships and sparsity simultaneously without needing to rely on such heuristics.

Completely Random Measures. By representing the graph as a point process on the plane, Caron and Fox [25] showed that it is possible to generate sparse graphs by appropriately choosing the mean measure of the point process. According to their model, for $i \neq j$,

$$Y_{ij} \mid w_i, w_j \overset{\text{ind}}{\sim} \text{Bernoulli}\left(1 - \exp\left\{-2 w_i w_j\right\}\right), \tag{2}$$

where $w_i > 0$ represents the *sociability* of node i: nodes with higher w_i have higher probability to interact with other nodes, and hence greater expected degree. Differently from the (mixed membership) SBMs, where all parameters governing nodes' interactions are sampled independently, here nodes' sociabilities are generated altogether using the jumps of a *completely random measure* (CRM) [35]. Each node is also independently associated to a real-valued location ℓ_i uniformly distributed on the real line, and then a set of potential nodes is defined by restricting to sociabilities with a sampled location in the interval $[0, \alpha]$,

$$W_\alpha = \{w_i : \ell_i \in [0, \alpha]\}. \tag{3}$$

Here $\alpha > 0$ controls the (random) number of nodes N_α in the network by determining the size of the interval in which jumps are included. Depending on the distribution of the jumps, the model can capture both sparse and dense graphs. Intuitively, as detailed in Sec. 5.1 of [25], for sparse networks the distribution of the jumps needs to place almost all of its mass near zero.

This model requires CRMs for which the sum $\bar{W}_\alpha = \sum_{i:\ell_i \in [0,\alpha]} w_i$ of the jumps in $[0, \alpha]$ is finite. A simple undirected network is then generated via a binary projection of an underlying directed multigraph, where the total number of edges n_{ij} from node i to node j is independently distributed as

$$n_{ij} \mid w_i, w_j \overset{\text{ind}}{\sim} \text{Poisson}(w_i w_j). \tag{4}$$

An edge is present in the undirected network if there is at least one directed edge in the latent directed multigraph, that is $Y_{ij} = \mathbb{1}\,(n_{ij} + n_{ji} \geq 1)$ for nodes $i \neq j$. Because the sum of independent Poisson random variables is Poisson, Eq. (4) implies $P(n_{ij} + n_{ji} = 0) = \exp\{-2w_i w_j\}$, from which Eq. (2) follows. The number N of nodes in the observed network then equals the number N_α of nodes that have at least one edge in the underlying multigraph. This construction of the binary matrix from a multigraph [25] is visualized in Fig. 1.

The sum-property of the Poisson distribution also implies that, given $\bar{W}_\alpha$, the total number of edges $\bar{D}_\alpha$ in the multigraph has a Poisson$(\bar{W}_\alpha^2)$ distribution. Since n_{ij} has a high probability of being 0 for most node pairs (i, j), it is more efficient to first sample $\bar{D}_\alpha$ and then independently assign each edge to a pair of nodes based on their sociabilities. In more detail, Eq. (4) is equivalent to

$$\bar{D}_\alpha \mid \bar{W}_\alpha \sim \text{Poisson}(\bar{W}_\alpha^2), \quad P(x_{ev} = i \mid W_\alpha) = \frac{w_i}{\bar{W}_\alpha}, \quad n_{ij} = \sum_{e=1}^{\bar{D}_\alpha} \mathbb{1}(x_{e1} = i)\mathbb{1}(x_{e2} = j), \tag{5}$$

where $x_{ev} \in \{1, 2, \dots\}$ for $v = 1, 2$ indicates the nodes sampled for edges $e \in \{1, \dots, \bar{D}_\alpha\}$. Caron and Fox [25] propose the *generalized gamma process* (GGP) [36, 37, 38] as a flexible but tractable CRM for W_α, with parameters $\tau \in (0, \infty)$, and $\sigma \in (-\infty, 0]$ for dense graphs or $\sigma \in (0, 1)$ for sparse graphs. Fig. 2 (black component) provides a visual summary of the generative process underlying the construction of a GGP random graph, using a directed graphical model that highlights the conditional dependencies (and independencies) between its variables. Fig. 3a shows a network sampled from the GGP model by means of a node-edge diagram.

Sparse Block Models. A limitation of the framework described above is that it does not model the community (block) structure of the network – a well-recognized feature of complex networks. Herlau et al. [28] generalized the approach in [25] to accommodate both sparse and dense networks with community structure. Specifically, they introduce a latent discrete variable $c_i \in \{1, \dots, K\}$ to indicate the assignment of node i to one of K communities, like in the SBMs. A bivariate CRM incorporates both the sociability weights and a set of parameters, denoted by $\eta_{c_i c_j}$, capturing the interaction strength between two communities c_i and c_j in the underlying multigraph. For their formulation, the total number of edges n_{ij} from node i to node j are independently distributed as $n_{ij} | c_i, c_j, w_i, w_j \overset{\text{ind}}{\sim} \text{Poisson}(\eta_{c_i c_j}\, w_i\, w_j)$, and the likelihood from Eq. (2) of the observed nodes is modified as $Y_{ij} \mid c_i, c_j, w_i, w_j \overset{\text{ind}}{\sim} \text{Bernoulli}(1 - e^{-2\eta_{c_i c_j} w_i w_j})$. We refer to this approach as the *stochastic block model generalized gamma process* (SBM-GGP).

Sparse Mixed Membership. The SBM-GGP models sparse and dense networks with community structure, but does not capture overlapping community structures; it induces networks whose nodes are partitioned into disjoint communities. In follow-up work, Todeschini et al. [29] extended the CRM-based framework discussed above by associating a vector $(w_{i1}, \dots, w_{iK})$ of sociabilities to each node i, to represent different levels of affiliation of nodes to the K latent communities. A node may have high levels of affiliation to more than a community, leading to the formation of edges across multiple communities. The vectors of node sociabilities are distributed according to a compound CRM [39]; specifically, their implementation uses a *compound generalized gamma process* (CGGP). The likelihood function is modified accordingly, by independently sampling undirected edges as $Y_{ij} \mid w_i, w_j \overset{\text{ind}}{\sim} \text{Bernoulli}(1 - \exp\{-2 \sum_{k=1}^{K} w_{ik} w_{jk}\})$. The latent community weights are modeled

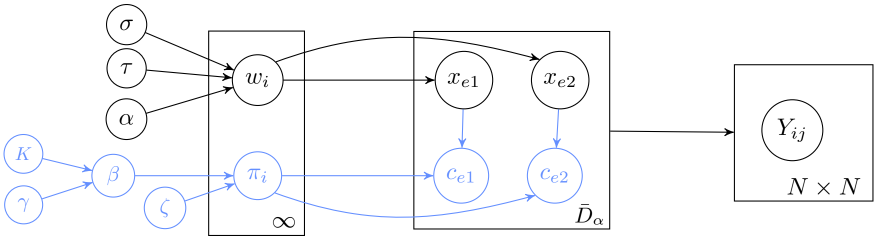


Figure 2: Directed graphical model representing the GGP random graphs of Caron and Fox [25] (black), and the additional variables (blue) in our thinned GGP. For GGP hyperparameters that induce sparse graphs ($\sigma > 0$) there is an infinite number of *potential* nodes, with sociabilities w_i and community memberships π_i.

as $w_{ik} = w_{i0}\psi_{ik}$, where w_{i0} are sampled from a GGP as in Eq. (3), and ψ_{ik} are gamma-distributed random variables $\psi_{ik} \overset{\text{ind}}{\sim} \text{Gamma}(a_k, b_k)$ whose shape a_k and rate b_k may be inferred from data.

As [29] point out, the CGGP model exploits an (unconstrained) non-negative matrix factorization to define the Bernoulli probability link. In the Appendix, we demonstrate that this lack of constraints hampers the ability of the CGGP model to recover the true community structure of the network. Better regularized approaches to non-negative matrix factorization, like the one we propose, may improve the identifiability of latent community structures and network parameters [40].

3 The Thinned CRM Network Model

We now add overlapping community memberships to the GGP network model [25] via vectors of probabilities sampled from a hierarchical Dirichlet distribution. Unlike the formulation of Todeschini et al. [29], our model enables learning the number of communities from data (see Secs. 2.1 and 4). By using probability vectors rather than unconstrained non-negative values to model community memberships, our model provides a regularized approach to inference in the GGP model, which in simulations (Sec. 5.1) provides a substantial increase in community detection accuracy.

Let each node i have both a sociability parameter w_i from the GGP as in (3), and a vector of probabilities $\pi_i = (\pi_{i1}, \ldots, \pi_{iK})$ drawn from a hierarchical Dirichlet distribution as in (1). Moreover, let the number of *potential* edges between nodes i and j in the latent multigraph depend only on their sociabilities, as in (4). For each of the n_{ij} potential edges, node i independently samples a community from $\text{Cat}(\pi_i)$, and node j samples a community from $\text{Cat}(\pi_j)$. If these community assignments match, the edge is retained; otherwise, it is *thinned* (i.e., discarded). See Figs. 1 and 3 for examples. More formally, let $\dot{n}_{ij}$ be the number of multigraph edges between a pair of nodes i, j that is retained (not thinned). Edges $e \in \{1, \ldots, \bar{D}_\alpha\}$ in the GGP multigraph are stochastically thinned as follows:

$$c_{e1} \mid x_{e1}, (\pi_1, \pi_2, \ldots) \overset{\text{ind}}{\sim} \text{Cat}(\pi_{x_{e1}}), \qquad c_{e2} \mid x_{e2}, (\pi_1, \pi_2, \ldots) \overset{\text{ind}}{\sim} \text{Cat}(\pi_{x_{e2}}), \tag{6}$$

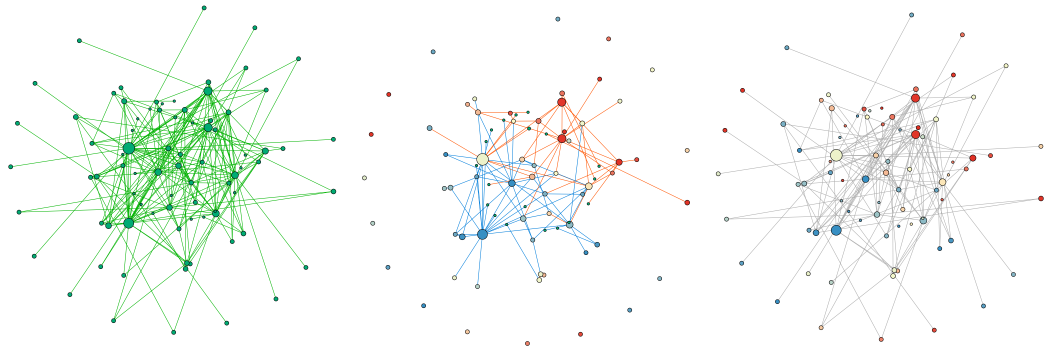
$$\dot{n}_{ij} = \sum_{e=1}^{\bar{D}_\alpha} \mathbb{1}(x_{e1} = i, x_{e2} = j, c_{e1} = c_{e2}), \qquad Y_{ij} = \mathbb{1}\left(\dot{n}_{ij} + \dot{n}_{ji} \geq 1\right). \tag{7}$$

Here $x_{ev} \in \{1, 2, \ldots\}$ for $v = 1, 2$ indicate the nodes associated with edge e as in Eq. (5). An edge Y_{ij} is then present in the undirected graph if and only if at least one multigraph edge is not thinned. Equivalently, Eq. (7) implies that the observed undirected graph is a binary projection of the multigraph edges that are *selected*, i.e. retained after the thinning process.

To determine the distribution of Y_{ij}, note that the probability that nodes i and j are assigned to the same community (marginalizing across communities) equals $P(c_{ei} = c_{ej}) = \sum_{k=1}^{K} \pi_{ik}\pi_{jk}$. Therefore, marginalizing over the latent multigraph, $\dot{n}_{ij} \mid w_i, w_j, \pi_i, \pi_j \overset{\text{ind}}{\sim} \text{Poisson}(w_i w_j \sum_{k=1}^{K} \pi_{ik}\pi_{jk})$. The likelihood of the observed network thus equals

$$Y_{ij} \mid w_i, w_j \overset{\text{ind}}{\sim} \text{Bernoulli}\left(1 - \exp\left\{-2 w_i w_j \sum_{k=1}^{K} \pi_{ik}\pi_{jk}\right\}\right). \tag{8}$$

Unlike the MMSB and GGP models, our *thinned generalized gamma process* (TGGP) model favors edges between nodes that have *both* large sociabilities and similar community memberships. The TGGP model is summarized graphically in Fig. 2 and illustrated in Figs. 1 and 3.



(a) Potential edges (b) Selected edges (c) Thinned edges

Figure 3: Visualization of networks simulated via a (thinned) GGP with $\alpha = 15, \sigma = 0.2, \tau = 1$. (a) Before thinning, *potential* edges are proposed according to nodes' sociabilities as in [25]. (b) After thinning, *selected* edges have colors (blue/red) corresponding to their assignment to $K = 2$ communities. (c) The *thinned* edges (gray) are those for which the connected nodes were assigned to different communities. In both (b) and (c), node colors represent their true community memberships, where lighter colors indicate more balanced memberships. Node sizes are proportional to betweenness centrality, and layout is determined by Gephi's Force Atlas 2 [41].

4 Monte Carlo Posterior Inference

In our TGGP model, the latent (unthinned) multigraph has a distribution (Eq. (5)) that depends only on the node sociability parameters w_i. The thinning process (Eq. (7)) then depends only on the community membership vectors π_i. As can be deduced from the Markov properties of the directed graphical model in Fig. 2, given these latent multigraphs, the posterior distributions of $(w_1, w_2, \dots)$ and $(\pi_1, \pi_2, \dots)$ are conditionally independent. This design simplifies posterior inference, and in particular allows us to apply Theorem 6 of [25] to derive the posterior of nodes' sociabilities, provided we can condition on the (latent) multigraph. We thus implement a Gibbs sampling strategy for posterior inference where we sample both thinned and retained edges in the latent multigraph, as well as their community assignments, to allow for efficient updates of other variables.

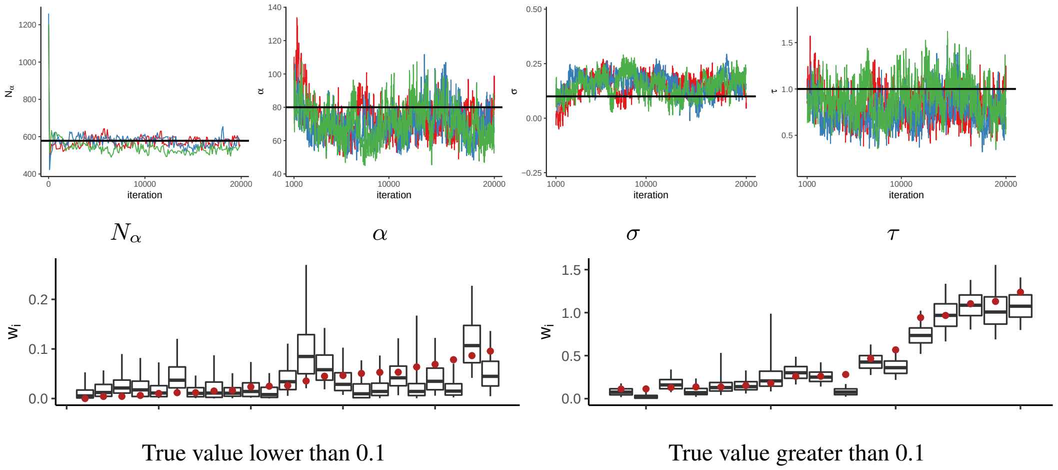
Some steps in the sampler are straightforwardly derived, considering that we can borrow the approach detailed in [25] to update node sociabilities and GGP hyperparameters (see the Appendix for a detailed outline). However, the implementation of the sampler requires some additional careful development – which we detail below – to sample the latent multigraph, due to the unobserved thinning process. Multigraph sampling is made more computationally efficient via model properties that are unique to sparse, as opposed to dense, stochastic block models. Letting (filled dot) $\dot{n}_{ij}$ denote the number of unthinned (i.e. selected) multigraph edges as in (7), and (empty dot) $\mathring{n}_{ij} = n_{ij} - \dot{n}_{ij}$ denote the number of thinned multigraph edges, we sample the latent multigraph as follows.

Sampling of $\dot{\mathbf{n}}_{\mathbf{ij}}$: From Eq. (7), there exists an edge in the observed graph if and only if there is at least one unthinned edge between nodes i and j in the latent multigraph. Thus rather than considering each pair i, j, we only need to sample $\dot{n}_{ij}$ when $Y_{ij} = 1$, an operation with cost linear in the number of observed (undirected) edges. Conversely, $Y_{ij} = 0$ implies $\dot{n}_{ij} = \dot{n}_{ji} = 0$. Since there must be at least one unthinned edge when $Y_{ij} = 1$, the posterior follows a *zero-truncated* Poisson distribution:

$$(\dot{n}_{ij} + \dot{n}_{ji}) \mid Y_{ij} = 1, w_i, w_j, \pi_i, \pi_j \sim \text{Zero-Trunc-Poisson}\left(2 w_i w_k \sum_{k=1}^{K} \pi_{ik}\pi_{jk}\right).$$

For each unthinned edge we sample a single community assignment, since by definition these are the edges whose nodes have been assigned to the same communities. We can sample the community assignment of an unthinned edge between i and j easily as a draw from $\text{Cat}(\pi_{i1}\pi_{j1}, \dots, \pi_{iK}\pi_{jK})$.

Sampling of $\mathring{\mathbf{n}}_{\mathbf{ij}}$: By construction, since thinned edges are unobserved, all pairs of nodes (i, j) may have edges that are thinned. These edges are auxiliary variables necessary to obtain the full conditional posterior distribution of W_α. Also, to update community memberships and global frequencies, it is necessary to assign every thinned edge to a pair of discordant (non-matching) communities. Thanks to the properties of Poisson processes, we do not need to sample $\mathring{n}_{ij}$ for all the pairs of nodes; we



N_α α σ τ

True value lower than 0.1 True value greater than 0.1

Figure 4: Validation of the MCMC sampler on a graph with $K = 15$ communities, and $N = 386$ observed nodes, simulated from the TGGP. *Top*: Traceplots of N_α, α, σ, and τ. Red, blue, and green indicate three distinct MCMC chains. Black horizontal lines mark the true values for the simulated graph. *Bottom*: True simulated values (red) and 95% credible intervals for the sociability parameters w_i of 40 randomly selected nodes.

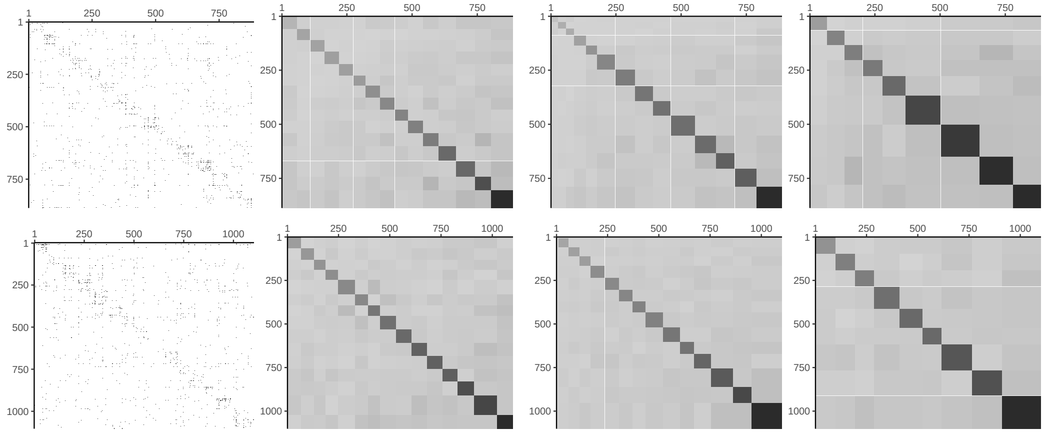
can develop a more efficient sampler that exploits the thinned model construction. We first sample a *proposed* total number of thinned edges $\mathring{N} \sim \text{Poisson}\left(2\sum_{i,j=1}^{N} w_i w_j\right)$. We then assign each of these independently to a pair of nodes according to their sociabilities as in Eq. (5), and subsequently to a pair of commmunities given the memberships of the assigned nodes as in Eq. (6). Finally, we determine $\mathring{n}_{ij}$ from the number of edges that were assigned to discordant communities.

Sampling unobserved nodes: As illustrated in Figs. 1 and 3, the total number of nodes N_α with at least one edge in the *unthinned* multigraph is likely to be different from the total number of nodes with at least one edge *after* thinning and – as a consequence – from the total number of nodes N in the observed binary graph Y. There is thus some latent number $N_\alpha - N \geq 0$ of nodes whose edges with the observed nodes have all been thinned. In order to learn this number, our MCMC sampler includes an approximate update of N_α according to the mean ratio N_α/N from graphs simulated from the GGP prior given the latest samples of the hyperparameters α, σ, τ. We found that this approximate method leads to convergence of the empirical MCMC-based estimate of N_α across all simulated and real data that we considered (see Fig. 4). Sampling of the thinned edges from these unobserved nodes introduces some additional complications that we efficiently resolve in the Appendix.

Sampling community memberships π_i and global frequencies β: From the conjugacy of Dirichlet priors to categorical likelihoods, node community memberships π_i have closed-form Dirichlet posteriors $\pi_i \mid M_i, \beta \overset{\text{ind}}{\sim} \text{Dirichlet}(\zeta\beta_1 + M_{i1}, \ldots, \zeta\beta_K + M_{iK})$, where M_{ik} equals the number of edges that node i participated in while being assigned to community k. Given community assignments of nodes in the latent multigraph, the global community frequencies β may be efficiently resampled using auxiliary-variable methods developed for the hierarchical Dirichlet process [6] (see Appendix). By learning β, node community memberships π_i become sparse, placing significant probability mass only on a data-dependent subset of the full set of K communities allocated in Eq. (1).

Computational complexity: For a K-community TGGP model of a graph with E observed edges, resampling the latent multigraph requires $\mathcal{O}(EK)$ operations. (The thinned multigraph has fewer edges, and thus its resampling increases costs by a small constant.) Closed-form resampling of community memberships is faster, requiring only $\mathcal{O}(NK)$ operations for an N-node graph. The CGGP model [29] uses Hamiltonian Monte Carlo proposals with similar cost, but empirically the CGGP sampler mixes slower and has inferior performance (see experiments). By exploiting sparse matrices and parallelization, both samplers may be scaled to networks with tens of thousands of nodes; very-large networks may require alternative approximate inference algorithms.

MCMC sampler validation: We evaluate the proposed posterior inference method on a graph with 15 communities simulated from the TGGP model. The simulated graph had 1571 undirected, community-concordant edges, $N = 386$ observed nodes (i.e., with at least one community-concordant edge), and $N_\alpha - N = 192$ unobserved nodes (i.e., with only community-discordant or self edges).



(a) Adjacency Matrix (b) True Blocks (c) Estimated, TGGP (d) Estimated, CGGP [29]

Figure 5: Results on two simulation experiments: a graph generated with the CGGP [29] (top) and a graph simulated with the proposed TGGP (bottom). Column (a) shows the true adjacency matrices, sorting nodes into blocks according to the strongest community membership; (b) shows the relative edge density in the true blocks, sorting blocks by intensity. Columns (c) and (d) show the blocks estimated, respectively, with the TGGP and the CGGP. In both simulations, we generate data setting $\alpha = 250$, $\sigma = 0.1$, $\tau = 1$, and $K = 15$. For the CGGP we set $a_k = 1/K$ and $b_k = 1$; for the TGGP we set $\beta = (1/K, ..., 1/K)$ and $\zeta = 1$. Inference uses the true number of communities for the CGGP, while our TGGP is given a loose upper bound of $K = 50$, $\gamma = 10$, $\zeta = 0.2$. The TGGP closely approximates the true community structure, while the CGGP consistently underestimates the true number of communities.

Fig. 4 (top) shows traceplots from three separate MCMC chains, each of which was run for 20,000 iterations. The traceplot of the total number of nodes (observed + unobserved) N_α shows all values sampled since initialization to demonstrate that even very different initializations led the three MCMC chains to quickly concentrate around the same (true) value of N_α. The traceplots of the GGP hyperparameters α, σ, and τ show samples after convergence (which took roughly 1000 iterations) and demonstrate that the posterior distribution effectively concentrated around the true values of these hyperparameters. Fig. 4 (bottom) shows summaries of the posterior distributions of nodes' sociabilities for 40 randomly selected nodes spanning from low to relatively large sociability, suggesting that the TGGP model also effectively recovers nodes' sociability parameters.

5 Experimental Results

5.1 Simulation

We discuss a simulation study where we investigated the performance of our proposed TGGP model, and the CGGP model of [29], based on simulated data generated from either model.

First, we present the results from a sparse graph with 15 communities, simulated from our TGGP model by setting $\alpha = 250$, $\sigma = 0.1$, $\tau = 1$ for the distribution of node sociabilities and $\gamma = 10$, $\zeta = 0.2$ for the distribution of node community memberships. The adjacency matrix of the resulting undirected graph is shown in Figure 5(a, bottom). Figure 5(b, bottom) sorts the simulated nodes into blocks according to their main membership and plots the density of edges in each block, demonstrating that the simulated graph has a clear block structure. We then run our MCMC sampler for 50,000 iterations, discarding the first 40,000 samples as burn-in. For our model fitting, we let 50 be the upper bound to the number of communities and we set $\gamma = 10$ and $\zeta = 0.5$ to allow for learning the number of communities. From a qualitative comparison of the (b) and (c) bottom panes in Figure 5, we see that the community memberships recovered from our model are close to the underlying truth. The credible intervals indicate that the posterior distributions of all parameters tend to concentrate around the true values used for simulation (see Appendix for plots). In contrast, the CGGP model [29] struggles to recover the true community structure of the network (Fig. 5(d bottom)), despite being fitted with the number of communities $K = 15$ set to match the truth. We also simulated a graph from the CGGP model of [29] with parameters $a_k = 1/K, b_k = 1$ chosen to induce a community membership prior similar to our TGGP. We fit both the CGGP model and the TGGP model to the generated data. When fitting with the CGGP we still set the number of communities equal to the truth ($K = 15$). The results plotted in Fig. 5(d) indicate that the block structures learned by our (misspecified) TGGP model are nevertheless closer to truth than those recovered by the CGGP.

To quantify the superior community recovery of our TGGP model, recall that for each observed node the CGGP estimates a K-dim. positive real vector of community-specific sociabilities $(w_{i1}, \dots, w_{iK})$, while our TGGP infers a K-dim. probability vector of community memberships $(\pi_{i1}, \dots, \pi_{iK})$. For quantitative comparison, we standardize true and inferred sociability vectors by dividing each w_{ik} by the sum $\sum_{h=1}^{K} w_{ih}$. Also, since the TGGP model was fitted with K set to an upper-bound of 50 rather than to the true number of communities, when analyzing TGGP results we pad the true vectors of community memberships (for TGGP-simulated data) and sociabilities (for CGGP-simulated data) with zeros. For estimated parameters, community indices are then permuted to maximize similarity with true parameters. Letting ψ_i and $\hat{\psi}_i$ be, respectively, true and inferred community memberships/sociabilities, we define similarity as $1 - \frac{1}{N} \sum_{i=1}^{N} \mathrm{d}(\psi_i, \hat{\psi}_i)$ where the distance is the L1 total variation $\mathrm{d}(\psi_i, \hat{\psi}_i) = \frac{1}{2} \sum_{k=1}^{K} |\psi_{ik} - \hat{\psi}_{ik}|$. Fig. 6 confirms that the TGGP more accurately recovers community memberships for both networks, and for nodes of both high and low degree.

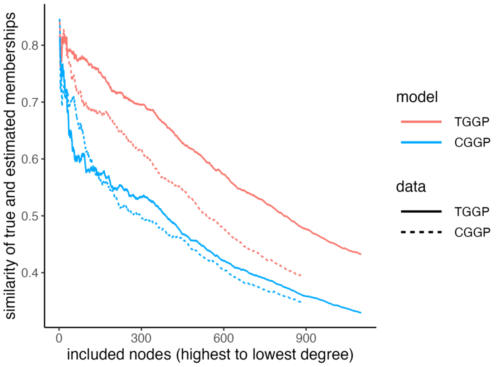


Figure 6: Plot of similarity between true and inferred community memberships for networks simulated (*data*) and fitted (*model*) with the TGGP and CGGP [29]. Our TGGP model uniformly achieves higher similarities.

5.2 Real network data

We compare the performance of our TGGP framework for adding overlapping communities to the GGP model with the approaches detailed in Sec. 2.2 and 2.1: the CGGP (sparse with mixed memberships) [29], the SBM-GGP model (sparse with single memberships) [28], and the more classical SBM (dense with single membership) [2] and MMSB (dense with mixed memberships) [5]. We run 50,000 iterations of the five models' MCMC on four real-world networks; see Appendix for data sources and pre-processing. Each model was fit to fully observed data to learn node-specific parameters (e.g., sociabilities and community memberships) and community-interaction probabilities, using the values from the last MCMC iteration. Two different measures of posterior predictive accuracy [42] were then used to assess the goodness of model fit.

For the first evaluation measure, we randomly selected 5% of the entries equal to 1 in the adjacency matrix and set them to 0. We then compute the probability of being 1 of all entries that are equal to 0 in the modified adjacency matrix to assess how well each model can distinguish the entries that should in fact be 1. This edge-retrieval measure is motivated by common applications of network analysis to recommendation tasks (suggesting which edges that are not observed should be present). Fig. 7 (top) plots recall (proportion of ones recovered) on the horizontal axis, and F-score (geometric mean of recall and precision, which is the proportion of ones correctly predicted among all entries predicted as ones) on the vertical axis. This retrieval task is very challenging because, for sparse networks, the missing edges are a tiny proportion of all zero entries in the modified adjacency matrix.

For the second evaluation measure, we randomly obscure 5% of *all* entries in the adjacency matrix to assess how well each model can recover whether an entry is equal to 0 or to 1. Fig. 7 (bottom) plots the receiver operating characteristic (ROC) curves, where false positive rate (number of zeros predicted as ones divided by the number of zeros) is on the horizontal axis, and true positive rate (proportion of ones correctly predicted as ones divided by the number of ones) is on the vertical axis. This second task has been used in many prior papers but is not as hard as the first one, because for a model to score well it suffices to learn small interaction probabilities, since almost all of the randomly selected entries are 0. Fig. 7 shows that our TGGP model does consistently better than all baseline methods for all networks considered, according to both evaluation measures.

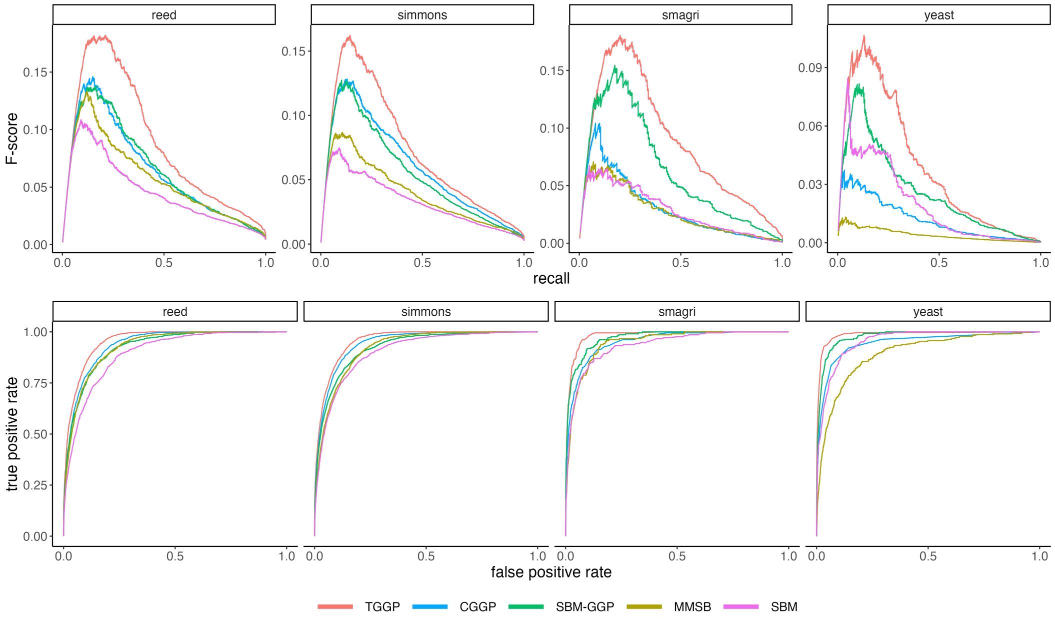


Figure 7: F-score versus recall for retrieval of missing edges (top), and ROC curves (bottom) for prediction of missing adjacency matrix entries, for four real-world networks of varying sparsity. Our TGGP uniformly outperforms the CGGP [29], SBM-GGP [28], MMSB [5], and SBM [2].

6 Discussion

We have proposed a framework for the analysis of binary network data that extends the GGP-based model of Caron and Fox [25] by allowing for overlapping community structures, depending both on the overall sociability of the nodes *and* the similarity of their community memberships. Our generative model uses a novel latent multi-graph framework where nodes can connect within the same or across different communities, but the edges formed across different communities are hidden (thinned) in the projection giving rise to the observed network. In contrast with alternative extensions of the original GGP network model [28, 29], our thinned generalized gamma process (TGGP) enables mixed memberships and facilitates regularization of the community distributions for each node, resulting in improved inference and reconstruction of the latent community structures. Also, our model allows for learning the number of communities directly from the data, by encouraging recovery of a set of non-empty communities smaller than a specified upper bound.

Monte Carlo inference for the TGGP scales linearly with the number of observed edges, and thus sub-quadratically with the number of nodes in sparse graphs. This leads to strong edge prediction performance for social and biological networks of moderate size. However, further inference innovations may be needed to scale to very-large networks with hundreds of thousands of nodes.

The proposed TGGP model is amenable to further extensions. As the adopted mixed membership framework relies on the well-studied Dirichlet-multinomial specification, the vast literature on hierarchical Dirichlet mixtures can be leveraged to capture more complex networks. It is possible to include additional node covariates or metadata to guide the allocation of nodes to communities. Finally, distributional and asymptotic properties of our model can be investigated by exploiting results related to finite mixtures and mixtures of finite mixtures [see, e.g., 43, 44, 45, 34].

Acknowledgments and Disclosure of Funding

This research supported in part by NSF CAREER Award No. IIS1758028, and by the HPI Research Center in Machine Learning and Data Science at UC Irvine.

References

[1] Anna Goldenberg, Alice X Zheng, Stephen E Fienberg, Edoardo M Airoldi, et al. A survey of statistical network models. *Foundations and Trends® in Machine Learning*, 2(2):129–233, 2010.

[2] Yuchung J Wang and George Y Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19, 1987.

[3] Charles Kemp, Joshua B Tenenbaum, Thomas L Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, volume 3, page 5, 2006.

[4] David Blackwell and James B MacQueen. Ferguson distributions via pólya urn schemes. *The Annals of Statistics*, 1(2):353–355, 1973.

[5] Edo M. Airoldi, David Blei, Stephen Fienberg, and Eric Xing. Mixed membership stochastic blockmodels. *Advances in Neural Information Processing Systems*, 21, 2008.

[6] Yee Teh, Michael Jordan, Matthew Beal, and David Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, December 2006.

[7] Dae Il Kim, Prem K Gopalan, David Blei, and Erik B. Sudderth. Efficient online inference for Bayesian nonparametric relational models. *Advances in Neural Information Processing Systems*, 26, 2013.

[8] Kurt Miller, Michael Jordan, and Thomas Griffiths. Nonparametric latent feature models for link prediction. *Advances in Neural Information Processing Systems*, 22, 2009.

[9] Konstantina Palla, David A. Knowles, and Zoubin Ghahramani. An infinite latent attribute model for network data. *Proc. 29th Int. Conf. Machine Learning*, page 395–402, 2012.

[10] Zoubin Ghahramani and Thomas Griffiths. Infinite latent feature models and the Indian buffet process. *Advances in Neural Information Processing Systems*, 18, 2005.

[11] Peter D Hoff. Multiplicative latent factor models for description and prediction of social networks. *Computational and Mathematical Organization Theory*, 15(4):261–272, 2009.

[12] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

[13] Pavel N Krivitsky, Mark S Handcock, Adrian E Raftery, and Peter D Hoff. Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social Networks*, 31(3):204–213, 2009.

[14] Steven H Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, 2001.

[15] Mark Newman. *Networks*. Oxford University Press, 2018.

[16] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *Nature*, 393(6684):440–442, 1998.

[17] David J Aldous. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598, 1981.

[18] Peter Orbanz and Daniel M Roy. Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2): 437–461, 2014.

[19] Mark EJ Newman. Assortative mixing in networks. *Physical Review Letters*, 89(20):208701, 2002.

[20] Hawoong Jeong, Zoltan Néda, and Albert-László Barabási. Measuring preferential attachment in evolving networks. *EPL (Europhysics Letters)*, 61(4):567, 2003.

[21] Fragkiskos Papadopoulos, Maksim Kitsak, M Serrano, Marián Boguná, and Dmitri Krioukov. Popularity versus similarity in growing networks. *Nature*, 489(7417):537–540, 2012.

[22] Prem K Gopalan, Chong Wang, and David Blei. Modeling overlapping communities with node popularities. *Advances in Neural Information Processing Systems*, 26, 2013.

[23] Brian Ball, Brian Karrer, and Mark EJ Newman. Efficient and principled method for detecting communities in networks. *Physical Review E*, 84(3):036103, 2011.

[24] Mingyuan Zhou. Infinite edge partition models for overlapping community detection and link prediction. In *Artificial Intelligence and Statistics*, pages 1135–1143. PMLR, 2015.

[25] François Caron and Emily B Fox. Sparse graphs using exchangeable random measures. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(5):1295–1366, 2017.

[26] Christian Borgs, Jennifer T Chayes, Henry Cohn, and Nina Holden. Sparse exchangeable graphs and their limits via graphon processes. *arXiv preprint arXiv:1601.07134*, 2016.

[27] Victor Veitch and Daniel M Roy. Sampling and estimation for (sparse) exchangeable graphs. *The Annals of Statistics*, 47(6):3274–3299, 2019.

[28] Tue Herlau, Mikkel N Schmidt, and Morten Mørup. Completely random measures for modelling block-structured sparse networks. *Advances in Neural Information Processing Systems*, 29, 2016.

[29] Adrien Todeschini, Xenia Miscouridou, and François Caron. Exchangeable random measures for sparse and modular graphs with overlapping communities. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(2):487–520, 2020.

[30] Sinead A Williamson. Nonparametric network models for link prediction. *The Journal of Machine Learning Research*, 17(1):7102–7121, 2016.

[31] Diana Cai, Trevor Campbell, and Tamara Broderick. Edge-exchangeable graphs and sparsity. *Advances in Neural Information Processing Systems*, 29, 2016.

[32] Harry Crane and Walter Dempsey. Edge exchangeable models for interaction networks. *Journal of the American Statistical Association*, 113(523):1311–1326, 2018.

[33] Sinead A Williamson and Mauricio Tec. Random clique covers for graphs with local density and global sparsity. In *Uncertainty in Artificial Intelligence*, pages 228–238. PMLR, 2020.

[34] Hemant Ishwaran and Mahmoud Zarepour. Dirichlet prior sieves in finite normal mixtures. *Statistica Sinica*, pages 941–963, 2002.

[35] Olav Kallenberg. *Probabilistic symmetries and invariance principles*, volume 9. Springer, 2005.

[36] Philip Hougaard. Survival models for heterogeneous populations derived from stable distributions. *Biometrika*, 73(2):387–396, 1986.

[37] Lancelot F James. Poisson process partition calculus with applications to exchangeable models and bayesian nonparametrics. *arXiv preprint math/0205093*, 2002.

[38] Antonio Lijoi, Ramsés H Mena, and Igor Prünster. Controlling the reinforcement in bayesian non-parametric mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):715–740, 2007.

[39] Jim E Griffin and Fabrizio Leisen. Compound random measures and their use in bayesian non-parametrics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(2):525–545, 2017.

[40] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *arXiv [cs.LG]*, 5:1457–1469, 2004.

[41] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: an open source software for exploring and manipulating networks. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 3, pages 361–362, 2009.

[42] Andrew Gelman, Xiao-Li Meng, and Hal Stern. Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica*, pages 733–760, 1996.

[43] Sylvia Frühwirth-Schnatter and Gertraud Malsiner-Walli. From here to infinity: sparse finite versus dirichlet process mixtures in model-based clustering. *Advances in Data Analysis and Classification*, 13(1):33–64, 2019.

[44] Jeffrey W Miller and Matthew T Harrison. Mixture models with a prior on the number of components. *Journal of the American Statistical Association*, 113(521):340–356, 2018.

[45] Subhashis Ghosal and Aad van der Vaart. *Fundamentals of Nonparametric Bayesian Inference*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2017.

[46] Alastair J Walker. New fast method for generating discrete random numbers with arbitrary frequency distributions. *Electronics Letters*, 10(8):127–128, 1974.

[47] Alastair J Walker. An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):253–256, 1977.

[48] Michael D Vose. A linear algorithm for generating random numbers with a given distribution. *IEEE Transactions on Software Engineering*, 17(9):972–975, 1991.

[49] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216–222, 1987.

[50] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

[51] Amanda L Traud, Peter J Mucha, and Mason A Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012.

[52] Andrej Mrvar and Vladimir Batagelj. Analysis and visualization of large networks with program package pajek. *Complex Adaptive Systems Modeling*, 4(1):1–8, 2016.

[53] Dongbo Bu, Yi Zhao, Lun Cai, Hong Xue, Xiaopeng Zhu, Hongchao Lu, Jingfen Zhang, Shiwei Sun, Lunjiang Ling, Nan Zhang, et al. Topological structure analysis of the protein–protein interaction network in budding yeast. *Nucleic Acids Research*, 31(9):2443–2450, 2003.

Appendix

A Monte Carlo Posterior Inference

We review the Gibbs sampling steps, some of which employ Metropolis-Hastings proposals rather than exact updates, for our thinned generalized gamma process (TGGP) model. We then describe each step in more detail.

Gibbs sampling steps: Overview

1. Update the edge counts and community assignments in the latent multigraph;
2. Update nodes' community memberships;
3. Update the global expected value of community memberships;
4. Update nodes' sociabilities;
5. Update the hyperparameters of the distribution of sociabilities;
6. Update the dimension of the latent multigraph.

A.1 Updating edge counts and community assignments in the latent multigraph

We start by recalling that, in our TGGP model, the observed simple undirected graph is a binary projection of an unobserved directed multigraph. For each edge in which a node participates in the multigraph, the node is assigned to a community. For a given (non-self) edge, if the communities that the two nodes are assigned to coincide, we say that the edge is *community-concordant*. Otherwise, if the communities differ, we say that the edge is *community-discordant*. We note there may be self-edges in the latent multigraph. However, since we assume that self-edges are not observed irrespective of community assignments, no distinction is needed between community-concordant and discordant self-edges and we refer to any of them simply as self-edges.

Two nodes have an edge in the observed undirected graph if there is at least one *community-concordant* edge between those nodes in the multigraph. *Observed nodes* are those that participate in at least one community-concordant edge. The nodes that in the multigraph are connected only within community-discordant or self-edges are *unobserved nodes*. We call N the number of observed nodes and we let $i = N+1, \ldots, N_\alpha$ denote the indices of unobserved nodes.

To easily sample from the posterior of nodes' sociabilities we need (for both observed and unobserved nodes) each node's total degree, which equals the total number of edges in the multigraph (both community-concordant and -discordant) in which the node participates. Also, to easily sample from the posterior of nodes' community memberships, we need their community assignment for each of the edges that they are connected to in the multigraph (both community-concordant and -discordant). For both purposes, it suffices to track the entries of a single $N_\alpha \times K$ matrix of summary statistics M, where M_{ik} equals the number of edges that node i participated in while being assigned to community k. The total degree of node i may then be obtained as $M_{i\cdot} = \sum_{k=1}^{K} M_{ik}$. We denote the counts of the assignments of a node i to each community (that is, row i of M) by $M_i = (M_{i1}, \ldots, M_{iK})$.

We will break down the sampling of the entries of M in three steps: first edge counts and community assignments for community-concordant edges, second community-discordant and self-edges between observed nodes, and finally community-discordant and self-edges involving unobserved nodes. See Figure 8 for an illustration.

A.1.1 Community-concordant edges

Recall that we assumed that an edge between two nodes is observed *if and only if* there is at least one community-concordant edge between those nodes in the latent multigraph. Then, the presence of an edge between nodes i and j in the observed graph tells us that $\dot{n}_{ij} + \dot{n}_{ji} > 0$, that is the sum of directed, community-concordant edges from i to j and from j to i in the latent multigraph is strictly positive. Similarly, the absence of an edge in the observed graph tells us that there is no community-concordant edge. Therefore, the number of community-concordant edges between two nodes only needs to be sampled for those pairs for which $Y_{ij} = 1$, $(i \geq j)$. An illustration is in Figure 8a. Because *a priori*, the number of community-concordant edges from i to j and from j to i have the same Poisson $\left(w_i w_j \sum_{k=1}^{K} \pi_{ik}\pi_{jk}\right)$ distribution, *a posteriori* we have

$$(\dot{n}_{ij} + \dot{n}_{ji}) \mid Y_{ij}, w_i, w_j, \pi_i, \pi_j \overset{\text{ind}}{\sim} \begin{cases} \text{0-Trunc-Poisson}\left(2w_i w_j \sum_{k=1}^{K} \pi_{ik}\pi_{jk}\right) & \text{if } Y_{ij} = 1, \\ \delta_0 & \text{if } Y_{ij} = 0. \end{cases}$$

By definition, the community assignment of the nodes involved in a community-concordant edge coincide. So we need to sample a single community assignment valid for both nodes i and j for each of the $\dot{n}_{ij} + \dot{n}_{ji}$ community-concordant edges:

$$(z_{ij1}, \ldots, z_{ijK}) \mid \dot{n}_{ij} + \dot{n}_{ji}, \pi_i, \pi_j \sim \text{Multinomial}\left(\dot{n}_{ij} + \dot{n}_{ji}, \left(\frac{\pi_{i1}\pi_{j1}}{\sum_{k=1}^{K} \pi_{ik}\pi_{jk}}, \ldots, \frac{\pi_{iK}\pi_{jK}}{\sum_{k=1}^{K} \pi_{ik}\pi_{jk}}\right)\right),$$

and we increment $M_i = (M_{i1}, \ldots, M_{iK})$ and $M_j = (M_{j1}, \ldots, M_{jK})$ accordingly.

Computational efficiency. Note that the computational cost of sampling community-concordant edges is linear in the number of observed edges.

A.1.2 Community-discordant and self-edges between observed nodes

Because community-discordant edges in the latent multigraph don't affect the observed simple graph, all pairs of observed nodes can have community-discordant edges, both those pairs i and j for which $Y_{ij} = 1$ and those pairs i' and j' for which $Y_{i'j'} = 0$.

We thus need to sample the total number $\mathring{n}_{ij} + \mathring{n}_{ji}$ of community-discordant edges for all pairs $i, j \in \{1, \ldots, N\}$ with $i \geq j$ (see Figure 8b). Because they are independent of the observed graph (conditional on nodes' sociabilities and memberships), the posterior distribution of community-discordant edges is the same as their prior:

$$(\mathring{n}_{ij} + \mathring{n}_{ji}) \mid w_i, w_j, \pi_i, \pi_j \overset{\text{ind}}{\sim} \text{Poisson}\left(2w_i w_j \left(1 - \sum_{k=1}^{K} \pi_{ik}\pi_{jk}\right)\right).$$

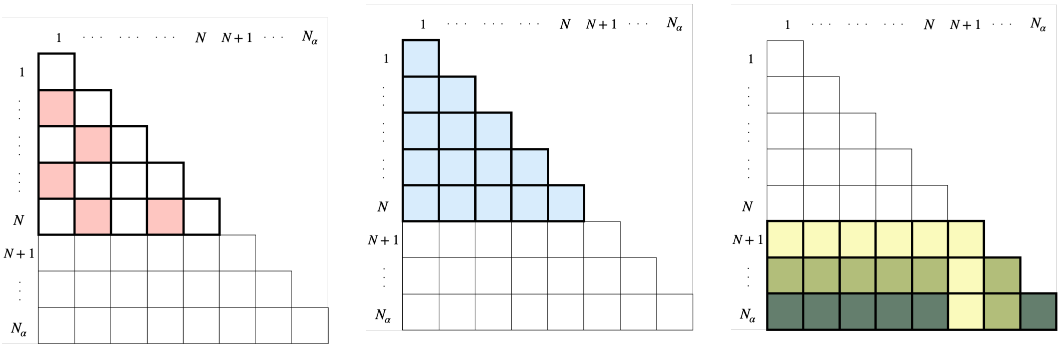
However, rather than considering every pair of nodes, we can leverage the properties of independent Poisson-distributed random variables to draw $(\mathring{n}_{ij} + \mathring{n}_{ji})$ for all pairs of nodes from the correct distribution in a more efficient way. Specifically, we can first sample an auxiliary variable $\tilde{N}_E$ from the prior distribution on the total number of edges and self-edges involving only observed nodes:

$$\tilde{N}_E \mid w_1, \ldots, w_N \sim \text{Poisson}\left(2\sum_{i=2}^{N}\sum_{j<i} w_i w_j + \sum_{i=1}^{N} w_i^2\right).$$

Then, we can proceed by associating each edge $e = 1, \ldots, \tilde{N}_E$ to a pair of nodes based only on nodes' sociabilities, and assigning each sampled node to a community based only on its own memberships:

$$x_{ev} \overset{\text{iid}}{\sim} \text{Cat}\left(\left(\frac{w_1}{\sum_{i=1}^{N} w_i}, \ldots, \frac{w_N}{\sum_{i=1}^{N} w_i}\right)\right), \qquad c_{ev} \mid x_{ev}, \pi_{x_{ev}} \overset{\text{ind}}{\sim} \text{Cat}(\pi_{x_{ev}}) \quad \text{for } v = 1, 2.$$

Finally, restricting to the resulting discordant-community edges and self-edges, we have obtained a sample for them from the correct distribution. Accordingly, we increment the counts $(M_{i1}, \ldots, M_{iK})$ of community assignments for observed nodes $i = 1, \ldots, N$, i.e. the first N rows of M.


1 ··· ··· ··· N N+1 ··· N_α
1
N
N+1
N_α
1 ··· ··· ··· N N+1 ··· N_α
1
N
N+1
N_α
1 ··· ··· ··· N N+1 ··· N_α
1
N
N+1
N_α

(a) Community-concordant edges

(b) Community-discordant edges (observed nodes)

(c) Community-discordant edges (unobserved nodes)

Figure 8: Illustration of the adjacency matrix resulting from an undirected version of the latent directed multigraph and corresponding sampling strategies. The entry located in row i and column j has the total (from i to j, plus from j to i) number for each of three types of edges. (a) *Community-concordant edges*: entries colored in pink mark pairs of nodes for which an edge is observed and for which community-concordant edges need to be sampled, as described in Sec. A.1.1. (b) *Community-discordant and self-edges between observed nodes*: all entries corresponding to observed nodes are colored in blue to emphasize that these type of edges must be sampled for all those pairs. This can be done by proposing the total across these entries first, as described in Sec. A.1.2. (c) *Community-discordant and self-edges involving unobserved nodes*: three different colors are used to mark which entries are sampled jointly according to the sequential sampling scheme described in Sec. A.1.3.

Computational efficiency. Note that, using the strategy described in this subsection, the computational cost for sampling community-discordant and self edges depends on the cost of assigning $\tilde{N}_E$ edges to a pair of nodes. Thanks to fast sampling techniques for sampling from discrete distributions like the alias method [46, 47, 48], this operation can be made to require only $\mathcal{O}(N \log N)$ computational time for pre-processing, after which edge assignments x_{ev} may be generated in constant time.

A.1.3 Community-discordant and self edges involving unobserved nodes

Let $N_\alpha - N$ be the number of nodes that are active (have at least one connection) in the latent multigraph, but *not* in the observed simple graph. Let the labels of the nodes be ordered so that $i = N+1, \ldots, N_\alpha$ correspond to the labels of these unobserved nodes. Conditional on these nodes not being observed but having connections in the latent multigraph, the information that we know is that each of them can only have community-discordant or self-edges, but also that they *must* be associated to *at least one* community-discordant or self-edge in the latent multigraph. This constraint implies that some care is required when sampling edges involving unobserved nodes. Specifically, we must generate the *edge counts and associated nodes*, that is simulate how many edges each unobserved node participates in, as well as what other nodes are associated to these edges. Given the nodes associated to an edge involving an unobserved node, community assignments can then be sampled quite easily, considering that they need to be discordant for non-self edges and that no restriction is applied to self-edges.

Edge counts and associated nodes. To correctly incorporate the constraint that each unobserved node must be associated to at least one community-discordant or self-edge, the strategy that we adopt is to sample the total number of edges and the community assignments of unobserved nodes sequentially, starting from those involving node $N+1$ with any other node (that is, the yellow entries in Figure 8c). Let s_i, for $i = N+1, \ldots, N_\alpha$, be an auxiliary indicator variable that takes value 0 if node i has been associated to an edge already and 1 otherwise. The variable s_i thus indicates if the constraint that node i must be associated to at least one edge is still active ($s_i = 1$) or is inactive ($s_i = 0$). As we start from node $(N+1)$, all constraints are active and so $s_i = 1$ for all $i = N+1, \ldots, N_\alpha$. We then sample the total number of edges $\mathring{n}_{(N+1)}$ between node $(N+1)$ and any other node from

$$\mathring{n}_{(N+1)} \mid (w_1, \ldots, w_{N_\alpha}), (\pi_1, \ldots, \pi_{N_\alpha}) \sim \text{0-Trunc-Poisson}\left(2w_{(N+1)}\mathring{p}_{(N+1)} + w^2_{(N+1)}\right),$$

$$\mathring{p}_{(N+1)} = \sum_{j=1}^{N} \mathring{p}_{(N+1)j} + \sum_{j=N+2}^{N_\alpha} \mathring{p}_{(N+1)j}, \qquad \mathring{p}_{(N+1)j} = w_j \left(1 - \sum_{k=1}^{K} \pi_{(N+1)k}\pi_{jk}\right).$$

Given the total number of edges associated with node $(N+1)$, we set $x_{e1}^{(N+1)} = i$ and associate another node (possibly node $N+1$ itself) to each edge $e = 1, \dots, \mathring{n}_{(N+1)}$, according to

$$P(x_{e2}^{(N+1)} = j) \propto \begin{cases} \mathring{p}_{(N+1)j} & \text{if } j = 1, \dots, N, N+2, \dots, N_\alpha, \\ w_i & \text{if } j = N+1. \end{cases}$$

We then set $s_{(N+1)} = 0$ and also $s_i = 0$ for every node $i \in \{N+2, \dots, N_\alpha\}$ that has been associated to some edges with node $(N+1)$.

Next, we can move on to sampling all the *remaining* edges involving node $N+2$, that is those with any other node *except* node $N+1$ (the light green entries in Figure 8c). In general, when we sample edges for node $(N+d)$ for $d = 2, \dots, (N_\alpha - N)$, there are two considerations we must keep into account: (1) the edges between node $(N+d)$ and nodes $(N+1), \dots, (N+d-1)$ have already been sampled and so those nodes should not be re-considered; (2) if $s_{(N+d)} = 0$, the constraint on node $(N+d)$ is no longer active and we may possibly sample 0 additional edges. Putting these considerations together and letting $\mathring{n}_{(N+d)}$ be the number of remaining edges to sample when considering node $(N+d)$, that is those with nodes $\{1, \dots, N, N+d, \dots, N_\alpha\}$, we have

$$\mathring{n}_{(N+d)} \mid (w_1, \dots, w_{N_\alpha}), (\pi_1, \dots, \pi_{N_\alpha}), s_{(N+d)} \sim \\ \begin{cases} \text{0-Trunc-Poisson} \left(2 w_{(N+d)} \mathring{p}_{(N+d)} + w_{(N+d)}^2\right) & \text{if } s_{(N+d)} = 1, \\ \text{Poisson} \left(2 w_{(N+d)} \mathring{p}_{(N+d)} + w_{(N+d)}^2\right) & \text{if } s_{(N+d)} = 0, \end{cases}$$

where

$$\mathring{p}_{(N+d)} = \sum_{j=1}^{N} \mathring{p}_{(N+d)j} + \sum_{j=N+d+1}^{N_\alpha} \mathring{p}_{(N+d)j}, \qquad \mathring{p}_{(N+d)j} = w_j \left(1 - \sum_{k=1}^{K} \pi_{(N+d)k}\pi_{jk}\right).$$

Then for $e = 1, \dots, \mathring{n}_{(N+d)}$, we set $x_{e1}^{(N+d)} = N+d$ and sample

$$P(x_{e2}^{(N+d)} = j) \propto \begin{cases} \mathring{p}_{(N+d)j}, & \text{if } j = 1, \dots, N, N+d+1, \dots, N_\alpha, \\ w_i & \text{if } j = N+d. \end{cases}$$

Community assignments. For every node $i = N+1, \dots, N_\alpha$ with $\mathring{n}_i > 0$, and for each $e = 1, \dots, \mathring{n}_i$, we sample a community assignment $c_e^{(i)}$ for node i given $x_{e2}^{(i)}$ from

$$P(c_e^{(i)} = k \mid \pi_i, x_{e2}^{(i)}, \pi_{x_{e2}^{(i)}}) \propto \begin{cases} \pi_{ik} \left(1 - \pi_{x_{e2}^{(i)}, k}\right), & x_{e2}^{(i)} \neq i, \\ \pi_{ik}, & x_{e2}^{(i)} = i, \end{cases}$$

for $k = 1, \dots, K$. For non-self-edges, we rescale π_{ik} accounting for the knowledge that the community to which the other node associated to edge e is assigned needs to differ from that to which node i is assigned for this edge. We increment $M_{ic_e^{(i)}}$ accordingly, and then given $c_e^{(i)}$ and $x_{e2}^{(i)}$, we can sample a community assignment $\bar{c}_e^{(i)}$ for node $x_{e2}^{(i)}$ from

$$P(\bar{c}_e^{(i)} = \ell \mid c_e^{(i)}, x_{e2}^{(i)}, \pi_{x_{e2}^{(i)}}) = \begin{cases} \dfrac{\pi_{x_{e2}^{(i)}\ell}}{(1 - \pi_{x_{e2}^{(i)} c_e^{(i)}})}, & x_{e2}^{(i)} \neq i, \ \ell \in \{1, \dots K\} \backslash c_e^{(i)}, \\ \pi_{i\ell}, & x_{e2}^{(i)} = i, \ \ell \in \{1, \dots K\}, \end{cases}$$

and then increment $M_{x_{e2}^{(i)} \bar{c}_e^{(i)}}$. Notice that for non-self-edges, we have removed the community sampled for node i from the range of available communities for the assignment of node $x_{e2}^{(i)}$, and rescaled $\pi_{x_{e2}^{(i)}}$ accordingly.

Computational efficiency. In our TGGP model, unobserved nodes will have low sociability parameters; otherwise, they would be likely to be linked to unthinned edges. Thus $\mathring{n}_{(N+d)}$ tends to be small for all $d = 1, \dots, N_\alpha - N$. For sampling their edges and community assignments, it is just necessary to compute a $(N+1) \times N_\alpha$ matrix $\mathring{P}$ whose entries $\mathring{p}_{ij}$ represent the relative probability of associating node j to an edge sampled for node i.

A.2 Updating nodes' community memberships

Here we describe the update of $\pi_i = (\pi_{i1}, \dots, \pi_{iK})$, the distribution over community memberships of node $i = 1, \dots, N_\alpha$. Recall the prior on π_i:

$$\pi_i \mid \beta \overset{iid}{\sim} \text{Dirichlet}(\zeta\beta_1, \dots, \zeta\beta_K),$$

where $\beta = (\beta_1, \dots, \beta_K)$ is a vector of community frequencies that itself has a Dirichlet prior, and equals the expected value of π_i. For our experiments we set $\zeta = 0.5$, to encourage community memberships of individual nodes to be more polarized than β and typically only use a subset of the available communities.

For each edge in the latent multigraph to which node i is associated, the generative model has node i being assigned to a community sampled independently from $(1, \dots, K)$ with probabilities $(\pi_{i1}, \dots, \pi_{iK})$, like in multinomial sampling. The Dirichlet and the multinomial are conjugate distributions, so the posterior of π_i is easily obtained. Indeed, conditional on the concentration parameter β and the $N_\alpha \times K$ matrix M whose rows record the number of times $M_i = (M_{i1}, \dots M_{iK})$ that an edge $i = 1, \dots, N_\alpha$ was assigned to each community $k = 1, \dots, K$, the posterior distribution of the vectors of community memberships are obtained as:

$$\pi_i \mid M_i, \beta \overset{\text{ind}}{\sim} \text{Dirichlet}(\zeta\beta_1 + M_{i1}, \dots, \zeta\beta_K + M_{iK}).$$

An advantage of this specification is that it allows sparse community memberships, in that the sample $(\pi_{i1}, \dots, \pi_{iK})$ may have elements that are (almost) exactly equal to zero.

A.3 Updating the global expected value of community memberships

The posterior distribution of the expected value $\beta = (\beta_1, \dots, \beta_K)$ of nodes' community memberships informs about summaries of the latent community structure of a network, such as the likely number of active communities and the relative size of different communities. Recall the prior on β:

$$\beta \sim \text{Dirichlet}\left(\frac{\gamma}{K}, \dots, \frac{\gamma}{K}\right),$$

where the value of γ controls the sparsity of the prior on β, with values of $\gamma \ll K$ being more sparsity-inducing. In our experiments, we fix $\gamma = 10$ but $K = 50$. This hierarchical prior approaches the hierarchical Dirichlet process [6] in the limit as $K \to \infty$.

Here we illustrate a strategy for sampling from the posterior of $\beta = (\beta_1, \dots, \beta_K)$ that relies on auxiliary variables drawn according to the so-called "Chinese restaurant franchise" representation of a hierarchical Dirichlet model [6]. We can represent each node $i = 1, \dots, N_\alpha$ as a restaurant, where every new community assignment for node i can be represented as a new customer entering restaurant i and ordering a dish (community) $k \in \{1, \dots, K\}$. When a new customer enters, they decide to sit at an existing table $t \in \{1, \dots, T_i\}$ with probability proportional to the number of customers currently sitting at that table (and eat the dish k_t that was ordered by the first customer who sat at the table); or, they can start a new table $T_i + 1$ with probability proportional to ζ, and pick a new dish k_{T_i+1} from the menu according to β. Let t_{ie} be the table chosen when customer e enters restaurant i, or equivalently the table chosen for the community assignment of node i for when edge e is associated to it. Then letting $T_i^{(e)}$ be the number of tables already started at restaurant i when customer e enters, and letting $q^{(e)} = (q_{i1}^{(e)}, \dots, q_{iT_i^{(e)}}^{(e)})$ be the number of customers already sitting at each of the tables when customer e enters:

$$p(t_{ie} = t \mid q^{(e)}) \propto \begin{cases} q_{it}^{(e)} & \text{if } t \in \{1, \dots, T_i^{(e)}\}, \\ \zeta & \text{if } t = T_i^{(e)} + 1. \end{cases}$$

Conditional on t_{ie} and letting $\tilde{\kappa}_t$ be the dish served at table t, the dish $\kappa_{ie} \in \{1, \dots, K\}$ ordered when customer e enters restaurant i is sampled according to

$$p(\kappa_{ie} = k \mid t_{ie} = t, \beta) = \begin{cases} 1 & \text{if } t \in \{1, \dots, T_i^{(e)}\}, \kappa_{ie} = \tilde{\kappa}_t, \\ 0 & \text{if } t \in \{1, \dots, T_i^{(e)}\}, \kappa_{ie} \neq \tilde{\kappa}_t, \\ \beta_k & \text{if } t = T_i^{(e)} + 1. \end{cases}$$

From the last sampling equation we see that the posterior distribution of β depends on how many *tables* (rather than customers) are being served each dish across all restaurants in the franchise. Let T_{ik} be the number of tables serving dish k at restaurant i and let $T_{\bullet k} = \sum_{i=1}^{N_\alpha} T_{ik}$ be the total number of tables serving dish k across all restaurants in the franchise. The summary statistics that we need for the posterior of β is the vector $T = (T_{\bullet 1}, \dots, T_{\bullet K})$.

We then sample, for every restaurant $i = 1, \dots, N_\alpha$ (i.e. for every node) and every dish $k = 1, \dots, K$ (i.e. for every community), the number of tables $T_{ik} \leq M_{ik}$ occupied by the M_{ik} customers eating dish k at restaurant i. Note that M_{ik} is the entry in row i and column k of the $N_\alpha \times K$ matrix M sampled at step 1 (see Section A.1).

Because the distribution of table arrangement is invariant to the order in which customers enter, we can assign one at a time each of the $e = 1, \dots, M_{ik}$ customers eating dish k at restaurant i to either sit at an existing table ($t_{ike} = 0$) or to start a new table ($t_{ike} = 1$) according to:

$$t_{ike} \mid \beta \sim \text{Bernoulli}\left(\frac{\zeta\beta_k}{(e-1) + \zeta\beta_k}\right),$$

where we have $\zeta\beta_k$ since we are conditioning on the customer being served dish k and we weight $\zeta\beta_k$ with respect to $(e-1)$ because, when considering customer e, there are already $(e-1)$ customers sitting at some table serving dish k. Thus, the chances that customer e sits at one of the existing tables is proportional to $(e-1)$. Finally, we set $T_{ik} = \sum_{e=1}^{M_{ik}} t_{ike}$.

After sampling the auxiliary variables, we can easily update β according to

$$\beta \mid T \sim \text{Dirichlet}\left(\frac{\gamma}{K} + T_{\bullet 1}, \dots, \frac{\gamma}{K} + T_{\bullet K}\right).$$

A.4 Updating nodes' sociabilities

Let $M_{i\bullet} = \sum_{k=1}^{K} M_{ik}$ be the total degree of node i (whose sampling was described in Section A.1). Conditional on the vector of total degrees $(M_{1\bullet}, \dots, M_{N_\alpha \bullet})$ and on the hyperparameters α, σ and τ, the update of nodes' sociabilities $(w_1, \dots, w_{N_\alpha})$ can be done via a Hamiltonian Monte Carlo step [49, 50] as detailed by Caron and Fox in Section 7.2 and in Appendix F.1 of [25].

A.5 Updating hyperparameters of sociabilities' distribution

We assume the same improper priors on the hyperparameters α, σ, and τ as in [25]:

$$P(\alpha) \propto \frac{1}{\alpha},$$
$$P(\sigma) \propto \frac{1}{1-\sigma},$$
$$P(\tau) \propto \frac{1}{\tau}.$$

To update these hyperparameters we follow the approach that Caron and Fox describe in Section 7.2 and in Appendix F.2 in [25].

A.6 Updating the dimension of the latent multigraph

Even though our interest lies in the sociabilities and community memberships of observed nodes $i = 1, \ldots, N$, to derive their exact posterior we need to also consider the unobserved nodes $N+1, \ldots, N_\alpha$, that is those nodes whose edges in the latent multigraph are all thinned, being all either discordant-community or self-edges. The distribution of the number of nodes N_α with at least one edge in the latent multigraph is not tractable, and we thus resort to an approximate method for updating N_α.

Our update of N_α is based on sampling Q multigraphs from the GGP prior evaluated at the current values of α, σ and τ. For simulated graph q, let $N^{(q)}$ be the number of nodes associated to at least one concordant-community edge, and $N_\alpha^{(q)}$ be the number of nodes associated to at least one edge (of any type). We then compute the average ratio $r_Q = \frac{1}{Q} \sum_{q=1}^{Q} \frac{N_\alpha^{(q)}}{N^{(q)}}$ and set the new value of $N_\alpha^{(\text{new})} = N r_Q$. If the new value $N_\alpha^{(\text{new})}$ is smaller than the older value $N_\alpha^{(\text{old})}$, we keep the $N_\alpha^{(\text{new})}$ unobserved nodes that currently have the largest sociabilities, along with their current community memberships. If $N_\alpha^{(\text{new})} > N_\alpha^{(\text{old})}$, we keep all of the current $N_\alpha^{(\text{old})}$ unobserved nodes, sample the sociabilities for the remaining $N_\alpha^{(\text{new})} - N_\alpha^{(\text{old})}$ nodes by resampling from the sociabilities of the current nodes, and assign community memberships from the prior evaluated at the current value of β. Our experiments on simulated data suggest that, by updating N_α every 100 iterations and setting $Q = 10$, this method allows MCMC chains to concentrate around the true value of N_α and of the hyperparameters α, σ, and τ (see Figure 4 (top) in the manuscript).

B Additional results on simulated data

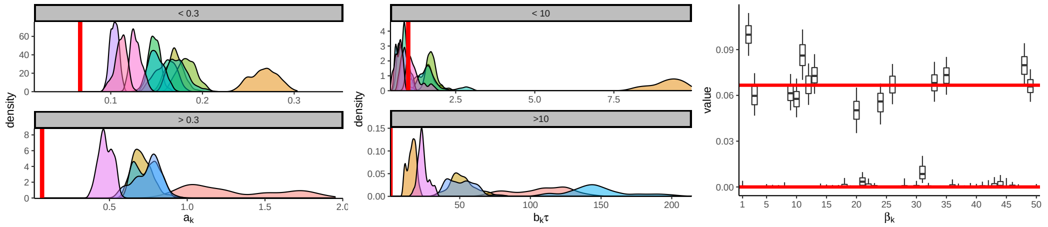
In Section 3 of the manuscript we observed that our TGGP specification is more regularized, and thus may be more easily identified, than the compound GGP formulation of [29]. Recall that, in our specification, the community memberships for node i are a probability vector π_i over K communities generated from

$$\pi_i \mid \beta \overset{\text{iid}}{\sim} \text{Dirichlet}(\zeta\beta_1, \ldots, \zeta\beta_K), \quad i = 1, \ldots, N_\alpha.$$

In contrast, for the compound GGP the membership of node i in community k is modeled as

$$w_{ik} = w_{i0}\psi_{ik}, \quad \psi_{ik} \overset{\text{ind}}{\sim} \text{Gamma}(a_k, b_k), \qquad \text{for } i = 1, \ldots, N_\alpha,\ k = 1, \ldots, K,$$

where nodes' baseline sociabilities $(w_{10}, w_{20}, \ldots)$ are drawn according to a generalized gamma process. In Section 5.1 of the manuscript, we noted how our results suggested that the TGGP model may be better at recovering the underlying community structure of both TGGP and CGGP simulated graphs. Here we provide additional insights into this comparison. Figures 9a and 9b show that, for a CGGP-simulated graph with 15 communities, $a_k = 1/15$ and $b_k = 1$ for all $k = 1, \ldots, 15$, the samples $(a_1, \ldots, a_{15})$ from the posterior estimated fitting the CGGP model are not concentrated around the true value or $1/15$, and several among $b_1\tau, \ldots, b_K\tau$ (which according to [29] is better identifiable than $b_1, \ldots, b_K$) are considerably larger than their true value of 1. Conversely, Figure 9c shows that the TGGP model fitted with $K = 50$ correctly concentrates β_k around 0 for most $k = 1, \ldots, K$ and, for the remaining communities, it tends to draw samples around the true value of $1/15$ (the simulated graph for the TGGP has 15 communities and a true value $\beta_k = 1/15$ for each of them).



(a) a_k densities (CGGP [29]) (b) $b_k\tau$ densities (CGGP [29]) (c) β_k boxplots (TGGP)

Figure 9: Summaries of the posteriors of the parameters controlling the distribution of community memberships. (a) densities of $a_1, \dots, a_K$ and (b) density of $b_1\tau, \dots, b_K\tau$ (which according to [29] is better identifiable than $b_1, \dots, b_K$) for the CGGP-simulated graph fitted with the CGGP model. Different communities correspond to different colors, red lines mark the true values, and the range of a_k and b_k are split between relatively small and large values to facilitate the plotting. (c) boxplots of the elements of the expected value β of nodes' community memberships for the TGGP-simulated graph fitted with the TGGP model; a red line marks the value $1/15$ that is the true value for 15 among the 50 fitted $(\beta_1, \dots, \beta_50)$, and another red line marks the value 0 that corrresponds to the true value for the remaining ones. Both the CGGP and TGGP models were fitted by running 50,000 iterations of the respective samplers and discarding the first 40,000 iterations as burn-in. Both the CGGP and TGGP simulated graphs had 15 underlying communities and hyperparameter values set to $\alpha = 250, \sigma = 0.1, \tau = 1$. The CGGP model was fitted setting the number of communities $K = 15$ equal to the truth, while the TGGP model was fitted with K = 50 and a sparsity-inducing prior on β.

C Real-data networks: preparation and references

The four real-data networks used in our experiments were pre-processed by extracting the main component for the networks, by removing self-edges and by making the observed network undirected. Summary information for each of these networks is displayed in Table 1.

Network name	Type	Reference	Number of nodes	Number of edges
Reed	Online social network	[51]	962	18812
Simmons	Online social network	[51]	1510	32984
SmaGri	Co-authorship network	[52]	1024	4916
Yeast	Protein interaction network	[53]	2224	6609

Table 1: Information on datasets used for real-data experiments

The networks Reed and Simmons were downloaded from https://archive.org/details/oxford-2005-facebook-matrix, the network SmaGri was downloaded from https://www.cise.ufl.edu/research/sparse/matrices/Pajek/SmaGri.html, and the network Yeast was obtained from http://vlado.fmf.uni-lj.si/pub/networks/data/bio/Yeast/Yeast.htm.