Journal of Machine Learning Research 24 (2023) 1-44 Submitted 3/22; Revised 11/23; Published 12/23

Bayesian Spanning Tree: Estimating the Backbone of the Dependence Graph

Leo L. Duan LI.DUAN@UFL.EDU
Department of Statistics, University of Florida

David B. Dunson DUNSON@DUKE.EDU
Department of Statistical Science, Duke University

Editor: Ryan Adams

Abstract

In multivariate data analysis, it is often important to estimate a graph characterizing dependence among p variables. A popular strategy in Gaussian graphical models and latent Gaussian graphical models uses the non-zero entries in a $p \times p$ covariance or precision matrix, typically requiring restrictive modeling assumptions for accurate graph recovery. To improve model robustness, we instead focus on estimating the *backbone* of the dependence graph. We use a spanning tree likelihood, based on a minimalist graphical model that is purposely overly-simplified. Taking a Bayesian approach, we place a prior on the space of trees and quantify uncertainty in the graphical model. In both theory and experiments, we show that this model does not require the population graph to be a spanning tree or the covariance to satisfy assumptions beyond positive-definiteness. The model accurately recovers the backbone of the population graph at a rate competitive with existing approaches but with better robustness. We show combinatorial properties of the spanning tree, which may be of independent interest, and develop an efficient Gibbs sampler for Bayesian inference. Analyzing electroencephalography data using a hidden Markov model with each latent state modeled by a spanning tree, we show that results are much more interpretable compared with popular alternatives.

Keywords: Graph-constrained Model, Incidence Matrix, Laplacian, Matrix Tree Theorem, Traveling Salesperson Problem.

1. Introduction

In multivariate data analysis, it is commonly of interest to make inferences on the dependence structure in a collection of p random variables. In Gaussian graphical models and latent Gaussian graphical models, the covariance matrix provides a typical summary of pairwise dependence between variables, while the inverse of the covariance or precision matrix is used to characterize conditional dependence relationships. To simplify inferences, a focus is commonly in inferring a dependence graph: $G = (V, E_G)$, with $V = \{1, \ldots, p\}$ nodes representing the p variables and $E_G = \{e_s\}$ the set of edges. If (j, k) is an edge in E_G, there is a dependence relationship between the two variables y_j and y_k.

There is a huge literature on Gaussian graphical models, encompassing different types of dependence, mostly defined through the covariance among the y_j's, Σ_0, or its inverse (precision), Ω_0. Popular examples include: assuming the variables follow a multivariate Gaussian distribution, (i) $\Sigma_{0:j,k} = 0$ implies y_j and y_k are statistically independent

©2023 Leo Duan and David Dunson.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v24/22-0252.html.

(Dempster, 1972; Cox and Wermuth, 1996) and (ii) $\Omega_{0:j,k} = 0$ implies y_k are conditionally independent given all other variables; such graphs have become very popular due to the graphical lasso (Friedman et al., 2008). (iii) using the lower-triangular decomposition of Ω_0 after some permutation $CC^{\mathrm{T}} = P\Omega_0 P^{\mathrm{T}}$ (P is a permutation matrix), those non-zero $C_{j,k}$'s give a directed acyclic graph (DAG) as a sequential data generating scheme (Rütimann and Bühlmann, 2009; Cao et al., 2019). There is a rich related literature including more complex elaborations, such as graphs that change over time (Basu and Michailidis, 2015), hyper-graphs (Roverato, 2002), copula graphical models (Dobra and Lenkoski, 2011) and Bayesian applications such as Dobra et al. (2004).

A major practical issue in inferring dependence graphs based on observations of multivariate vectors $y^{(i)} = (y_1^{(i)}, \ldots, y_p^{(i)})^{\mathrm{T}}$, for $i = 1, \ldots, n$, is that the number of possible graphs is immense for large p. For example, for covariance graphs (i), there are $2^{p(p-1)/2}$ possible graphs, which clearly increases extremely rapidly with p. This creates two practical issues. Firstly, even for moderate p, we cannot visit all possible graphs so it becomes challenging to identify the "best" graph that is most likely given the observed data. Secondly, even if we could identify one best graph, there is likely a large number of alternative graphs that are equally plausible given the observed data. Hence, whenever we estimate a dependence graph in more than a few variables, we inherently expect a large amount of uncertainty. There are many available algorithms that deal with the first problem, ranging from the graphical lasso to thresholding the empirical covariance. However, the resulting point estimates should be interpreted carefully given the second problem.

Most graphical selection procedures leverage on estimates of the covariance or precision, $\hat{\Sigma}$ or $\hat{\Omega}$. To obtain fewer errors in graph estimation, one typically needs to first achieve high accuracy in $\hat{\Sigma}$ or $\hat{\Omega}$. In large p settings, this is challenging. Cai et al. (2016b) showed that the empirical covariance $\hat{\Sigma} = S_n$ converges to the population Σ_0 in $\|\hat{\Sigma} - \Sigma_0\|_{op}^2 = O(p/n)$, with $\|.\|_{op}$ the operator norm. Hence, the sample size n may need to be substantially larger than p. To obtain an accurate estimate under the $n < p$ scenario, the true Σ_0 has to satisfy restrictions. For example, Bickel and Levina (2008a) assumed Σ_0 is sparse and proposed a simple thresholding estimator. Bickel and Levina (2008b) instead assume the off-diagonal elements in Σ_0 between y_j and y_k decay with $|j - k|$, and propose a banding/tapering estimator. Yuan and Lin (2007) and Friedman et al. (2008) supposed that the true Ω_0 is very sparse, motivating the graphical lasso (glasso). For a survey of large p covariance/precision estimators, see Cai et al. (2016c). Roughly speaking, one can recover Σ_0 (or Ω_0) at an error rate diminishing at $O(\log p/n)$ if the corresponding assumption holds for the true Σ_0. These assumptions are difficult to verify in practice and violations (e.g, true graph is dense) may lead to poor performance.

This motivates us to consider a less ambitious task — if we cannot accurately recover the whole edge set E_G, can we estimate a smaller subset, as an important summary statistic of G? Intuitively, a useful edge subset corresponds to the "backbone" graph, in which we use as few edges as possible, while connecting as many nodes as permitted. This leads us to consider a classic graph/combinatorial statistic called the "spanning tree" (Kruskal, 1956), as the smallest graph that still connects all the nodes. This tree is commonly used to solve the "traveling salesperson problem", by finding a simple travel plan that approximately minimizes the total distance traveled between p cities. Similarly, we can imagine a "minimal" generative process: starting from one variable, we sequentially generate

a new variable, that only depends on one of the existing variables. This leads to a spanning tree likelihood.

There has been a recent surge of interest in exploiting spanning trees in a Bayesian model. Teixeira et al. (2019); Luo et al. (2023) use spanning trees to produce a contiguous partition of the temporal and spatial space; Luo et al. (2021, 2022) use spanning tree as a modeling tool to accommodate irregularly shaped partition and build new nonparametric regression models on manifolds; Duan and Roy (2023) show that treating disjoint union of spanning trees as a latent graph variable leads to a generative model associated with the spectral clustering algorithms; Natarajan et al. (2023) explore edge-union of spanning trees for building new graphical priors. Despite these advanced applications, there is a lack of exposition on the fundamental properties of spanning tree that would be useful for general Bayesian researchers, and a lack of large sample theory for using spanning tree for graph estimation, which motivate this article.

We equip the spanning tree with a prior distribution, allowing us to obtain a posterior distribution over all possible spanning trees, hence quantifying model uncertainty. Importantly, in both theory and numerical experiments, we demonstrate that this model does not require the population G to be a spanning tree, nor the population covariance Σ_0 to satisfy any assumptions besides positive-definiteness; yet, it can accurately recover the backbone of the population graph G, as the minimum spanning tree transform based on Σ_0. Our theory is in the same spirit as the celebrated result of White (1982), who studied the asymptotic behavior of a restricted model estimator when the data are generated from a different full model; as well as the more recent spiked covariance model (Donoho et al., 2018) for the optimal estimation of the leading eigenvalues of Σ_0 using a restricted parameterization. In our case, the posterior distribution on the spanning tree concentrates rapidly (at a negative exponential rate in n) around the spanning tree summary of the true graph. In contrast, if we try to obtain the full graph, we necessarily concentrate critically slowly unless we make overly strong assumptions.

The spanning tree has been used previously in a variety of statistical contexts. Examples include approximating discrete distributions (Chow and Liu, 1968), hypothesis testing (Friedman and Rafsky, 1979), classification (Juszczak et al., 2009) and network analysis (Tewarie et al., 2015). It has also been considered as a graphical model (Meilă and Jaakkola, 2006; Edwards et al., 2010; Byrne and Dawid, 2015) with various extensions such as mixtures of trees (Meilă and Jordan, 2000) and algorithms for tree selection [see Chapter 7 of Højsgaard et al. (2012)]. Chow and Wagner (1973) showed consistency and Tan et al. (2011) quantified the convergence rate when the data are generated from a spanning tree graphical model. Various spanning tree posterior sampling algorithms using Metropolis-Hastings have been proposed (Dai, 2008; Green and Thomas, 2013) along with closed-form solutions for some marginal quantities (Schwaller et al., 2019).

We are inspired by this literature; however, our unique contributions include: (i) we characterize the posterior concentration rate (competitive to the graphical lasso) under a general case when the data may not be generated according to a spanning tree — this is distinct from prior literature where one assumes an oracle tree structure (Tan et al., 2011); (ii) we propose a novel Bayesian strategy of tree selection by harnessing global-local shrinkage priors (Polson and Scott, 2010) to induce a posterior concentrated near the minimum spanning tree; (iii) we propose to use rank constraint of the incidence matrix to

guarantee the connectivity of a spanning tree, which leads to convenient algorithms such as one for finding cut partition.

2. Bayesian Spanning Tree

To provide some intuition about the spanning tree as the backbone of a graph, we first briefly review the solutions to the traveling salesperson problem, demonstrate the simplicity of the spanning tree and motivating its use in a generative model.

2.1 Traveling on a Graph via the Spanning Tree

Suppose we have a graph $G = (V, E_G)$ with nodes $V = \{1, \ldots, p\}$ and edges $E_G = \{e_1, \ldots, e_M\}$. Each edge is undirected $e_s = (j, k) \equiv (k, j)$, and associated with a weight $w_{j,k} = w_{k,j} \geq 0$.

Consider the traveling salesperson problem: suppose G is a connected graph — for any two nodes j and k, there is a path$(j, k) = \{(j, l_1), (l_1, l_2), \ldots, (l_{m-1}, l_m), (l_m, k)\}$ (a subset of E_G) that allows us to travel from j to k. With each node representing a city and $w_{j,k}$ the distance between two cities, a salesperson wants to go to every city, while trying to minimize the total distance traveled. This is a combinatorial optimization problem:

$$\min_{\mathcal{I}} Q(\mathcal{I}) = \min_{\mathcal{I}} \sum_{(j,k)\in\mathcal{I}} w_{j,k},$$

where $\mathcal{I}$ is the itinerary, as an ordered sequence of edges.

Finding the optimal itinerary is a challenging problem. Nevertheless, we can consider a simpler problem that gives a close-to-optimal solution: since those p cities can be connected via $p - 1$ edges (roads), what if we first find the best $p - 1$ edges with the shortest total distance, and then develop an itinerary on them?

It is not hard to see that we only need to travel at most twice over each of those $p - 1$ edges (shown in Figure 1); hence we have a relaxed problem:

$$\min_{\mathcal{I}} Q(\mathcal{I}) \leq 2 \min_{T\in\mathcal{T}} \sum_{(j,k)\in T} w_{j,k}$$

where T is known as the spanning tree:

$$T = (V, E_T) : E_T \subseteq E_G, |E_T| = p - 1, T \text{ is connected},$$

that is, T is the subgraph of G having the smallest number of edges, while still connecting all the nodes; and $\mathcal{T}$ is the collection of all the spanning trees of G.

Unlike the original problem, the relaxed one (also known as the "minimum spanning tree" problem) can be solved easily — this is an M-convex problem [the discrete equivalent of convex (Murota, 1998)]; hence simple greedy algorithms (Kruskal, 1956; Prim, 1957) converge to the global optimum.

To link spanning trees to a graphical model, imagine that we generate a new variable y_j each time we reach a new node, where each y_j depends on one of the existing y_k's. Then

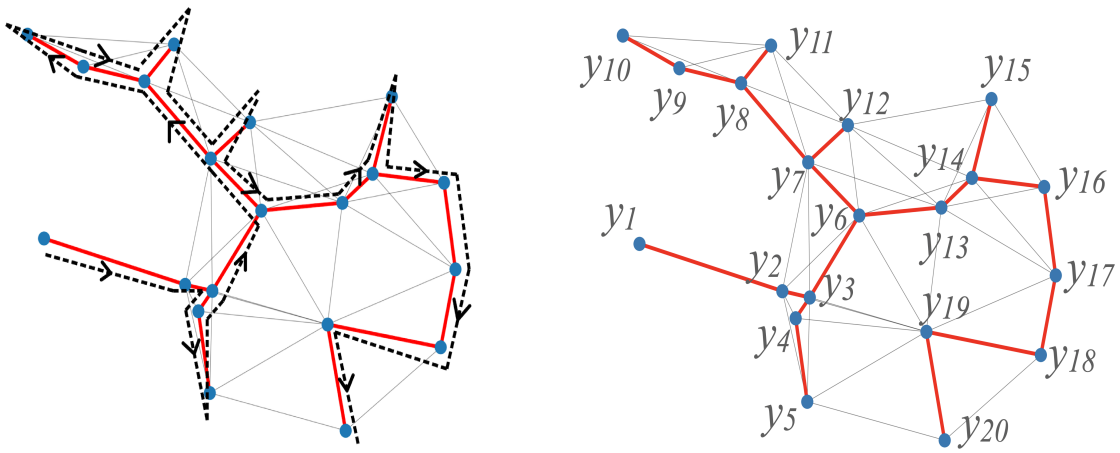
the graph would become a spanning tree with likelihood

$$\mathcal{L}(y;T) = \Pi(y_1) \prod_{j=2}^{p} \Pi\left[y_j \mid y_k : k \in \{1,\ldots,j-1\}, (j,k) \in E_T\right]. \tag{1}$$

This generative graph has two advantages: (i) the optimal point estimate T is tractable; (ii) it gives us a simplified graph of $(p-1)$ edges that estimate the "backbone" dependencies among all the variables — as we later show in the theory section, the population backbone graph can be formalized as:

$$T_0 = \arg\min_{T \in \mathcal{T}} \sum_{(j,k) \in T} \mathbb{E}\|y_j - y_k\|_2^2, \tag{2}$$

where the expectation is taken with respect to the generative distribution of the data.



(a) Traveling salesperson problem: reduce the total distance traveled to all cities [the nodes (blue dots), connected by edges (grey and red lines)]. The minimum spanning tree (red) gives a close-to-optimal solution.

(b) Generative graph conditioned on a spanning tree: a variable is generated when reaching a new node, as dependent on one of the existing variables. The tree is a random subgraph of the underlying graph.

Figure 1: Illustration of the minimum spanning tree and its graphical model.

2.2 Bayesian Spanning Tree Model

Slightly simplifying (1), we have the spanning tree graphical model:

$$\mathcal{L}(y;T,\theta) = \Pi(y_1) \prod_{e_s \in T} \Pi[y_k \mid y_j, e_s = (j,k), \theta_s], \tag{3}$$

where y_1 is the first variable generated that we refer to as the "root", we use $e_s \in T$ as shorthand for $e_s \in E_T$, and $\theta = \{\theta_s\}_{s=1}^{p-1}$ are the other parameters associated with the edges. We will refer to (3) as the spanning tree likelihood, and will complete its specification in the next subsection.

We can improve the model flexibility by taking a Bayesian approach and assigning a prior on T. This has the appealing advantages of (i) enabling regularization on the tree through the prior specification, (ii) allowing us to obtain a set of trees in the high posterior probability region, as opposed to a single estimated tree, and (iii) as will be shown in Section 3.2, we can analytically marginalize out T and obtain a marginal probability estimate of whether (j, k) are connected.

We specify the tree prior in the following form:

$$\Pi_0(T) = g(T), \qquad T \in \mathcal{T}. \tag{4}$$

and we use $\mathcal{T}$ to denote the set of all spanning trees with p nodes, and $g(T) \geq 0$ is a probability mass function that sums to one over $\mathcal{T}$; the set $\mathcal{T}$ is a combinatorial space, with a cardinalty $|\mathcal{T}| = p^{(p-2)}$ (Buekenhout and Parker, 1998). The posterior of T is a discrete distribution:

$$\Pi(T \mid y, \theta) = \frac{\mathcal{L}(y; T, \theta)\Pi_0(T)}{\sum_{T' \in \mathcal{T}} \mathcal{L}(y; T', \theta)\Pi_0(T')}. \tag{5}$$

The marginal density of the root $\Pi(y_1)$ is canceled in $\Pi(T \mid y, \theta)$, and hence we do not need to specify $\Pi(y_1)$ in the likelihood (the choice of y_1 as the root may impact $\Pi\left[y_j \mid y_k : (j, k), \theta_s\right]$, which will be addressed in the next subsection).

2.3 Location Dependence and Likelihood Specification

For ease of exposition, we assume each $y_k^{(i)} \in \mathbb{R}$ is continuous, and denote the n samples as $\vec{y}_k = [y_k^{(1)}, \ldots, y_k^{(n)}] \in \mathbb{R}^n$. Following a typical convention in graphical modeling (Wang, 2012; Tan et al., 2015), we assume each $\vec{y}_k$ is centered and standardized. To specify $\Pi\big[y_k \mid y_j, e_s = (j, k), \theta_s\big]$, we consider the commonly used location-scale family density:

$$\Pi\big[y_k \mid y_j, e_s = (j, k), \sigma_s\big] = \frac{1}{\sigma_s^n} f\left(\frac{\vec{y}_k - \vec{y}_j}{\sigma_s}\right), \tag{6}$$

where the conditional density of y_k is centered on y_j, $\sigma_s > 0$ is a scale parameter associated with the edge e_s, and $f : \mathbb{R}^p \to [0, \infty)$ integrates to one over $\mathbb{R}^p$.

In choosing a specific f, we focus on obtaining root exchangeability of this graphical model. In choosing a particular variable as the root node, we obtain a directed acyclic graph having a corresponding variable ordering. However, given that the choice of root node is typically arbitrary, it is desirable to remove dependence on its choice from the resulting posterior distribution of T.

Fortunately, this root exchangeability can be achieved as long as f satisfies the symmetry about zero constraint:

$$f(\vec{x}) = f(-\vec{x}) \qquad \forall \vec{x} \in \mathbb{R}^n. \tag{7}$$

Changing the root choice corresponds to applying a particular permutation of the variable/node index $\{\pi(1), \ldots, \pi(p)\}$. Considering the $e_s = (j, k)$ edge in the initial graph, after permutation we have $y_{\pi(j)} = y_j$, $y_{\pi(k)} = y_k$, and $\Pi\big[y_k \mid y_j, e_s = (j, k), \sigma_s\big]$ replaced by $\Pi\big[y_{\pi(j)} \mid y_{\pi(k)}, e_s = (\pi(k), \pi(j)), \sigma_s\big]$. Therefore, we have

$$\Pi[T_{(1,\ldots,n)} \mid (y_1, \ldots, y_p), \theta] = \Pi[T_{\pi(1),\ldots,\pi(p)} \mid y_{(\pi(1)}, \ldots, y_{\pi(p))}, \theta],$$

where $T_{\pi(1),\ldots,\pi(p)}$ denotes the permuted graph obtained by replacing k with $\pi(k)$ for each node, and (j,k) with $(\pi(j),\pi(k))$ for each edge. Hence, the structure of the tree does not change, but only the node number labels changes.

Remark 1 *The above exchangeability property is different from node exchangeability, where one would permute the variable/node index alone without changing the graph node index. For a more comprehensive discussion of graph exchangeability, see Cai et al. (2016a).*

To satisfy such symmetry constraints, and simplify our theoretical developments, we focus on a simple Gaussian density,

$$\Pi\big[y_k \mid y_j, e_s = (j,k), \sigma_s\big] = \frac{1}{(2\pi\sigma_s^2)^{n/2}|R|^{1/2}} \exp\left[-\frac{(\vec{y}_k - \vec{y}_j)^{\mathrm{T}} R^{-1} (\vec{y}_k - \vec{y}_j)}{2\sigma_s^2}\right], \tag{8}$$

where R is an $n \times n$ positive definite matrix allowing correlation between the samples $\vec{y}_k = \{y_k^{(1)}, y_k^{(2)}, \ldots, y_k^{(n)}\}$. If the multivariate data samples are uncorrelated, we simply let $R = I_n$. In more complex settings, such as when the samples have temporal dependence, R is a nuisance parameter, as our focus is on dependence across the outcomes and not the samples. The approach for handling R necessarily depends on the sampling design; for time series one may use an AR-1 structure, for spatially indexed data one may use a spatially decaying correlation function, etc. For simplicity, we focus on using $R = I_n$.

We refer to (8) as the Gaussian spanning tree likelihood.

Remark 2 *Although* (8) *may look similar to a regular multivariate Gaussian density, the key parameters in this likelihood are the choices of (j,k)'s in T, which determine which $(y_j - y_k)$'s enter into the likelihood. Further, note that when viewing σ_s^2 as a latent variable and equipping it with appropriately chosen distribution, we obtain a Gaussian scale-mixture spanning tree likelihood, which covers many useful non-Gaussian distributions (West, 1987).*

With σ_s^2 varying across edges, it is important to choose a distribution to regularize the spanning tree model. With this goal in mind, we build on the popular global-local shrinkage prior framework (Polson and Scott, 2010) in the next subsection.

2.4 Global-Local Shrinkage Prior for Tree Selection

Letting $\Pi_0(T)$ denote the prior probability of tree T, the posterior probability of choosing a particular spanning tree under the Gaussian spanning tree likelihood is

$$\Pr(T = T_\star \mid y, \theta) = \frac{\exp\left[-(1/2)\sum_{(j,k)\in T_\star}\left(\|\vec{y}_k - \vec{y}_j\|^2/\sigma_s^2\right)\right]\Pi_0(T_\star)}{\sum_{T'\in\mathcal{T}}\exp\left[-(1/2)\sum_{(j,k)\in T'}\left(\|\vec{y}_k - \vec{y}_j\|^2/\sigma_s^2\right)\right]\Pi_0(T')}. \tag{9}$$

Intuitively, if most of the σ_s's are small, then the posterior distribution will be dominated by trees T^* having most $\|\vec{y}_k - \vec{y}_j\|$'s small. Hence, a prior favoring small σ's will favor a smaller high probability region of spanning trees, leading to greater interpretability. Nevertheless, as shown in Figure 2, in order to form a valid spanning tree, we may have to choose a few long edges with large $\|\vec{y}_k - \vec{y}_j\|$; hence, the prior for σ_s should ideally be concentrated at small values with heavy tails.

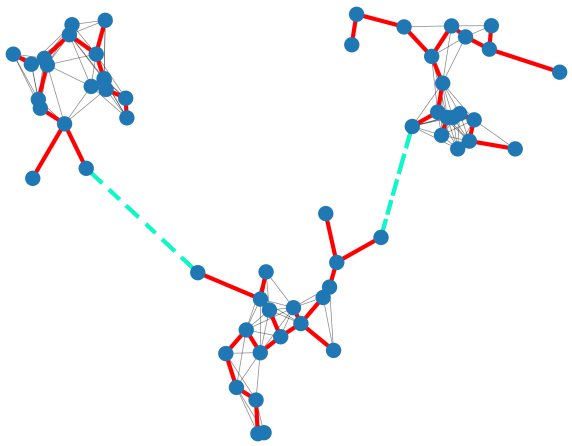


Figure 2: Illustration of the estimated spanning trees in the high posterior probability region. The minimum spanning tree is shown in red and cyan, and the alternative trees are shown in grey and cyan. Two edges with large $\|\vec{y}_k - \vec{y}_j\|$ (cyan) are necessary for forming a valid spanning tree.

To satisfy both properties, we use a global-local prior as:

$$\sigma_s = \lambda_s \tau, \qquad \lambda_s \overset{iid}{\sim} \Pi_\lambda, \qquad \tau \sim \Pi_\tau. \tag{10}$$

where $\tau > 0$ is the global scale with Π_τ concentrated near zero, so that $\tau \approx 0$ provides an overall strong shrinkage; whereas $\lambda_s > 0$ is the local scale from Π_λ, a heavy-tailed distribution so that a few λ_s can have very large values.

Although there are a broad variety of global-local priors that would suffice for our purposes, the generalized double Pareto (Armagan et al., 2013) is particularly convenient due to the closed-form marginal. We focus on the multivariate extension of Xu and Ghosh (2015), with $\lambda_s^2 \sim \int \mathrm{Ga}[\lambda_s^2; (n+1)/2, \kappa_s^2/2]\mathrm{Ga}(\kappa_s; \alpha, 1)\mathrm{d}\kappa_s$. Marginalizing over λ_s, we have (omitting a constant not involving α, τ, with complete details provided in the appendix):

$$\Pi\big[y_k \mid y_j, e_s = (j,k), \tau\big] \propto \frac{\Gamma(\alpha+n)}{\Gamma(\alpha)} \frac{1}{\tau^n} \left(1 + \frac{\|\vec{y}_j - \vec{y}_k\|_2}{\tau}\right)^{-(\alpha+n)}. \tag{11}$$

To favor a small global scale τ while being adaptive to the data, we use an informative exponential prior $\tau \sim \mathrm{Exp}(1/\mu_\tau)$ with the prior mean set to $\mu_\tau = \min_{(j,k:j\neq k)} \|\vec{y}_j - \vec{y}_k\|_2/n$, as an empirical estimate of the smallest scale among vectors $(\vec{y}_j - \vec{y}_k)$'s. We choose $\alpha = 5$ as a balanced choice between shrinkage and tail-robustness; details can found in Armagan et al. (2013). Alternatively, when there is no need to accommodate for a few edges with large $\|y_j - y_k\|_2$, one may use conjugate priors as suggested by Schwaller et al. (2019). If one needs to further reduce the graph estimate from a spanning tree, one could consider a continuous spike-and-slab distribution for σ_s^2 (George and McCulloch, 1995), which allows removal of some edges that are associated with the slab (large scale).

2.5 Prior for the Tree

We can obtain further regularization via the tree prior $\Pi_0(T)$. We discuss a few choices here, ranging from informative to non-informative.

Perhaps the simplest informative choice is an edge-based prior, including information that certain edges are more likely to be in the graph through a $p \times p$ matrix containing $\eta_{j,k} \geq 0$, and letting

$$\Pi_0(T \mid \eta) = z^{-1}(\eta) \prod_{(j,k)\in T} \eta_{j,k}, \tag{12}$$

where $z(\eta) = \sum_{T\in\mathcal{T}} \prod_{(j,k)\in T} \eta_{j,k}$ is the normalizing constant. Those (j,k) with larger $\eta_{j,k}$ will be more likely to be in T *a priori*. For example, in brain connection networks, one may favor connections between regions that are closer together spatially by letting $\eta_{j,k} = \exp(-\|x_j - x_k\|_2)$ with x_j the associated spatial coordinate. As another example, if one wants to block connections between j and k, one can simply set $\eta_{j,k} = 0$.

Often we do not know which edges are more likely, but may have prior preferences for certain graph statistics. Here we consider the degree, as the total number of edges for each node $D_j = \sum_{k=1}^{p} 1[(j,k) \in T]$. To obtain a degree-based prior, we propose to set $\eta_{j,k} = v_j v_k$, leading to

$$\Pi_0(T \mid v_1, \ldots, v_p) = z^{-1}(\eta) \prod_{j=1}^{p} v_j^{D_j}, \tag{13}$$

with $(v_1, \ldots, v_p)$ encoding prior knowledge of which nodes have more edges, and the normalizing constant having closed form $z(\eta) = (\sum_{j=1}^{p} v_j)^{p-2} \prod_{j=1}^{p} v_j$; a proof is provided in the appendix.

We now explore the case in which the v_js are assigned a Dirichlet hyper-prior

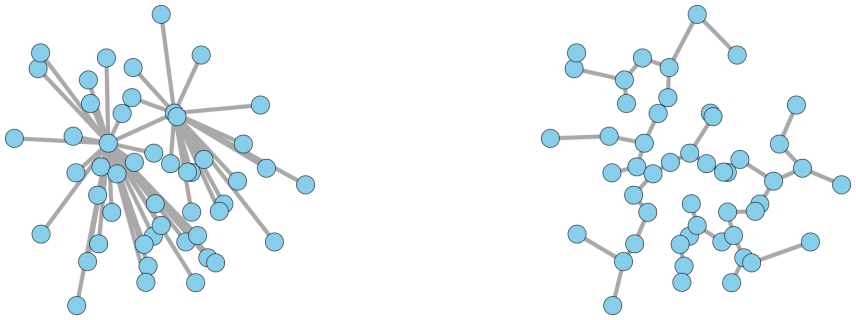
$$(v_1, \ldots, v_p) \sim \mathrm{Dir}(\alpha, \ldots, \alpha), \tag{14}$$

where α is the concentration parameter. Since $\sum_{j=1}^{p} v_j = 1$, we have $\Pi_0(T, v_1, \ldots, v_p \mid \alpha) \propto v_j^{(D_j+\alpha-2)}$. Conjugacy allows us to integrate out $v = (v_1, \ldots, v_p)$ and obtain the marginal prior on the degrees

$$\Pi_0(T \mid \alpha) \propto \prod_{j=1}^{p} \Gamma(D_j + \alpha - 1),$$

where $\sum_{j=1}^{p} D_j = 2(p-2)$. Due to the rapid increase of the gamma function $\Gamma(x) \approx \sqrt{2\pi x}(x/e)^x$, for small to moderate α (such as when $\alpha \ll p$), this prior is skewed towards having a few dominating large D_j's — as a result, the graph will contain a few "hubs", each connecting to a large number of nodes [Figure 3(a)]. Since the sum of D_j is fixed, α effectively controls the variance of degrees.

Alternatively, as a non-informative prior, one could use the uniform distribution over the space of $\mathcal{T}$, $\Pi_0(T) = 1/p^{(p-2)} 1(T \in \mathcal{T})$ [Figure 3(b)]. We use this as the default throughout the article.



(a) A hub graph generated from Dirichlet degree-based tree prior, with high degrees in only a few nodes.

(b) A graph generated from the uniform tree prior, with similar degrees across nodes.

Figure 3: Illustration of graphs generated from the Dirichlet degree-based tree prior and the non-informative uniform prior.

Regardless of the choice, all the above priors enjoy a simple form in the posterior distribution, as a product separable over the edges $\Pi(T \mid y) \propto \prod_{(j,k)\in T} \left\{\eta_{j,k}\Pi[y_k \mid y_j, (j,k), \theta_s]\right\}$. This separable form allows us to easily update each edge in posterior computation, while leading to useful closed-form quantities in the marginal posterior, as presented below.

3. Properties

3.1 Partition Function and Marginal Connecting Probability

The spanning tree is supported in a large combinatorial space. Despite the high complexity, some quantities related to the marginal posterior distribution are available analytically in closed-form. These results are related to findings in Schwaller et al. (2019).

For generality, we focus on the posterior distribution of T in the following form:

$$\text{pr}(T \mid y) = \frac{\prod_{(j,k)\in T} \exp(q_{j,k})}{z_q},$$

where $\exp(q_{j,k}) = \eta_{j,k}\Pi[y_k \mid y_j, (j,k), \theta_s]$ as shown above. The denominator z_q is commonly known as the partition function:

$$z_q = \sum_{T\in\mathcal{T}} \prod_{(j,k)\in T} \exp(q_{j,k}). \tag{15}$$

Letting $A_q = \{\exp(q_{j',k'})\}_{(j',k')}$, with $A_{q:j,j} = 0$ in the diagonal, we show in the following Theorem that z_q can be computed easily. All proofs are deferred to the appendix.

Theorem 3 (Kirchhoff's matrix tree theorem for partition function) *Let A_q be defined as above, $D_q = diag\{\sum_{k\neq j} \exp(q_{j,k})\}_j$, $L_q = D_q - A_q$ be the Laplacian matrix, and J be a matrix of ones. Then*

$$z_q = det(L_q + J/p^2).$$

In summarizing the posterior distribution of T, it is useful to examine the chance that a particular (j,k) is picked by the spanning tree. In particular, we focus on the marginal

posterior probability that edge (j,k) is in T, which corresponds to the sum of the posterior probabilities of all trees having edge (j,k). Remarkably these marginal posterior edge probabilities are available in closed form. Differentiating the log-partition function, we have

$$\begin{aligned}\mathrm{pr}[T \ni (j,k) \mid y] &= \sum_{T\in\mathcal{T}} \mathbf{1}[(j,k)\in T]\mathrm{pr}(T \mid y) \\ &= \frac{\sum_{\text{all } T\ni(j,k)} \exp(q_{j,k}) \prod_{(h,l)\in T:(h,l)\neq j,k} \exp(q_{h,l})}{z_q} \\ &= \frac{\partial \log z_q}{\partial q_{j,k}} \\ &= (\Omega^L_{j,j} + \Omega^L_{k,k} - 2\Omega^L_{j,k}) \exp(q_{j,k}),\end{aligned}$$

where $\Omega^L = (L_q + J/p^2)^{-1}$. We refer $\mathrm{pr}[T \ni (j,k) \mid y]$ as the "marginal connecting probability" for (j,k).

The closed-form for the marginal connecting probability is dependent on other parameters, such as the scale τ and hyper-parameters $\{v_j\}$. Hence, it is still necessary to perform computation for the joint posterior distribution of those parameters and tree.

3.2 Connectivity Guarantee via Matrix Rank Constraint

The parameter T needs to satisfy the connected graph constraint, which may appear challenging to enforce computationally – if we want to propose an update to T during a sampling algorithm, how can we ensure that the proposal is still a spanning tree? A naïve way would be checking consecutively over the edges in E_T to ensure the connectivity — this procedure is commonly known as "graph traversal".

We propose an alternative to bypass the need for graph traversal. Consider the incidence matrix B, a $p\times(p-1)$ matrix that records the *node-to-edge* relationship. For $s = 1,\ldots,p-1$, if $e_s = (j,k)$, we set $B_{j,s} = 1$, $B_{k,s} = -1$ and all other $B_{l,s} = 0$ for $l \neq j$ or k. The matrix B is useful, because a graph of p nodes is connected if and only if the rank of its incidence matrix $\mathrm{rank}(B) = p-1$ [Theorem 2.3 of Bapat (2010)]; that is, B is of full column rank. Therefore, we can convert the combinatorially constrained space into a simple rank-constrained problem:

$$\mathcal{T} = \{T : B \text{ is the incidence matrix of } T,\ \mathrm{rank}(B) = p-1\}.$$

We show the full-rankness of B implies an appealing combinatorial property, which allows us to quickly find the "graph cut partition" related to each edge. To formalize, in a spanning tree, removing an edge $e_s = (j,k)$ will create two disconnected components; we want to find the graph cut partition as the two disjoint sets of vertices:

$$\begin{aligned}&\mathrm{Cut}[(j,k)] = (V_1, V_2) \\ &\text{such that } j\in V_1, k\in V_2, \\ &\qquad V_1\cup V_2 = V,\ \ V_1\cap V_2 = \varnothing \\ &\qquad G(V_1) \text{ connected}, G(V_2) \text{ connected},\end{aligned}$$

where $G(V_k)$ is the sub-graph of G containing only the nodes in V_k. Finding $\mathrm{Cut}[(j,k)]$ is non-trivial – in a brute-force approach, one starts from node j, traversing the edges

$E_T \setminus \{(j,k)\}$ and adding all the visited nodes to V_1; after visiting all the nodes accessible in the path from j, one assigns the remaining nodes to V_2.

The rank constraint can significantly reduce the burden — due to the full column rank, the projection of any column $\vec{B}_s$ (corresponding to an edge e_s) into the nullspace of the others would be a non-zero vector. Interestingly, the output of this projection contains only two unique values, allowing us to directly find V_1 and V_2.

Theorem 4 (Traversal-free solution to find the graph cut partition) *Denote the sth column of B by $\vec{B}_s$, and other columns by $B_{[-s]}$. Then the p-element vector*

$$\vec{\beta}_s = \{I - B_{[-s]}(B_{[-s]}^{\mathrm{T}} B_{[-s]})^{-1} B_{[-s]}^{\mathrm{T}}\}\vec{B}_s,$$

contains only two unique values $\beta_{1,s}^*, \beta_{2,s}^* \in \mathbb{R}$ with $\beta_{1,s}^* \neq \beta_{2,s}^*$. The $Cut(e_s) = (V_1, V_2)$ can be found using $V_1 = \{j : \beta_{j,s} = \beta_{1,s}^*\}$ and $V_2 = \{j : \beta_{j,s} = \beta_{2,s}^*\}$.

Remark 5 *The matrix inversion $(B_{[-s]}^{\mathrm{T}} B_{[-s]})^{-1}$ can be computationally costly for large p with a complexity at $O(p^3)$. To address this issue, we develop an efficient algorithm that can extract $(B_{[-s]}^{\mathrm{T}} B_{[-s]})^{-1}$ from $(B^{\mathrm{T}}B)^{-1}$, and update the value of $(B^{\mathrm{T}}B)^{-1}$ when a column of B changes. This allows us to only evaluate $(B^{\mathrm{T}}B)^{-1}$ for one time during the algorithm initialization, and reduce the cost of computing $\vec{\beta}_s$ to $O(p)$ during the posterior sampling. The details are provided in the appendix.*

4. Posterior Computation

We develop an efficient Gibbs sampler for sampling from the posterior distribution over spanning trees. To facilitate fast exploration of the high posterior probability region, we develop: (i) a graph update step that can rapidly change the shape of the spanning tree; (ii) an initialization that gives the approximate posterior mode of the spanning tree.

4.1 Gibbs sampling with the Cut-and-Reconnect Step

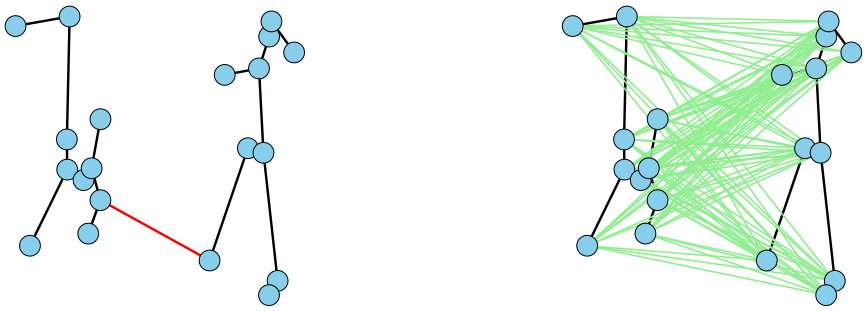
4.1.1 Update T

Since the full conditional distribution of T takes a product form over (j,k), one may use the random-walk cover algorithm (Aldous, 1990; Mosbah and Saheb, 1999), loop-erasing algorithm (Wilson, 1996) or approximation (Durfee et al., 2017) to directly sample T. This would give independent samples of spanning trees.

On the other hand, to allow potential extension for sampling T from a non-product form distribution, a Gibbs sampler would be of interest as well. At the current state of T with $E_T = \{e_1, \ldots, e_{p-1}\}$, to make an update to the spanning tree, we propose a "cut-and-reconnect" step for each edge $e_s = (j,k)$: we first remove this edge and find its graph cut partition $\mathrm{Cut}(e_s) = (V_1, V_2)$, and then sample a new edge (j', k') across V_1 and V_2, so that we obtain a new spanning tree T^*.

The transition that replaces (j,k) with (j',k') is reversible and has a simple multinomial probability:

$$\mathrm{pr}[(j',k') \in T^* \mid E_T \setminus (j,k)] = \frac{\exp(q_{j',k'})}{\sum_{j\in V_1, k\in V_2} \exp(q_{j,k})}.$$



(a) Cutting an edge (red) gives two disconnected subgraphs (with node sets V_1 and V_2), calculated using the Theorem 4.

(b) Drawing a new edge from the multinomial distribution with $|V_1| \times |V_2|$ choices (green) and form a new tree.

Figure 4: At each step, the cut-and-reconnect step explores a change of edge over a large number of candidates, leading to rapid state exploration in the spanning tree space.

Repeating this for $s = 1, \ldots, p-1$ rapidly changes the shape of the tree.

To understand why this edge-based update can efficiently explore the space of $\mathcal{T}$, we compare it with an alternative node-based update — removing a node j from the graph and reattaching it to one of the other nodes. For the node-based update, there are only at most $(p-1)$ candidates in the multinomial draw; whereas for the edge-based update, there are $|V_1|(p-|V_1|)$ candidates, which has an order up to $O(p^2)$. To illustrate the large number of candidates in the edge-based update, we plot a diagram in Figure 4.

4.1.2 Update the other parameters

Using the marginal density in (11), the parameters can be updated via

- Sample $\tau = |\tilde{\tau}|$ using random-walk Metropolis, by proposing $\tilde{\tau}^* \sim \text{Uniform}(\tilde{\tau}-\delta, \tilde{\tau}+\delta)$ with $\delta > 0$ a tuning parameter, and accepting with probability:

$$\min\left\{1, \frac{\prod_{(j,k)\in T}\left[\frac{1}{|\tilde{\tau}^*|^n}\left(1+\frac{\|\vec{y}_j-\vec{y}_k\|_2}{|\tilde{\tau}^*|}\right)^{-(\alpha+n)}\right]\exp(-|\tilde{\tau}^*|/\mu_\tau)}{\prod_{(j,k)\in T}\left[\frac{1}{|\tilde{\tau}|^n}\left(1+\frac{\|\vec{y}_j-\vec{y}_k\|_2}{|\tilde{\tau}|}\right)^{-(\alpha+n)}\right]\exp(-|\tilde{\tau}|/\mu_\tau)}\right\}.$$

- If the degree-based prior in (13) is used, update $(v_1, \ldots, v_p) \sim \text{Dir}(D_1+\alpha-1, \ldots, D_p+\alpha-1)$.

For the first step, we use an adaptation period at the beginning of the MCMC algorithm to tune $\delta > 0$, so that the acceptance rate of this step is around 0.3.

In the above algorithm, updating one edge of the tree has a computational complexity of $O(p)$; hence sampling all edges has a complexity of $O(p^2)$. To further accelerate the

algorithm, one could use the random scan Gibbs sampler (Levine and Casella, 2006), which in each iteration randomly chooses and updates a subset of the edges, hence reducing the complexity to $O(p)$.

In terms of the computational time, for a graph containing $p = 200$ nodes, sampling a $1,000$ steps takes about 2 minutes using Python on a quad-core laptop. The Markov chain mixes rapidly, and we show diagnostics in the appendix.

Remark 6 *As an alternative to sampling, if the marginal connecting probability is the main interest, one can obtain a fast estimate of* $pr[T \ni (j,k) \mid y, \hat{\tau}]$ *using a reasonable point estimate of* τ*. For example, one could first find the conditional posterior mode of the spanning tree* $\hat{T}$ *(presented in the next subsection), then use the maximizer in the density* (11) *given the tree:* $\hat{\tau} = \alpha \sum_{(j,k)\in\hat{T}} \|y_j - y_k\|_2/[n(p-1)]$*. Computing the marginal connecting probability (as well as the other quantities such as the partition function) is almost instantaneous and takes at most a few seconds for* $p = 10^4$*.*

4.2 Conditional Posterior Mode Estimation

Our next goal is to find a good initialization for the spanning tree; for this purpose, we will use the posterior mode of the spanning tree given some initial estimate of (τ, η). In this article, we initialize τ at μ_τ, and all $\eta_{j,k} = 1/p^2$. Denote the adjacency matrix of the tree by $A = \{a_{j,k}\}$, $a_{j,k} = 1$ if $(j,k) \in T$, $a_{j,k} = 0$ otherwise; and the space of all adjacency matrices for spanning trees by $\mathcal{A}_T$. Then the posterior mode of the adjacency matrix is

$$\arg\max_{A\in\mathcal{A}_T} \sum_{j>k} a_{j,k} q_{j,k}, \qquad \text{with } q_{j,k} = -(\alpha+n)\log\left(1 + \frac{\|\vec{y}_j - \vec{y}_k\|_2}{\tau}\right) + \log\eta_{j,k}. \tag{16}$$

Therefore, the mode is equivalent to the minimum spanning tree in a complete and weighted graph, with the edge weight $(-q_{j,k})$. There are several algorithms that can quickly find the globally optimal solution (Prim, 1957; Dijkstra, 1959; Kruskal, 1956; Karger et al., 1995). We choose to present Prim's algorithm due to its simplicity.

Algorithm 1: Finding the conditional posterior mode $\hat{A}$

Initialize $U = \{1\}$, $E_T = \varnothing$ and $\bar{U} = \{1, \ldots, p\} \setminus U$;
while $|U| < p$ **do**
  Find $(j,k) = \arg\max_{(l,l')\in(U,\bar{U})} q_{l,l'}$, and (j,k) into E_T.
  Move l' from $\bar{U}$ to U.
end

To explain this algorithm, we initialize the tree as a single node $\{1\}$, then add one edge at a time; each time, among the edges that connect the current tree and the remaining nodes, we pick the one with the largest $q_{l,l'}$. This is a greedy algorithm that finds the locally optimal solution at each step; nevertheless, since the minimum spanning tree problem is M-convex [a discrete extension of continuous convexity], the greedy algorithm is guaranteed to converge to the global optimum [see Chapter 6.7 of Murota (1998)].

On the Prim's algorithm above, one thing of independent interest is that at each step, we only need to find the *index* of largest $q_{l,l'}$ in $(U, \bar{U})$, hence only the ordering of $q_{l,l'}$'s

impacts the tree estimate. This immediately leads to an invariance property of maximum likelihood estimator for the tree.

Theorem 7 (Invariance of maximum likelihood tree estimator) *For the maximum likelihood tree estimator,*

$$\hat{A}^* = \arg\max_{A \in \mathcal{A}_T} \sum_{j>k} a_{j,k} \log \Pi\big[y_k \mid y_j, e_s = (j,k), \theta\big],$$

if $\Pi\big[y_k \mid y_j, e_s = (j,k), \theta\big]$ *is monotonically decreasing in the divergence/distance between* y_j *and* y_k*:* $div(y_j, y_k) < div(y_h, y_l) \Rightarrow \Pi\big[y_k \mid y_j, e_s = (j,k), \theta\big] > \Pi\big[y_h \mid y_l, e_s = (h,l), \theta\big]$*, then*

$$\hat{A}^* = \arg\min_{A \in \mathcal{A}_T} \sum_{j>k} a_{j,k} div(y_j, y_k).$$

As one application, for any spanning tree model specified with $\Pi\big[y_k \mid y_j, e_s = (j,k), \theta\big]$ decreasing in $\|y_j - y_k\|$ (such as the generalized double Pareto density, location-scale t-distribution density for continuous $y_j \in \mathbb{R}^n$, or hamming distance-based probability for binary $y_j \in \{0,1\}^n$), $\hat{A}^*$ would be the same as the one under a Gaussian/Gaussian-form $\Pi\big[y_k \mid y_j, e_s = (j,k), \theta\big] \propto \exp(-\|y_j - y_k\|_2^2/2)$. Further, in our case (16), since the prior probability gets overwhelmingly dominated by the likelihood as $n \to \infty$, we see that our posterior mode converge to the maximum likelihood estimator under a Gaussian density as well. Therefore, we will now discuss mostly on Gaussian spanning tree models in the theoretic section, and the results can be generalized to non-Gaussian models.

5. Theoretic Study

In this section, we provide a more theoretical exposition on our spanning tree model.

5.1 Connection to Gaussian Graphical Models

We now focus on the Gaussian spanning tree model, and compare it with Gaussian graphical models. Following Ravikumar et al. (2011), we assume $y^{(i)} = (y_1^{(i)}, \ldots, y_p^{(i)})$ is a zero-mean random vector with covariance Σ_0. We denote the empirical covariance by $S_n = \sum_{i=1}^n y^{(i)} y^{(i)\mathrm{T}}/n$.

The incidence matrix B can be used as a contrast matrix for computing $\vec{y}_j - \vec{y}_k$. Therefore, we have a matrix representation for the posterior:

$$\begin{aligned} \mathcal{L}(y; T, \theta)\Pi_0(T) &\propto |\Psi|^{-n/2} \exp\left\{ -\frac{1}{2} \sum_{i=1}^n \mathrm{tr}(y^{(i)\mathrm{T}} B \Psi^{-1} B^{\mathrm{T}} y^{(i)}) + \mathrm{tr}(A \log \eta) \right\} \\ &= |\Psi|^{-n/2} \exp\left\{ -\frac{n}{2} \mathrm{tr}(B \Psi^{-1} B^{\mathrm{T}} S_n) + \mathrm{tr}(A \log \eta) \right\}, \end{aligned} \tag{17}$$

where $\Psi = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_{p-1}^2)$, $\log \eta$ is calculated element-wise on $\eta_{j,k}$, and A is the adjacency matrix for T. For a tractable theoretic analysis, we will treat η as fixed with all $|\log \eta_{j,k}|$ finite.

We can immediately see some similarity of (17) to regularized Gaussian graphical models, such as the one associated with the graphical lasso (Friedman et al., 2008), with a regularized likelihood proportional to $|\hat{\Omega}|^{-1/2}\exp\{-n/2\mathrm{tr}(\hat{\Omega}S_n) - \lambda|\hat{\Omega}|_{1,1}\}$, and $\hat{\Omega}$ the precision matrix assumed to be sparse, as induced by the $(1,1)$-matrix norm.

Indeed, we will show that (17) is equivalent to a regularized Gaussian graphical model, except the structure of the precision matrix is determined by a spanning tree. Consider a weighted spanning tree, with weighted adjacency $(A_\phi)_{j,k} = \sigma_s^{-2}$ for the edge $e_s = (j,k)$ and $(A_\phi)_{j,k} = 0$ if (j,k) is not an edge; and denote its degree matrix by D_ϕ.

Lemma 8 *The Laplacian matrix $L_\phi = D_\phi - A_\phi$ has $L_\phi = B\Psi^{-1}B^{\mathrm{T}}$.*

Using the spectral property of the Laplacian, the smallest eigenvalue $\lambda_{(1)}(L_\phi) = 0$ with its eigenvector $\vec{1}/\sqrt{p}$; and the number of zero eigenvalues is equal to the number of isolated subgraph(s) — since the spanning tree is connected, there is only one subgraph hence only one eigenvalue equal to zero. Therefore, it is not hard to see the matrix

$$\tilde{\Omega} = L_\phi + \epsilon J$$

is strictly positive definite with $\epsilon > 0$, which can be viewed as a precision matrix. Further, we have the following identity.

Lemma 9 *With $\tilde{\Omega} = L_\phi + \epsilon J$, we have $|\hat{\Omega}| = p^2\epsilon|\Psi^{-1}|$.*

Therefore, (17) becomes (omitting constant):

$$\mathcal{L}(y;T,\theta)\Pi_0(T) \propto |\tilde{\Omega}|^{-n/2}\exp\left\{-\frac{n}{2}\mathrm{tr}(S_n\tilde{\Omega})\right\}\exp\{\mathrm{tr}(A_T\log\eta)\}, \tag{18}$$

which is a restricted Gaussian graphical model with the precision matrix parameterized by the spanning tree.

5.2 Convergence of the Tree

With the connection to Gaussian graphical models established, a natural question is what is the advantage of the spanning tree-based model, compared to other less restricted models?

Other than immense computational advantages, the key advantage is that we do not require any assumption on the population Σ_0 to accurately recover the minimum spanning tree based on Σ_0. In comparison, most of the existing approaches require specific strong assumptions on Σ_0 such as sparsity or norm constraints, unless $n \gg p$.

For the ease of analysis, we consider the non-informative prior $\Pi_0(T) \propto 1$. Integrating out σ_s^2 over the generalized double Pareto prior and examining the Prim's algorithm, we see the following equivalence:

$$\underset{(j,k)\in(U,\bar{U})}{\arg\min}\ (\alpha+n)\log\left(1+\frac{\|\vec{y}_j-\vec{y}_k\|_2}{\tau}\right) = \underset{(j,k)\in(U,\bar{U})}{\arg\min}\ \|\vec{y}_j-\vec{y}_k\|_2^2,$$

due to the monotonicity of the function $(\alpha+n)\log(1+x/\tau)$ in $x>0$, for any $\tau>0$. As we can verify that $\|\vec{y}_j-\vec{y}_k\|_2^2/n = S_{n:j,j} + S_{n:k,k} - 2S_{n:j,k}$, with $S_{n:j,k}$ the (j,k)th element of the empirical covariance S_n, we have the posterior mode of the tree as

$$\hat{T} = \arg\min_{T\in\mathcal{T}}\sum_{(j,k)\in T} W_{n:j,k}, \qquad W_{n:j,k} = S_{n:j,j} + S_{n:k,k} - 2S_{n:j,k}. \tag{19}$$

It is easy to see that as $n \to \infty$, $S_n \to \Sigma_0$ in probability and the posterior mode will converge to

$$T_0 = \arg\min_{T \in \mathcal{T}} \sum_{(j,k) \in T} W_{0:j,k}, \quad W_{0:j,k} = \Sigma_{0:j,j} + \Sigma_{0:k,k} - 2\Sigma_{0:j,k} = \mathbb{E}\|y_j - y_k\|_2^2. \tag{20}$$

That is, asymptotically, our model recovers the minimum spanning tree of W_0, providing accurate partial information about the population covariance Σ_0 .

The next crucial question is, how fast does (19) converge to (20)? At a finite n, to successfully recover T_0 at $\hat{T}$, we only need the *ordering* in $\{W_{n:j,k}\}_{(j,k)}$ to partly match the one in $\{W_{0:j,k}\}_{(j,k)}$. Intuitively, this condition is much easier to meet, compared to having $\|\hat{\Omega} - \Sigma_0^{-1}\| \approx 0$ as in other approaches using a full covariance/precision matrix estimation.

We now formalize this intuition. For the required ordering condition, we first state an important property of the minimum spanning tree. For generality, we consider the case when the minimum spanning tree may be not unique (that is, there could be multiple equivalent solutions in (20)).

Theorem 10 (Path strict optimality) *For a complete graph with edge weights $\{W_{j,k}\}_{j,k}$, denote $\{T_0^{(1)}, T_0^{(2)}, \ldots, T_0^{(M)}\}$ as the set of all the minimum spanning trees. Any edge outside the trees $(h,l) \notin \cup_{m=1}^M T_0^{(m)}$ has higher weight than every edge on the tree path $(j,k) \in T_0^{(m)} : (j,k) \in path(h,l)$ for $m = 1, \ldots, M$; that is, we have $W_{h,l} > W_{j,k}$ strictly.*

Using the above theorem, we can define a "separability constant" $\delta > 0$ to quantify how separable the minimum spanning tree(s) is from the other spanning trees:

$$\delta = \min_{(h,l,j,k) \in \mathcal{I}} (W_{h,l} - W_{j,k}),$$
$$\text{where } \mathcal{I} = \Big\{ (h,l,j,k) : (h,l) \notin \cup_{m=1}^M T_0^{(m)}, (j,k) \in T_0^{(m)} : (j,k) \in \text{path}(h,l) \text{ for } m = 1, \ldots, M \Big\}.$$

We use Figure 5(a) to explain the above separability constant — let (h^*, l^*, j^*, k^*) be the indices so that we reached $W_{h^*,l^*} - W_{j^*,k^*} = \delta$, with (j^*, k^*) on one of the minimum spanning trees $T_0^{(1)}$ and (h^*, l^*) not on any of the $T_0^{(m)}$. If we remove the edge (j^*, k^*), the path(h^*, l^*) will be disrupted; hence the graph cut will lead to $h^* \in V_1$ and $l^* \in V_2$. As a result, adding (h^*, l^*) will form a new spanning tree. This new tree is sub-optimal in terms of having the total weights δ larger than $T_0^{(1)}$.

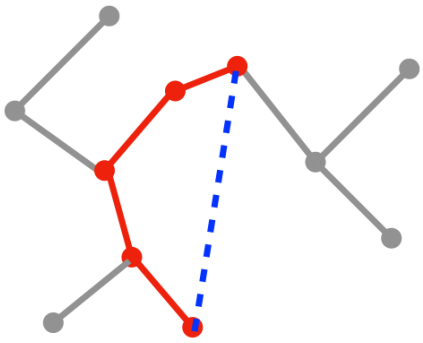
We are now ready to state the convergence rate for the posterior mode.

Theorem 11 *Assume $y^{(i)} \overset{iid}{\sim} \mathcal{F}$, $i = 1, \ldots, n$, with $\mathcal{F}$ a p-variate distribution with mean $\vec{0}$ and covariance matrix Σ_0, and each $y_j^{(i)}$ sub-Gaussian with bound parameter λ. Denote the set of all minimum spanning trees based on Σ_0 as in (20) by $\mathcal{T}_0 = \{T_0^{(1)}, T_0^{(2)}, \ldots, T_0^{(M)}\}$. Denoting a posterior mode of the spanning tree by $\hat{T}$,*

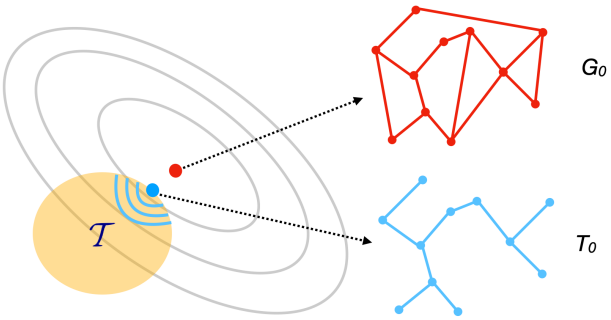
$$pr(\hat{T} \notin \mathcal{T}_0) \leq \frac{2}{3} \exp\Big\{ -\frac{n\delta^2}{8(\beta_0^2 + \beta_0\delta)} + 3\log p \Big\},$$

where $\beta_0 = 2(11\lambda^2)^3/(v^2)$ *and* $v^2 = \min_{j,k}[\mathbb{E}(\vec{y}_j - \vec{y}_k)^4 - W^2_{0:j,k}]$.

We do not impose a Gaussian assumption on $\mathcal{F}$, but just the tail concentration condition on y. The probability of falling outside of $\mathcal{T}_0$ drops to zero rapidly, at an exponentially decaying rate in the separability constant δ and sample size n. Therefore, we only need $n \gg \log p$ without requiring any conditions on Σ_0 or its associated graph G_0. Figure 5(b) illustrates the intuition. Note that above result holds for any $\mathcal{F}$ with finite support (since they are sub-Gaussian), such as one for binary $y_j \in \{0,1\}^n$.



(a) Path optimality of the minimum spanning tree (shown in solid lines): any edge not on the minimum spanning tree (blue) has strictly higher weight than every edge in the tree path (red) connecting the two nodes.



(b) Intuition about the faster convergence of the restricted graph model: the oracle graph G_0 sits in the unrestricted graph space. Due to the high dimension $p \gg n$, the covariance/precision matrix-based estimator has a large uncertainty (grey contours) and critically slow convergence rate. The spanning tree model is in a restricted parameter space $\mathcal{T}$ (orange), where the tree estimator has a much smaller uncertainty (blue contours) and faster convergence to T_0 — the minimum spanning tree transform of G_0.

Figure 5: Illustration of the theory results.

Remark 12 *To provide a probabilistic interpretation of the above theorem, note that* (18) *is a discrete distribution quickly converging to the posterior mode set as* $n \to \infty$. *Therefore, the theorem shows that the large sample posterior is correctly concentrated toward* $\mathcal{T}_0$, *as a*

transform of the ground-truth G_0. On the other hand, it remains an open theoretic question on how to quantify the differences between $\Pi(T \mid y)$ and $\Pi(G \mid y)$.

6. Numerical Experiments

6.1 Comparing Point Estimates with Existing Approaches on Backbone Estimation

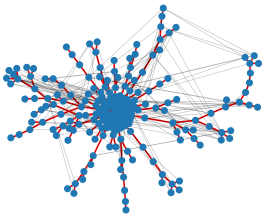
We compare our Bayesian spanning tree model against a few popular graph estimation approaches: the thresholding estimator on the absolute empirical correlation; the graphical lasso with a chosen α, as the multiplier to the $l_{1,1}$-norm of the precision matrix; the graphical lasso with α chosen by cross-validation (as implemented in the scikit-learn package). We record the graph edge estimate (j, k) as where $\hat{\Sigma}_{j,k} \neq 0$ in the thresholding estimator, and $\hat{\Omega}_{j,k} \neq 0$ in the graphical lasso.

First, we consider the common assumption that the graph is very sparse. We use the scikit-learn package to generate a precision matrix Ω_0, with edge density level at 3% (number of non-zero elements divided by p^2), and non-zero correlations of magnitude between $(0.3, 0.9)$. At $p = 200$, this leads to approximately 600 edges in each experiment. Then we simulate repeats of the data $(y_1^{(i)}, \ldots, y_p^{(i)}) \sim \mathrm{N}(0, \Omega_0^{-1})$ for $i = 1, \ldots, n$ and obtain graph estimates $\hat{G}$ from each method. Each setting is repeated over different n's, and for each n the mean of 10 experiments is shown with the 95% confidence interval for each reported quantity.

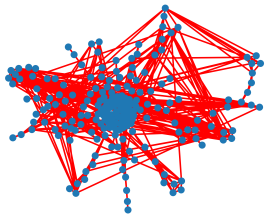
Denote the oracle graph by G_0 and its minimum spanning tree by T_0. Ideally, we want the graph estimate to fully cover the backbone subgraph $\hat{G} \supseteq T_0$, while having $\hat{G} \subseteq G_0$ so that we do not obtain too many falsely positive edge estimates. Therefore, a useful benchmark for the estimation error is $|T_0 \setminus \hat{G}| + |\hat{G} \setminus G_0|$; we show the details in the appendix.

Remark 13 *To be fair, due to the constraints of spanning trees, there will be false negative estimates in $G_0 \setminus \hat{G}$ if G_0 has more than $(p-1)$ edges. Nevertheless, as our focus is on recovering the backbone graph (to be exact, the backbone support graph of the oracle covariance matrix as in (20), regardless if G_0 is disconnected), we choose to benchmark using the number of false positive edges as estimation error.*

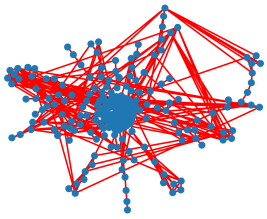
As shown in Figure 6, the graphical lasso using cross-validation produces the largest number of edges; while its estimates cover T_0, they also contain too many falsely positive $|\hat{G} \setminus G_0|$. Empirically tuning $\alpha = 0.8$ in the graphical lasso somewhat reduces this problem. The correlation thresholding estimator using $|\rho|_{j,k} \geq 0.5$ (as a common "default" choice in practice) seems to produce the best results among the existing approaches. The Bayesian spanning tree shows a competitive performance to the best existing approach. At the same time, it shows a low error for estimating T_0: at $n \geq 100$, it almost perfectly recovers all the edges in T_0.



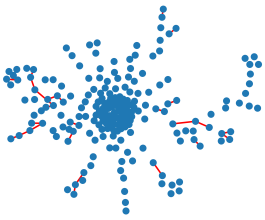
(a) Oracle graph, with its minimum spanning tree shown in red.



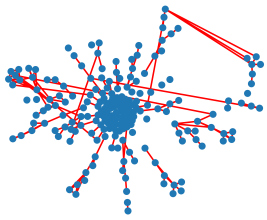
(b) Graphical lasso with α chosen by cross-validation.



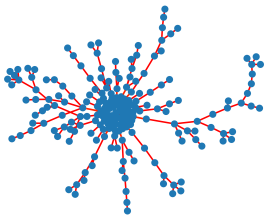
(c) Graphical lasso using $\alpha = 0.8$.



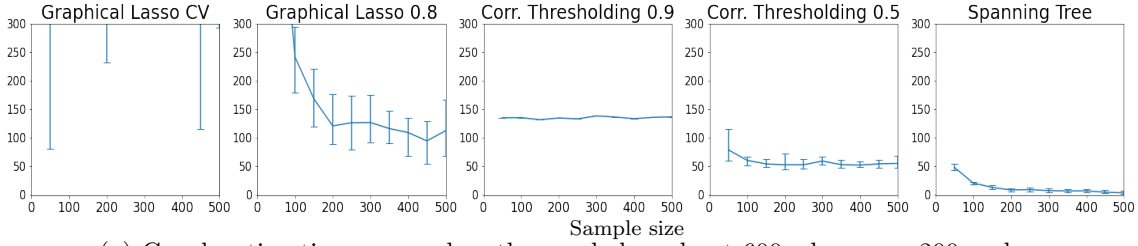
(d) Thresholding absolute correlation at 0.9.



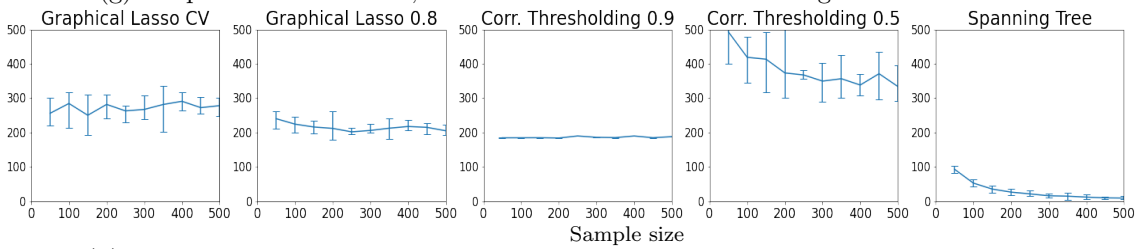
(e) Thresholding absolute correlation at 0.5.



(f) Bayesian spanning tree (posterior mode shown).



(g) Graph estimation error, when the oracle has about 600 edges over 200 nodes.



(h) Graph estimation error, when the oracle has about $4,000$ edges over 200 nodes.

Figure 6: Simulated experiments on graph estimation. Panels (b-f) are point estimates obtained at $n = 100$ for the oracle graph with about 600 edges.

Next, we slightly change the experiment setting by increasing the denseness of the oracle graph. We adjust the edge density level to around 20%, which leads to a graph with about $4,000$ edges over 200 nodes. This time, all the existing approaches show much larger estimation error, likely due to the breakdown of the sparsity assumption. On the other hand, the Bayesian spanning tree still maintains a good performance, with the estimation error rapidly decreasing in n.

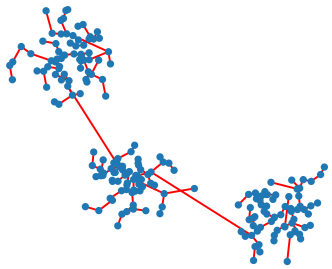
We provide the detail in the appendix, and illustrate the sensitivity in solely relying on comparing the magnitude of empirical precision matrix elements for graph estimation.

6.2 Uncertainty Quantification of the Graph Estimates

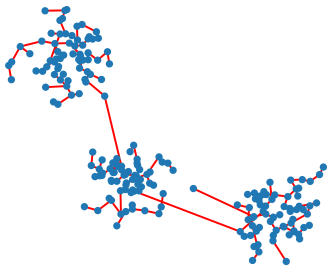
We now demonstrate the capability of the uncertainty quantification of our Bayesian spanning tree model.

First, we consider the common example of a latent position graph (Hoff et al., 2002) associated with three communities. In the latent space, each community is a group of points generated from a bivariate Gaussian. As shown in Figure 7, the likely spanning tree T is the one containing three component trees, each spanning the points within a community, and two long edges with large $\|\vec{y}_k - \vec{y}_j\|$ binding the three trees together.

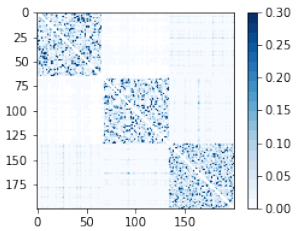
There is a large amount of uncertainty in this model, as can be seen via comparing Panels (a) and (b): (i) within a community, each point has a large number of other points in its neighborhood, hence there are multiple ways to form a tree with high posterior probability (that is, we have a low separability constant δ); (ii) when connecting two communities together, these candidate long edges do not differ much in the density (11) (due to the near polynomial density tail of generalized double Pareto), hence they are almost equally likely to enter T. As shown in Panel (c), most of the edges have a relatively low marginal connecting probability $\mathrm{pr}[T \ni (j,k) \mid y]$, indicating the posterior probability of T is scattered over a large number of different trees.



(a) One spanning tree sampled from the posterior distribution.



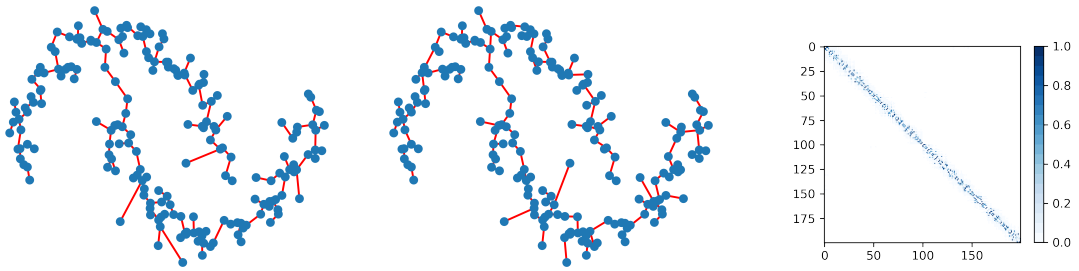
(b) Another spanning tree sampled from the posterior distribution.



(c) The matrix of marginal connecting probabilities $\mathrm{pr}[T \ni (j,k) \mid y]$.

Figure 7: High uncertainty of spanning trees when estimating a latent position graph with 3 communities, each formed by a bivariate Gaussian.

Next, we move to the case of a graph formed near two manifolds. We use the two-moon example provided in the scikit-learn package. As shown in Figure 8, each point has only one or a few points in the neighborhood, therefore, the posterior distribution of T is highly concentrated near the posterior mode. As a result, the two random posterior samples do not seem to differ much; and we have high values of $\sum_T \mathrm{pr}[(j,k) \in T, T \mid y]$ near the diagonal of the matrix.



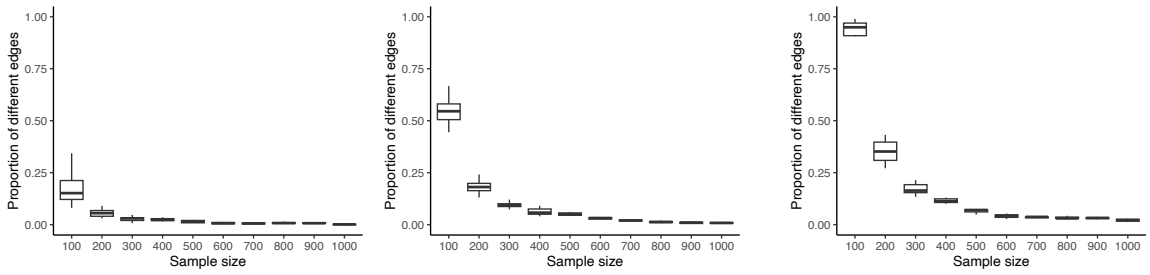
(a) One spanning tree sampled from the posterior distribution.

(b) Another spanning tree sampled from the posterior distribution.

(c) The matrix of marginal connecting probabilities $\mathrm{pr}[T \ni (j,k) \mid y]$.

Figure 8: Low uncertainty of spanning trees for when estimating a graph formed by the two-moon manifold.

We now empirically assess the concentration rate of the posterior distribution to the posterior mode. Under $p = 100, 200$ or 300, we generate a dense $p \times p$ covariance matrix Σ_0 using the scikit-learn package, and generate n data from $\mathrm{N}(0, \Sigma_0)$. Under each sample size n, we fit our model and obtain 100 sampled trees and compare them with the posterior mode. Figure 9 shows a very rapid concentration of posterior distribution to the posterior mode.



(a) $p = 100$

(b) $p = 200$

(c) $p = 300$

Figure 9: The posterior distribution concentrates rapidly to the posterior mode, as sample size n increases. Each boxplot is generated based on 100 trees drawn from the posterior distribution, and the proportions $|T \setminus \hat{T}|/(p-1)$ characterizing the difference between each tree sample and the posterior mode $\hat{T}$.

7. Spanning Tree Modeling of Brain Networks

We use our Bayesian spanning tree model to analyze data from a neuroscience study on human working memory. The study involves 20 human subjects participating in the Sternberg verbal memory test: first, each subject reads a list of g numbers on the screen, trying to memorize them for 2 seconds; then with the numbers removed from the screen, the subject answers if a particular number was in the list shown earlier. During this process, electroencephalogram (EEG) signals are obtained from electrode channels placed over 128 regions of interest (ROIs) of each subject's brain, over 5 seconds covering 512 times points. The goal

of this study is to not assess whether the subject correctly answers the question, but to find out how the brain acts differently when the subject is performing a simpler task with $g = 2$ numbers versus a more challenging task with $g = 6$ numbers.

We denote each EEG time series by $y_j^{[s,g,t]}$, as the signal collected from the jth ROI for subject s under the task load g at time t. To flexibly model these time series, we use the following hidden Markov model, based on K latent graph states, each modeled by a spanning tree $(T^{(1)}, \ldots, T^{(K)})$:

$$
\begin{aligned}
&y_1^{[s,g,t]}, y_2^{[s,g,t]}, \ldots, y_{128}^{[s,g,t]} \sim \mathrm{BSTM}(\tilde{T}_{s,g,t}; \tau), \\
&\mathrm{pr}[\tilde{T}_{s,g,0} = T^{(k)}] = q_{0,k}, \\
&\mathrm{pr}[\tilde{T}_{s,g,t} = T^{(k)} \mid \tilde{T}_{s,g,t-1} = T^{(k')}] = q_{k',k}^{[g]} \text{ for } t > 1, \\
&(q_{0,1}, \ldots, q_{0,K}) \sim \mathrm{Dir}(0.5, \ldots, 0.5), \\
&(q_{k',1}^{[g]}, \ldots, q_{k',K}^{[g]}) \sim \mathrm{Dir}(0.5, \ldots, 0.5) \text{ for } k' = 1, \ldots, K, \\
&\Pi_0(T^{(k)}) \propto 1,
\end{aligned}
$$

where $\mathrm{BSTM}(\tilde{T}; \tau)$ represents the Bayesian spanning tree model with the tree $\tilde{T}$ and the density (11) with the scale parameter τ; $(q_{0,1}, \ldots, q_{0,K})$ are the initial probability distribution for the K states. To enable borrowing of information across subjects and tasks, we let the parameters τ, $q_{0,k}$'s and the dictionary of trees $T^{(k)}$'s to be shared across subjects and tasks. On the other hand, to characterize the difference between two tasks, we set each $q_{k',k}^{[g]}$, the transition probability from state k' to state k, to be different according to the task load $g = 2$ or 6.

We use the Dirichlet distribution with concentration parameter 0.5 to induce approximate sparsity in the values of the initial and transition probabilities, and we set $K = 20$ for as an upper bound. In posterior samples, we found the maximum number of states used by the model is only 5, indicating $K = 20$ is sufficient as an upper bound.

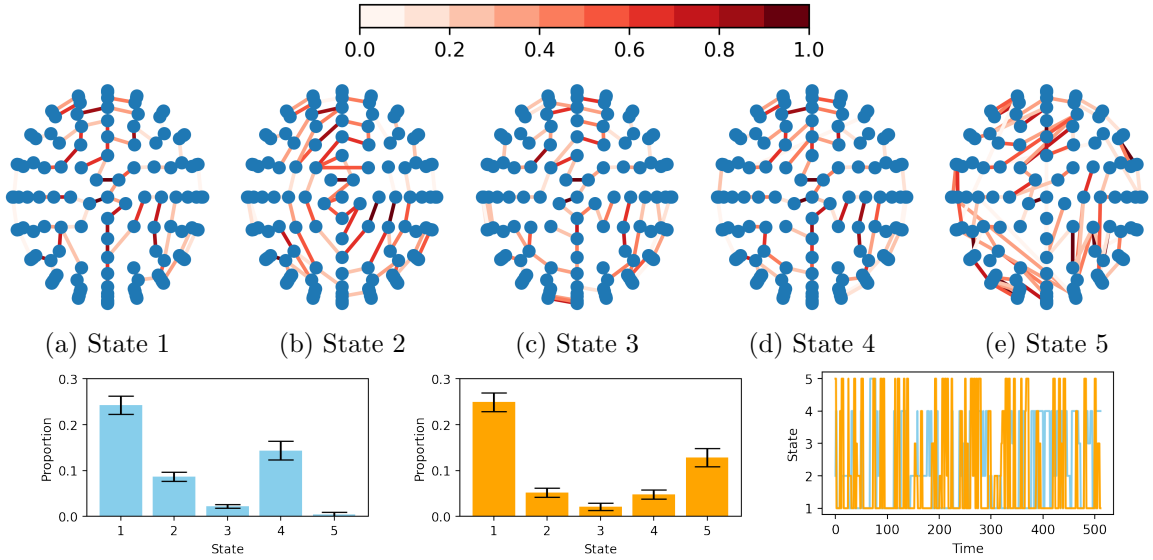
We run our MCMC algorithm for $20,000$ iterations, and discard the initial $10,000$ as a burn-in. Figure 10 shows the results for the data analysis. We plot the posterior mode spanning tree for each latent state, while showing the uncertainty on each edge using the marginal connecting probability.

The results are quite interpretable — as shown in Panels a and d, the two dominating states correspond to having each ROI connect to another that is spatially close. There is separation between the front of the brain (upper part in each plot) and the rear (lower part). Comparing the task of memorizing $g = 2$ numbers (Panel f) against the one of $g = 6$ numbers (Panel g), the latter involves more time spent in the State 5, during which the brain is more active and has more long-range connectivities over the brain (Panel e).

To validate our trained model, we use a previously reserved set of EEG data collected from another 20 subjects (hence 40 time series), and using our estimated model to classify whether a time series is likely to be collected under $g = 2$ or $g = 6$. Specifically, for each testing time series $\tilde{y}^{[s,\tilde{g},t]} = (\tilde{y}_1^{[s,\tilde{g},t]}, \tilde{y}_2^{[s,\tilde{g},t]}, \ldots, \tilde{y}_{128}^{[s,\tilde{g},t]})_t$, and for $g = 2$ or 6, we sample the latent state assignments given $\hat{T}^{(k)}$'s fixed at the posterior mode, $\hat{q}_{0,k}$'s and $\hat{q}_{k',k}^{[g]}$'s fixed at the posterior mean from the trained model, over the course of $10,000$ MCMC iterations. We average the last $5,000$ iterations to compute the likelihood $\Pi(\tilde{y}^{[s,\tilde{g},t]} \mid \hat{T}^{(k)}, \hat{q}_{0,k}, \hat{q}_{k',k}^{[g]})$

marginalizing out the state assignment. Comparing $g = 2$ and 6, we obtain the classification probability:

$$\text{pr}(g = 2 \mid .) = \frac{\Pi(\tilde{y}^{[s,\tilde{g},t]} \mid \hat{T}^{(k)}, \hat{q}_{0,k}, \hat{q}^{[2]}_{k',k} \text{ all } k, k')}{\Pi(\tilde{y}^{[s,\tilde{g},t]} \mid \hat{T}^{(k)}, \hat{q}_{0,k}, \hat{q}^{[2]}_{k',k} \text{ all } k, k') + \Pi(\tilde{y}^{[s,\tilde{g},t]} \mid \hat{T}^{(k)}, \hat{q}_{0,k}, \hat{q}^{[6]}_{k',k} \text{ all } k, k')}.$$



(a) State 1 (b) State 2 (c) State 3 (d) State 4 (e) State 5

(f) Distribution of state assignment for memory task load $g = 2$.

(g) Distribution of state assignment for memory task load $g = 6$.

(h) Comparison of the two time series in the state assignment for one subject.

Figure 10: Results of analyzing the electroencephalogram (EEG) time series collected over 128 channels on the human brain, using the Hidden Markov Model with the spanning trees as the latent states. The posterior distribution contains five latent states (Panels a-e, viewed from the top of the head), where each panel shows the posterior mode spanning tree with the color representing the marginal connecting probability. Comparing the task of memorizing $g = 2$ numbers (Panel f) to the one of $g = 6$ numbers (Panel g), the latter involves more time spent in the State 5, during which the brain is more active and has more long-range connectivities over the brain (Panel e). Panel h shows a comparison in the times series of the latent state assignments, when a given subject is memorizing $g = 2$ (blue) versus $g = 6$ (orange) numbers.

Using $g = 2$ if $\text{pr}(g = 2 \mid .) > 0.5$ and $g = 6$ otherwise, we obtain a low misclassification rate of 15% when comparing with true $\tilde{g}$. In the receiver operating characteristic curve, we calculate the area under the curve (AUC) and obtain 89%. This suggests that our model provides an adequate characterization the differences between the groups, given the low signal-to-noise ratio in EEG data.

To compare, we also run the same hidden Markov model except with each latent state modeled by a multivariate Gaussian distribution $\mathrm{N}(\vec{\mu}_k, \Omega_k^{-1})$, with Ω_k estimated via the graphical lasso with $\alpha = 0.5$. Setting K at 20, we obtain a competitive validation result with the misclassification rate at 15% and AUC at 85%; however, a major drawback is that this Gaussian hidden Markov model involves 16 latent states with nontrivial probabilities ($\geq 5\%$), hence is much more difficult to interpret than the 5 states from the spanning tree-based model. In addition, we test the Gaussian hidden Markov model with K reduced to 10, and it leads to much worse validation performance, with the misclassification rate at 45% and AUC at 53%, which is almost close to a random guess; similarly, we test $\alpha = 0.1$ and $\alpha = 1$ in the graphical lasso, and obtain similarly poor performance. Therefore, the Gaussian hidden Markov model is much less parsimonious than the spanning tree-based model. Lastly, we fit logistic regression on the raw data, which leads to misclassification rate at 39% and an AUC of 55%. This large classification error is likely due to the high dimension and noisiness of EEG data, whereas fitting a graphical model seems to improve the signal-to-noise ratio.

8. Discussion

In this article, we propose to use the spanning tree as a restricted graph model to estimate the backbone of a latent graph. We study its mathematical properties and demonstrate good performances in both theory and applications in recovering important subsets of edges. There are several interesting extensions worth exploring. First, instead of using only one spanning tree, one could consider the union of multiple spanning trees as a more flexible graphical model; there are several open questions that need to be addressed, such as the identifiability issue due to the overlap of multiple trees as well as its finite sample recovery theory. Second, one could consider the spanning tree as the opposite extremity of the clique graph [a completely connected graph with $p(p-1)/2$ edges]; therefore, one could view the broad class of connected graphs as in some continuum between those two extremal graphs, potentially creating useful new graph models. This is related to thin junction tree (Bach and Jordan, 2001) and bounded treewidth Bayesian networks (Elidan and Gould, 2008).

Acknowledgement

This work was partially supported by grant DMS-2319551 of the United States National Science Foundation, grants R01-MH118927-01 and R01ES027498 of the United States National Institutes of Health, and grant N00014-21-1-2510-01 of the United States Office of Naval Research.

References

David J Aldous. The Random Walk Construction of Uniform Spanning Trees and Uniform Labelled Trees. *SIAM Journal on Discrete Mathematics*, 3(4):450–465, 1990.

Artin Armagan, David B Dunson, and Jaeyong Lee. Generalized Double Pareto Shrinkage. *Statistica Sinica*, 23(1):119, 2013.

Francis Bach and Michael Jordan. Thin Junction Trees. In *Advances in Neural Information Processing Systems*, volume 14, 2001.

Ravindra B Bapat. *Graphs and Matrices*, volume 27. Springer, 2010.

Sumanta Basu and George Michailidis. Regularized Estimation in Sparse High-Dimensional Time Series Models. *The Annals of Statistics*, 43(4):1535–1567, 2015.

Peter J Bickel and Elizaveta Levina. Covariance Regularization by Thresholding. *The Annals of Statistics*, 36(6):2577–2604, 2008a.

Peter J Bickel and Elizaveta Levina. Regularized Estimation of Large Covariance Matrices. *The Annals of Statistics*, 36(1):199–227, 2008b.

Francis Buekenhout and Monique Parker. The Number of Nets of the Regular Convex Polytopes. *Discrete Mathematics*, 186(1-3):69–94, 1998.

V. V. Buldygin and Yu. V. Kozachenko. *Metric Characterization of Random Variables and Random Processes*. American Mathematical Society, 2000. ISBN 0821805339.

Simon Byrne and A Philip Dawid. Structural Markov Graph Laws for Bayesian Model Uncertainty. *The Annals of Statistics*, 43(4):1647–1681, 2015.

Diana Cai, Trevor Campbell, and Tamara Broderick. Edge-Exchangeable Graphs and Sparsity. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4249–4257, 2016a.

T Tony Cai, Weidong Liu, and Harrison H Zhou. Estimating Sparse Precision Matrix: Optimal Rates of Convergence and Adaptive Estimation. *The Annals of Statistics*, 44 (2):455–488, 2016b.

T Tony Cai, Zhao Ren, and Harrison H Zhou. Estimating Structured High-Dimensional Covariance and Precision Matrices: Optimal Rates and Adaptive Estimation. *Electronic Journal of Statistics*, 10(1):1–59, 2016c.

Xuan Cao, Kshitij Khare, and Malay Ghosh. Posterior Graph Selection and Estimation Consistency for High-Dimensional Bayesian DAG Models. *The Annals of Statistics*, 47 (1):319–348, 2019.

Seth Chaiken and Daniel J Kleitman. Matrix Tree Theorems. *Journal of Combinatorial Theory, Series A*, 24(3):377–381, 1978.

C Chow and Cong Liu. Approximating Discrete Probability Distributions With Dependence Trees. *IEEE transactions on Information Theory*, 14(3):462–467, 1968.

C Chow and T Wagner. Consistency of an Estimate of Tree-Dependent Probability Distributions. *IEEE Transactions on Information Theory*, 19(3):369–371, 1973.

David Roxbee Cox and Nanny Wermuth. *Multivariate Dependencies: Models, Analysis and Interpretation*, volume 67. CRC Press, 1996.

Hongsheng Dai. Perfect Sampling Methods for Random Forests. *Advances in Applied Probability*, 40(3):897–917, 2008.

Arthur P Dempster. Covariance Selection. *Biometrics*, 28(1):157–175, 1972.

Edsger W Dijkstra. A Note on Two Problems in Connexion With Graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

Adrian Dobra and Alex Lenkoski. Copula Gaussian Graphical Models and Their Application to Modeling Functional Disability Data. *The Annals of Applied Statistics*, 5(2A):969–993, 2011.

Adrian Dobra, Chris Hans, Beatrix Jones, Joseph R Nevins, Guang Yao, and Mike West. Sparse Graphical Models for Exploring Gene Expression Data. *Journal of Multivariate Analysis*, 90(1):196–212, 2004.

David L Donoho, Matan Gavish, and Iain M Johnstone. Optimal shrinkage of eigenvalues in the spiked covariance model. *The Annals of Statistics*, 46(4):1742, 2018.

Leo L Duan and Arkaprava Roy. Spectral Clustering, Bayesian Spanning Forest, and Forest Process. *Journal of the American Statistical Association*, in press:1–14, 2023.

David Durfee, Rasmus Kyng, John Peebles, Anup B Rao, and Sushant Sachdeva. Sampling Random Spanning Trees Faster Than Matrix Multiplication. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 730–742, 2017.

David Edwards, Gabriel CG De Abreu, and Rodrigo Labouriau. Selecting High-Dimensional Mixed Graphical Models Using Minimal AIC or BIC Forests. *BMC Bioinformatics*, 11 (1):1–13, 2010.

Gal Elidan and Stephen Gould. Learning Bounded Treewidth Bayesian Networks. In *Advances in Neural Information Processing Systems*, volume 21, 2008.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse Inverse Covariance Estimation With the Graphical Lasso. *Biostatistics*, 9(3):432–441, 2008.

Jerome H Friedman and Lawrence C Rafsky. Multivariate Generalizations of the Wald-Wolfowitz and Smirnov Two-Sample Tests. *The Annals of Statistics*, 7(4):697–717, 1979.

Edward I George and Robert E McCulloch. Stochastic Search Variable Selection. *Markov chain Monte Carlo in practice*, 68:203–214, 1995.

CJ Goh. *Duality in Optimization and Variational Inequalities*, volume 2. Taylor & Francis, 2002.

Peter J Green and Alun Thomas. Sampling Decomposable Graphs Using a Markov Chain on Junction Trees. *Biometrika*, 100(1):91–110, 2013.

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.

Søren Højsgaard, David Edwards, and Steffen Lauritzen. *Graphical Models With R*. Springer Science & Business Media, 2012.

Piotr Juszczak, David MJ Tax, Elżbieta Pe, and Robert PW Duin. Minimum Spanning Tree Based One-Class Classifier. *Neurocomputing*, 72(7-9):1859–1869, 2009.

David R Karger, Philip N Klein, and Robert E Tarjan. A Randomized Linear-Time Algorithm to Find Minimum Spanning Trees. *Journal of the ACM (JACM)*, 42(2):321–328, 1995.

Joseph B Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.

Richard A Levine and George Casella. Optimizing Random Scan Gibbs Samplers. *Journal of Multivariate Analysis*, 97(10):2071–2100, 2006.

Zhao Tang Luo, Huiyan Sang, and Bani Mallick. BAST: Bayesian Additive Regression Spanning Trees for Complex Constrained Domain. In *Advances in Neural Information Processing Systems*, volume 34, pages 90–102. Curran Associates, Inc., 2021.

Zhao Tang Luo, Huiyan Sang, and Bani Mallick. BAMDT: Bayesian Additive Semi-Multivariate Decision Trees for Nonparametric Regression. In *Proceedings of the 39th International Conference on Machine Learning*, pages 14509–14526. PMLR, June 2022. ISSN: 2640-3498.

Zhao Tang Luo, Huiyan Sang, and Bani Mallick. A Nonstationary Soft Partitioned Gaussian Process Model via Random Spanning Trees. *Journal of the American Statistical Association*, 0(0):1–12, 2023. ISSN 0162-1459.

Marina Meilă and Tommi Jaakkola. Tractable Bayesian Learning of Tree Belief Networks. *Statistics and Computing*, 16(1):77–92, 2006.

Marina Meilă and Michael I Jordan. Learning with Mixtures of Trees. *Journal of Machine Learning Research*, 1(Oct):1–48, 2000.

Mohamed Mosbah and Nasser Saheb. Non-Uniform Random Spanning Trees on Weighted Graphs. *Theoretical computer science*, 218(2):263–271, 1999.

Kazuo Murota. Discrete Convex Analysis. *Mathematical Programming*, 83(1-3):313–371, 1998.

Abhinav Natarajan, Willem van den Boom, Kristoforus Bryant Odang, and Maria de Iorio. On a Wider Class of Prior Distributions for Graphical Models. *Journal of Applied Probability*, pages 1–14, June 2023. ISSN 0021-9002, 1475-6072. Publisher: Cambridge University Press.

Nicholas G Polson and James G Scott. Shrink Globally, Act Locally: Sparse Bayesian Regularization and Prediction. *Bayesian Statistics*, 9(501-538):105, 2010.

Robert Clay Prim. Shortest Connection Networks and Some Generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957.

Pradeep Ravikumar, Martin J. Wainwright, Garvesh Raskutti, and Bin Yu. High-Dimensional Covariance Estimation by Minimizing L1-Penalized Log-Determinant Divergence. *Electronic Journal of Statistics*, 5:935–980, 2011. ISSN 19357524. doi: 10.1214/11-EJS631.

Alberto Roverato. Hyper Inverse Wishart Distribution for Non-Decomposable Graphs and Its Application to Bayesian Inference for Gaussian Graphical Models. *Scandinavian Journal of Statistics*, 29(3):391–411, 2002.

Philipp Rütimann and Peter Bühlmann. High Dimensional Sparse Covariance Estimation via Directed Acyclic Graphs. *Electronic Journal of Statistics*, 3:1133–1160, 2009.

Loïc Schwaller, Stéphane Robin, and Michael Stumpf. Closed-Form Bayesian Inference of Graphical Model Structures by Averaging Over Trees. *Journal de la Société Française de Statistique*, 160(2):1–23, 2019.

Kean Ming Tan, Daniela Witten, and Ali Shojaie. The Cluster Graphical Lasso for Improved Estimation of Gaussian Graphical Models. *Computational Statistics & Data Analysis*, 85: 23–36, 2015.

Vincent YF Tan, Animashree Anandkumar, Lang Tong, and Alan S Willsky. A Large-Deviation Analysis of the Maximum-Likelihood Learning of Markov Tree Structures. *IEEE Transactions on Information Theory*, 57(3):1714–1735, 2011.

Leonardo V. Teixeira, Renato M. Assunção, and Rosangela H. Loschi. Bayesian Space-Time Partitioning by Sampling and Pruning Spanning Trees. *Journal of Machine Learning Research*, 20(85):1–35, 2019.

Prejaas Tewarie, Edwin van Dellen, Arjan Hillebrand, and Cornelis J Stam. The Minimum Spanning Tree: An Unbiased Method for Brain Network Analysis. *Neuroimage*, 104: 177–188, 2015.

Hao Wang. Bayesian Graphical Lasso Models and Efficient Posterior Computation. *Bayesian Analysis*, 7(4):867–886, 2012.

Mike West. On Scale Mixtures of Normal Distributions. *Biometrika*, 74(3):646–648, September 1987.

Halbert White. Maximum Likelihood Estimation of Misspecified Models. *Econometrica: Journal of the Econometric Society*, 50(1):1–25, 1982.

David Bruce Wilson. Generating Random Spanning Trees More Quickly Than the Cover Time. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 296–303, 1996.

Xiaofan Xu and Malay Ghosh. Bayesian Variable Selection and Estimation for Group Lasso. *Bayesian Analysis*, 10(4):909–936, 2015.

Ming Yuan and Yi Lin. Model Selection and Estimation in the Gaussian Graphical Model. *Biometrika*, 94(1):19–35, 2007.

Appendix A. Appendix

A.1 Proof of Theorem 3

Proof The proof slightly extends Chaiken and Kleitman (1978), which states

$$z_q = \frac{1}{p}\prod_{j=2}^{p}\lambda_{(j)}(L_q),$$

where $\lambda_{(2)}(L_q) \leq \ldots \lambda_{(p)}(L_q)$ are the largest $p-1$ eigenvalues of L_q. Since the smallest eigenvalue of L_q is 0 with a corresponding eigenvector $1_p/\sqrt{p}$, adding $J/p/p$ to L_q and taking the determinant yields the result. ■

A.2 Proof of Theorem 4

Proof

For ease of notation, let $M = I - B_{[-s]}(B_{[-s]}^{\mathrm{T}}B_{[-s]})^{-1}B_{[-s]}^{\mathrm{T}}$. After removing edge e_s, we obtain two separated subgraphs, denoted by G_1 and G_2. Considering two nodes j_1 and j_2 in the same connected subgraph (without loss of generality, in G_1), we use a p-element binary vector $\vec{x}(j_1, j_2)$ to represent auxiliary edge (j_1, j_2), with the j_1th element equal to 1 and j_2th element equal to -1, and all other elements 0.

We know there is a path in G_1 that connects the nodes j_1 and j_2. We can represent the auxiliary edge $\vec{x}(j_1, j_2)$ as a linear combination of the columns of $B_{[-s]}$ using the edges in the path. That is, there exists a_h's taking value in $\{-1, 1\}$ such that:

$$\vec{x}(j_1, j_2) = \sum_{h:e_h\in\mathrm{path}(j_1,j_2)} a_h\vec{B}_h.$$

This means that $\vec{x}(j_1, j_2)$ is in the column space of $B_{[-s]}$, therefore $M\vec{x}(j_1, j_2) = \vec{0}$.

Multiplying M to $\vec{x}(j_1, j_2)$ corresponds to creating a contrast between the columns j_1 and j_2 of M. Hence if j_1, j_2 are in the same subgraph, $M_{l,j_1} = M_{l,j_2}$ for all $l = 1, \ldots, p$; since M is symmetric, $M_{j_1,l} = M_{j_2,l}$ for all $l = 1, \ldots, p$. Therefore, for two index sets $V_1 = \{j : j \in G_1\}$ and $V_2 = \{k : k \in G_2\}$, the matrix M can be divided into four blocks; within each the elements have the same value: $M_{j_1,j_2} = m_{1,1}$ for $j_1 \in V_1, j_2 \in V_1$, $M_{j_2,j_1} = M_{j_1,j_2} = m_{1,2}$ for $j_1 \in V_1, j_2 \in V_2$, $M_{j_1,j_2} = m_{2,2}$ for $j_1 \in V_2, j_2 \in V_2$.

Since $\vec{B}_s = \vec{x}(j, k)$ [or $-\vec{x}(j, k)$, we will use the former without loss of generality], we know:

$$M\vec{B}_s = (M_{l,j} - M_{l,k})_{l=1,\ldots,p},$$

where $M_{l,j} - M_{l,k} = m_{1,1} - m_{1,2}$ if $l \in V_1$, and $M_{l,j} - M_{l,k} = m_{1,2} - m_{2,2}$ if $l \in V_2$. That is, the vector $M\vec{B}_s$ is also partitioned into two parts according to $l \in V_1$ or $l \in V_2$.

It remains to show those two values are distinct $m_{1,1} - m_{1,2} \neq m_{1,2} - m_{2,2}$. We use proof by contradiction. Supposing equality holds, we have $M\vec{B}_s = \vec{1}_p c$, for some scalar $c \in \mathbb{R}$. Since $B = [B_{[-s]}\ \vec{B}_s]$ has full rank $p-1$, $\vec{B}_s$ should not be in the column space of $B_{[-s]}$; hence $c \neq 0$. However, we know $1_p^{\mathrm{T}}M = 1_p^{\mathrm{T}} - 1_p^{\mathrm{T}}B_{[-s]}(B_{[-s]}^{\mathrm{T}}B_{[-s]})^{-1}B_{[-s]}^{\mathrm{T}} = 1_p^{\mathrm{T}}$, as

each column of $B_{[-s]}$ adds up to zero — that is, $1_p^{\mathrm{T}} M\vec{B}_s = 1_p^{\mathrm{T}}\vec{B}_s = 0$, which contradicts $\vec{1}_p^{\mathrm{T}}\vec{1}_p c = pc \neq 0$. ■

A.3 Proof of Lemma 8

The proof is trivial by checking each element of $B\Psi^{-1}B^{\mathrm{T}}$.

A.4 Proof of Lemma 9

Proof Introduce augmented matrices $B^* = (B\ 1_p)$ and $\Lambda^* = \mathrm{diag}\{\Psi^{-1}, \epsilon\}$. As both matrices are square and full rank, we have

$$|\hat{\Omega}| = |B^*\Lambda^* B^{*\mathrm{T}}| = |B^*||\Lambda^*||B^{*\mathrm{T}}| = |B^*B^{*\mathrm{T}}||\Lambda^*| = |BB^{\mathrm{T}} + J||\Lambda^*|.$$

Using the previous lemma with binary $(A_\phi)_{j,k} \in \{0,1\}$, BB^{T} is the Laplacian of an unweighted spanning tree $\tilde{T}$, with eigenvalues $\tilde{\lambda}_p \geq \ldots \geq \tilde{\lambda}_2 > \tilde{\lambda}_1 = 0$, and

$$|BB^{\mathrm{T}} + J| = p\prod_{l=2}^{p}\tilde{\lambda}_l.$$

Using Kirchhoff's matrix tree theorem (Buekenhout and Parker, 1998), the product of the top $(p-1)$ eigenvalues of the Laplacian for graph G is related to the number of spanning trees $t(G)$ contained in G via

$$\prod_{l=2}^{p}\tilde{\lambda}_l = pt(G).$$

As $t(\tilde{T}) = 1$, $|BB^{\mathrm{T}} + J| = p^2$. Combining the above, we have $|\hat{\Omega}| = p^2\epsilon|\Psi^{-1}|$. ■

A.5 Proof of Theorem 10

Proof We first state the theorem from Goh (2002).

Theorem 14 *For a complete graph with edge weights $\{W_{j,k}\}$, a spanning tree T is a minimum spanning tree if and only if: for every edge $(h,l) \notin T$, $W_{j,k} \leq W_{h,l}$ for every $(j,k) \in T : (j,k) \in path(h,l)$.*

We prove by contradiction that we must have $W_{j,k} \neq W_{h,l}$ for $(h,l) \notin \cup_{m=1}^{M} T_0^{(m)}$. Suppose $W_{j,k} = W_{h,l}$ for a certain (h,l) not in the tree and a $(j,k) \in \mathrm{path}(h,l)$ in the minimum spanning tree $T_0^{(m)}$. Since $(j,k) \in \mathrm{path}(h,l)$, we can disconnect (j,k) and replace it with (h,l); we have a new path such that $(h,l) \in \mathrm{path}(j,k)$, forming a new minimum spanning tree, which contradicts the condition $(h,l) \notin \cup_{m=1}^{M} T_0^{(m)}$. ■

A.6 Proof of Theorem 11

Proof For better clarity, in this proof, we use simplified notations $m_{j,k} := W_{n:j,k}$, $M_{j,k} := W_{0:j,k}$, $S := S_n$ and $\Sigma := \Sigma_0$. The posterior mode corresponds to

$$\hat{T} = \arg\min_{T} \sum_{e_s=(j,k)\in T} (S_{j,j} + S_{k,k} - 2S_{j,k}),$$

where $j \neq k$, and $m_{j,k} = (S_{j,j} + S_{k,k} - 2S_{j,k})$. Using $S_{j,k} = 1/n \sum_{i=1}^{n} y_j^{(i)} y_k^{(i)}$, we have

$$m_{j,k} = n^{-1} \sum_{i=1}^{n} \{y_j^{(i)} - y_k^{(i)}\}^2.$$

Letting $z_{j,k}^{(i)} = y_j^{(i)} - y_k^{(i)}$, $z_{j,k}^{(i)}$ has mean 0, and

$$\begin{aligned} \mathbb{E}\exp(tz_{j,k}^{(i)}) &\overset{(a)}{\leq} \sqrt{\mathbb{E}\exp(2ty_j^{(i)})\mathbb{E}\exp(-2ty_k^{(i)})} \\ &\overset{(b)}{\leq} \exp(4t^2\lambda^2/2) \end{aligned}$$

where (a) uses Cauchy-Schwarz inequality and (b) uses $\mathbb{E}\exp(tX) \leq \exp(\lambda^2 t^2/2)\ \forall t \in \mathbb{R}$ for the $\lambda-$sub-Gaussian random variable. Therefore, $z_{j,k}^{(i)}$ is 2λ-sub-Gaussian.

We have the mean of $\{z_{j,k}^{(i)}\}^2$ as

$$M_{j,k} = \mathbb{E}\{z_{j,k}^{(i)}\}^2 = \Sigma_{j,j} + \Sigma_{k,k} - 2\Sigma_{j,k},$$

and it is not hard to see that $M_{j,k} = \mathbb{E}m_{j,k} = \mathbb{E}(S_{j,j} + S_{k,k} - 2S_{j,k})$ as well. Since Σ is positive definite, letting x be a p-element vector with $x_j = 1$, $x_k = -1$ and all other elements 0, we have $M_{j,k} = x^{\mathrm{T}}\Sigma x > 0$. Further, $M_{j,k} \leq 4\max(\Sigma_{j,j}, \Sigma_{k,k})$ due to $|\Sigma_{j,k}| \leq \sqrt{\Sigma_{j,j}\Sigma_{k,k}} \leq \max(\Sigma_{j,j}, \Sigma_{k,k})$.

1. Show that $(m_{j,k} - M_{j,k})$ is sub-exponential via the Bernstein's condition.

Our next goal is to check the Bernstein's moments condition for $w_{j,k}^{(i)} = \{z_{j,k}^{(i)}\}^2 - M_{j,k}$, for all $q = 2, 3, \ldots$:

$$|\mathbb{E}\{w_{j,k}^{(i)}\}^q| \leq q! 2^{-1} v_{j,k}^2 \beta^{q-2}, \tag{21}$$

where $v_{j,k}^2$ is the variance of $w_{j,k}^{(i)}$ and we want to find a valid constant β that satisfies the inequality.

We now focus on a given index (i, j, k). For ease of notation, we omit the subscript (j, k) and superscript i. For $q = 2$, (21) holds trivially, hence we now focus on $q \geq 3$. Using Lemma 1.4 from Buldygin and Kozachenko (2000), for any $q > 0$, the moments of a 2λ-sub-Gaussian random variable has

$$\mathbb{E}|z|^q \leq 2(q/e)^{(q/2)}(2\lambda)^q. \tag{22}$$

We have for any $q = 3, 4, \ldots$:

$$\begin{aligned} \mathbb{E}|z^2|^q/q! &\leq 2(2q/e)^q(2\lambda)^{2q} \\ &= 2^{3q+1}(q/e)^q\lambda^{2q}/q! \\ &\overset{(a)}{\leq} (2/e)(8\lambda^2)^q \end{aligned} \tag{23}$$

where (a) uses $q! \geq e(q/e)^q$.

$$
\begin{aligned}
|\frac{\mathbb{E}w^q}{q!}|^{1/q} = |\frac{\mathbb{E}(z^2 - M_{j,k})^q}{q!}|^{1/q} \\
&\overset{(a)}{\leq} (\frac{\mathbb{E}|z^2 - M_{j,k}|^q}{q!})^{1/q} \\
&\overset{(b)}{\leq} \frac{(\mathbb{E}z^{2q})^{1/q} + M_{j,k}}{(q!)^{1/q}} \\
&\overset{(c)}{\leq} (2/e)^{1/q} 8\lambda^2 + 4\lambda^2/\{e^{1/q}(q/e)\} \\
&\overset{(d)}{\leq} 8\lambda^2 + 4\lambda^2/\{e^{1/3}(3/e)\} \\
&< 11\lambda^2.
\end{aligned}
$$

where (a) uses $|\mathbb{E}x| \leq \mathbb{E}|x|$; (b) is due to the Minkowski inequality, $(\mathbb{E}|X+Y|^q)^{1/q} \leq (\mathbb{E}|X|^q)^{1/q} + (\mathbb{E}|Y|^q)^{1/q}$ for any $q \geq 1$; (c) uses (23), $q! \geq e(q/e)^q$, and $M_{j,k} \leq 4\max(\Sigma_{j,j}, \Sigma_{k,k})$ and Lemma 1.2 from Buldygin and Kozachenko (2000), that for λ-sub-Gaussian $y_j^{(i)}$, its variance $\Sigma_{j,j} \leq \lambda^2$; (d) uses $(2/e)^{1/q} < 1$, and $e^{1/q}(q/e)$ is increasing in $q \geq 3$. Therefore,

$$|\mathbb{E}w^q| \leq q!(11\lambda^2)^q,$$

hence the next goal is to find β such that $q!(11\lambda^2)^q \leq q!2^{-1}v^2\beta^{q-2}$. Slight manipulation yields that, for $q \geq 3$, we need β large enough such that

$$(11\lambda^2)^{q-2} \leq \frac{v^2}{2(11\lambda^2)^2}\beta^{q-2}.$$

In addition,

$$v^2 = \mathbb{E}w^2 = \mathbb{E}(z^2 - M_{j,k})^2 = \mathbb{E}z^4 - M^2 \leq \mathbb{E}z^4 \overset{(a)}{\leq} 2(4/e)^2(2\lambda)^4 \leq 70\lambda^4,$$

where (a) uses the sub-Gaussian moment bound (22). Therefore, we know $v^2/\{2(11\lambda^2)^2\} < v^2/(70\lambda^4) \leq 1$, and a valid constant is

$$\beta \geq \{2(11\lambda^2)^2/(v^2)\} \times (11\lambda^2) = 2(11\lambda^2)^3/(v^2).$$

Further, note that $\beta \geq 2(11\lambda^2)^3/(70\lambda^4) \geq 38\lambda^2 > v$.

Now including the index (i,j,k), $w_{j,k}^{(i)}$ is sub-exponential with parameters $v_{j,k}$ and $\beta_{j,k}$. This gives the Bernstein inequality for any $\epsilon > 0$,

$$
\begin{aligned}
\Pr(|m_{j,k} - M_{j,k}| \geq \epsilon) = \Pr(|\frac{1}{n}\sum_{i=1}^{n} w_{j,k}^{(i)}| \geq \epsilon) &\leq 2\exp\{-\frac{n\epsilon^2}{2(v_{j,k}^2 + \beta_{j,k}\epsilon)}\} \\
&\overset{(a)}{\leq} 2\exp\{-\frac{n\epsilon^2}{2(\beta_{j,k}^2 + \beta_{j,k}\epsilon)}\}, \\
&\overset{(b)}{\leq} 2\exp\{-\frac{n\epsilon^2}{2(\beta_0^2 + \beta_0\epsilon)}\},
\end{aligned}
$$

where (a) uses $v \leq \beta$, and in (b) we set $\beta_0 = \max_{\text{all}(j,k)} \beta_{j,k}$.

2. Analyze Prim's greedy algorithm to find the concentration inequality

Let T_0 be the minimum spanning tree based on the oracle covariance:

$$T_0 = \arg\min_T \sum_{e_s=(j,k)\in T} M_{j,k},$$

where $M_{j,k} = \Sigma_{j,j} + \Sigma_{k,k} - 2\Sigma_{j,k}$. We can now analyze the Prim's greedy algorithm applied on $m_{j,k}$'s and bound the probability of finding a spanning tree $\hat{T}$ different than T_0.

For simplicity, let us start with the case when the oracle minimum spanning spanning tree is unique. At the step with two sets of nodes U and $\bar{U}$, if an edge $(j',k') \in \mathcal{E}(U,\bar{U})$ (the edge set between U and $\bar{U}$) but $(j',k') \notin T_0$, then there must be an edge $(j,k) \in \mathcal{E}(U,\bar{U})$, $(j,k) \in T_0$ such that $(j,k) \in \text{path}(j',k')$. By the path optimality of the oracle minimum spanning tree, $M_{(j,k)} < M_{(j',k')}$. The probability of not having $m_{j,k} < m_{j',k'}$ has:

$$\begin{aligned}
\text{pr}(m_{j,k} \geq m_{j',k'}) &= \text{pr}\{(m_{j,k} - M_{j,k}) - (m_{j',k'} - M_{j',k'}) \geq (M_{j',k'} - M_{j,k})\} \\
&\overset{(a)}{\leq} \text{pr}\{(m_{j,k} - M_{j,k}) - (m_{j',k'} - M_{j',k'}) \geq \delta\} \\
&\overset{(b)}{\leq} \text{pr}\{|m_{j,k} - M_{j,k}| + |m_{j',k'} - M_{j',k'}| > \delta\} \\
&\overset{(c)}{\leq} \text{pr}\{|m_{j,k} - M_{j,k}| > \delta/2\} + \text{pr}\{|m_{j',k'} - M_{j',k'}| > \delta/2\} \\
&\leq 4\exp[-\frac{n\delta^2}{8(\beta_0^2 + \beta_0\delta)}]
\end{aligned}$$

where (a) uses the condition $M_{(j',k')} - M_{(j,k)} \geq \delta$; (b) is due to the former implies the latter; (c) uses the union bound.

Given a node set U and its complement $\bar{U}$, denote the event $G(U)$ as picking *any* edge (j',k') from $\mathcal{E}(U,\bar{U})$ but not in T_0. Let $\tilde{n}(U)$ be the number of edges in $\mathcal{E}(U,\bar{U}) \cap E_{T_0}$, then using union bound we have:

$$\begin{aligned}
\text{pr}\{G(U)\} &\leq \{|\mathcal{E}(U,\bar{U})| - \tilde{n}(U)\}4\exp[-\frac{n\delta^2}{8(\beta_0^2 + \beta_0\delta)}] \\
&\overset{(a)}{\leq} \{|\mathcal{E}(U,\bar{U})| - 1\}4\exp[-\frac{n\delta^2}{8(\beta_0^2 + \beta_0\delta)}],
\end{aligned} \tag{24}$$

where (a) uses $\tilde{n}(U) \geq 1$.

Letting $\{U_1, U_2, \ldots, U_p\}$ be the sequence used to obtain the minimum spanning tree in the Prim's algorithm, with $U_1 = \{1\}$ and $U_p = \{1,\ldots,n\}$, we have

$$\begin{aligned}
&\text{pr}\{\bigcup_{k=1}^{p} G(U_k)\} \leq \sum_{k=1}^{p}(|\mathcal{E}(U_k,\bar{U}_k)| - 1)4\exp[-\frac{n\delta^2}{8(\beta_0^2 + \beta_0\delta)}] \\
&\overset{(a)}{\leq} \frac{2p^3}{3}\exp[-\frac{n\delta^2}{8(\beta_0^2 + \beta_0\delta)}] \\
&= \frac{2}{3}\exp[-\frac{n\delta^2}{8(\beta_0^2 + \beta_0\delta)} + 3\log p]
\end{aligned}$$

where (a) uses $\sum_{k=1}^{p}\{(p-k)k-1\} = p(p^2-7)/6 \le p^3/6$.

Now consider the case when the oracle minimum spanning tree is not unique. Given a node set U and its complement $\bar{U}$, denote the event $G(U)$ as picking *any* edge (j', k') in the edges between $(U, \bar{U})$ but not in one of T_0's. Letting $\tilde{n}(U)$ be the number of edges in $\mathcal{E}(U, \bar{U}) \cap (\cup_{k=1}^{K} E_{T_0^{(k)}})$, clearly, $\tilde{n}(U) \ge 1$, hence (24) still holds. Therefore, the rest of the result follows. ■

A.7 Calculation of the normalizing constant in the degree-based prior

Let $r = \sum_{j=1}^{p} v_j$. Since $\eta = vv^{\mathrm{T}}$, we have $z(\eta) = p^{-1}\prod_{j=2}^{p}\lambda_{(j)}(L) = |(\mathrm{diag}(rv) - vv^{\mathrm{T}})_{2:p,2:p}|$ due to the proof of Theorem 1, and the equivalence between the $1/p$ of the product of the top $(p-1)$ eigenvalues and the cofactor. Therefore using the matrix determinant lemma

$$z(\eta) = r^{p-1}(\prod_{j=2}^{p} v_j)(1 - r^{-1}v_{2:n}^{\mathrm{T}}\mathrm{diag}(v_{2:p}^{-1})v_{2:p}) = r^{p-2}(\prod_{j=2}^{p} v_j)(r - \sum_{j=2}^{p} v_j) = r^{p-2}\prod_{j=1}^{p} v_j.$$

A.8 Calculation of the multivariate generalized double Pareto density

Letting $\vec{\beta} = \vec{y}_j - \vec{y}_k \in \mathbb{R}^p$, we first multiply with $\Pi(\lambda_s^2)$ and integrate out λ_s^2

$$\begin{aligned}
&\int_0^\infty \left(\frac{1}{2\pi\tau^2\lambda_s^2}\right)^{n/2} \exp\left[-\frac{\|\vec{\beta}\|^2}{2\tau^2\lambda_s^2}\right] \frac{\left(\frac{\kappa_s^2}{2}\right)^{\frac{n+1}{2}} \left(\lambda_s^2\right)^{\frac{n+1}{2}-1}}{\Gamma\left(\frac{n+1}{2}\right)} \exp\left[-\kappa_s^2\lambda_s^2/2\right] \mathrm{d}\lambda_s^2 \\
&= \frac{1}{\tau^n}\left(\frac{1}{2\pi}\right)^{\frac{n-1}{2}} \frac{1}{\Gamma\left(\frac{n+1}{2}\right)} \left(\frac{\kappa_s^2}{2}\right)^{\frac{n-1}{2}} \\
&\times \int_0^\infty \left(\frac{1}{2\pi\lambda_s^2}\right)^{1/2} \exp\left[-\frac{\|\vec{\beta}\|^2}{2\tau^2\lambda_s^2}\right] \left(\frac{\kappa_s^2}{2}\right) \exp\left[-\kappa_s^2\lambda_s^2/2\right] \mathrm{d}\lambda_s^2 \\
&= \frac{1}{\tau^n}\left(\frac{1}{2\pi}\right)^{\frac{n-1}{2}} \frac{1}{\Gamma\left(\frac{n+1}{2}\right)} \left(\frac{\kappa_s^2}{2}\right)^{\frac{n-1}{2}} \left(\frac{\kappa_s}{2}\right) \exp\left[-\kappa_s\frac{\|\vec{\beta}\|}{\tau}\right].
\end{aligned}$$

Next, we multiply with $\Pi(\kappa_s)$ and integrate out κ_s,

$$\begin{aligned}
&\int_0^\infty \frac{1}{\tau^n}\left(\frac{1}{2\pi}\right)^{\frac{n-1}{2}} \frac{1}{\Gamma\left(\frac{n+1}{2}\right)} \left(\frac{1}{2}\right)^{\frac{n+1}{2}} \kappa_s^n \exp\left[-\kappa_s\frac{\|\vec{\beta}\|}{\tau}\right] \frac{1}{\Gamma(\alpha)}\kappa_s^{\alpha-1}\exp(-\kappa_s)\mathrm{d}\kappa_s \\
&= \frac{1}{\tau^n}\left(\frac{1}{2\pi}\right)^{\frac{n-1}{2}} \frac{1}{\Gamma(\alpha)\Gamma\left(\frac{n+1}{2}\right)} \left(\frac{1}{2}\right)^{\frac{n+1}{2}} \int_0^\infty \kappa_s^{n+\alpha-1}\exp\left[-\kappa_s(1+\frac{\|\vec{\beta}\|}{\tau})\right]\mathrm{d}\kappa_s \\
&= \frac{1}{2^n\Gamma\left(\frac{n+1}{2}\right)\pi^{\frac{n-1}{2}}} \frac{\Gamma(\alpha+n)}{\Gamma(\alpha)} \frac{1}{\tau^n}\left(1+\frac{\|\vec{\beta}\|}{\tau}\right)^{-(\alpha+n)}.
\end{aligned}$$

A.9 Efficient Calculation of $(B_{[-s]}^{\mathrm{T}}B_{[-s]})^{-1}$ and Computational Complexity

The matrix inversion can be computationally intensive for large p. In order to avoid a direct matrix inversion at each Gibbs sampling step, we develop a fast computing method that can: (i) extract $(B_{[-s]}^{\mathrm{T}}B_{[-s]})^{-1}$ from $(B^{\mathrm{T}}B)^{-1}$, (ii) update $(B^{\mathrm{T}}B)^{-1}$ when there is a change in one column of B.

For (i), suppose that we have the value of $(B^{\mathrm{T}}B)^{-1}$, without loss of generality, let the matrix $B = [B_{[-s]}\ \vec{B}_s]$, using block matrix form:

$$(B^{\mathrm{T}}B)^{-1} = \begin{bmatrix} B_{[-s]}^{\mathrm{T}}B_{[-s]} & B_{[-s]}^{\mathrm{T}}\vec{B}_s \\ \vec{B}_s^{\mathrm{T}}B_{[-s]} & \vec{B}_s^{\mathrm{T}}\vec{B}_s \end{bmatrix}^{-1} = \begin{bmatrix} M_{1,1} & M_{1,2} \\ M_{1,2}^{\mathrm{T}} & M_{2,2}, \end{bmatrix}$$

where $M_{j,k}$ is corresponding block of $(B^{\mathrm{T}}B)^{-1}$. Using the block matrix inversion formula, we have:

$$(B_{[-s]}^{\mathrm{T}}B_{[-s]})^{-1} = M_{1,1} - M_{1,2}M_{2,2}^{-1}M_{1,2}^{\mathrm{T}}.$$

Since $M_{2,2}$ is a scalar, the above can be evaluated rapidly.

For (ii), supposing that we have updated $\vec{B}_s$ to $\vec{B}_s^*$, and denoting $B^* = [B_{[-s]}\ \vec{B}_s^*]$, we want to calculate $(B^{*\mathrm{T}}B^*)^{-1}$. Note that

$$(B^{*\mathrm{T}}B^*)^{-1} = \begin{bmatrix} B_{[-s]}^{\mathrm{T}}B_{[-s]} & B_{[-s]}^{\mathrm{T}}\vec{B}_s^* \\ \vec{B}_s^{*\mathrm{T}}B_{[-s]} & \vec{B}_s^{*\mathrm{T}}\vec{B}_s^* \end{bmatrix}^{-1} = \begin{bmatrix} M_{1,1}^* & M_{1,2}^* \\ M_{1,2}^{*\mathrm{T}} & M_{2,2}^* \end{bmatrix}.$$

We have:

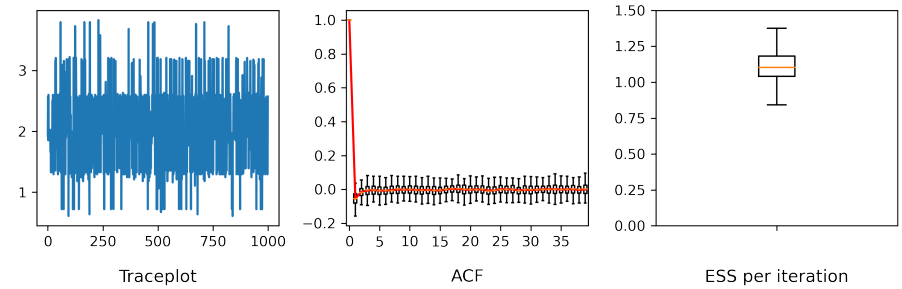
$$\begin{aligned}
M_{2,2}^* &= \{\vec{B}_s^{*\mathrm{T}}\vec{B}_s^* - \vec{B}_s^{*\mathrm{T}}B_{[-s]}(B_{[-s]}^{\mathrm{T}}B_{[-s]})^{-1}B_{[-s]}^{\mathrm{T}}\vec{B}_s^*\}^{-1} \\
&= (\vec{B}_s^{*\mathrm{T}}P_s\vec{B}_s^*)^{-1}, \\
M_{1,2}^* &= -(B_{[-s]}^{\mathrm{T}}B_{[-s]})^{-1}B_{[-s]}^{\mathrm{T}}\vec{B}_s^*M_{2,2}^*, \\
M_{1,1}^* &= (B_{[-s]}^{\mathrm{T}}B_{[-s]})^{-1} + (B_{[-s]}^{\mathrm{T}}B_{[-s]})^{-1}B_{[-s]}^{\mathrm{T}}\vec{B}_s^*M_{2,2}^*\vec{B}_s^{*\mathrm{T}}B_{[-s]}(B_{[-s]}^{\mathrm{T}}B_{[-s]})^{-1} \\
&= (B_{[-s]}^{\mathrm{T}}B_{[-s]})^{-1} + M_{1,2}^*M_{2,2}^{*-1}M_{1,2}^{*\mathrm{T}},
\end{aligned}$$

where $P_s = I - B_{[-s]}(B_{[-s]}^{\mathrm{T}}B_{[-s]})^{-1}B_{[-s]}^{\mathrm{T}}$, and we can use step (i) to compute $(B_{[-s]}^{\mathrm{T}}B_{[-s]})^{-1}$. Since $M_{2,2}$ and $M_{2,2}^*$ are scalars, all matrix inversions are avoided.

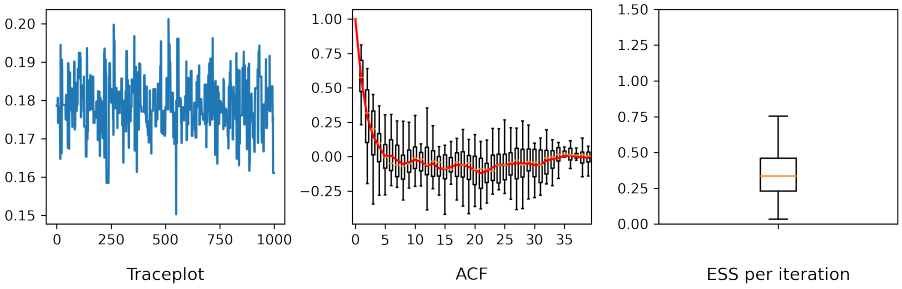
Therefore, throughout the posterior estimation, we only need to invert $B^{\mathrm{T}}B$ for one time to calculate the initial value. Examining the computational complexity, if using serial computation, the slowest matrix product/addition has a complexity of $O(p^2)$. Since most of the existing linear algebra toolboxes are optimized with some parallelization, we now check the parallel computing complexity for each term above. Computing $(B_{[-s]}^{\mathrm{T}}B_{[-s]})^{-1}$ involves a matrix subtraction and vector-scalar-vector product, which have complexity $O(1)$. Similarly, computing $M_{1,1}^*$ has complexity $O(1)$. Computing $M_{2,2}^*$ and $M_{1,2}^*$ involves matrix-vector products with complexity of $O(p)$. Lastly, when computing $\vec{\beta}_s$, we can bypass the matrix-matrix product by changing the order of multiplication to $\vec{\beta}_s = \{I - B_{[-s]}(B_{[-s]}^{\mathrm{T}}B_{[-s]})^{-1}B_{[-s]}^{\mathrm{T}}\}\vec{B}_s = \vec{B}_s - B_{[-s]}(B_{[-s]}^{\mathrm{T}}B_{[-s]})^{-1}(B_{[-s]}^{\mathrm{T}}\vec{B}_s)$; hence, it involves only matrix-vector product with a complexity of $O(p)$. As the result, overall the projection has a parallel computing complexity of $O(p)$.

A.10 Rapid Mixing of the Gibbs Sampler

The proposed Gibbs sampler exhibits apparent rapid mixing empirically. As shown in Figure 11 using the two-moon manifold simulation, the degrees of the tree change quickly from iteration to iteration, with the autocorrelation dropping to near zero almost within 1 lag; the update of the scale parameter τ using the random-walk Metropolis shows a fast drop to near zero within a lag of 10. We found similar performance in all of the experiments and data applications demonstrated in the article.



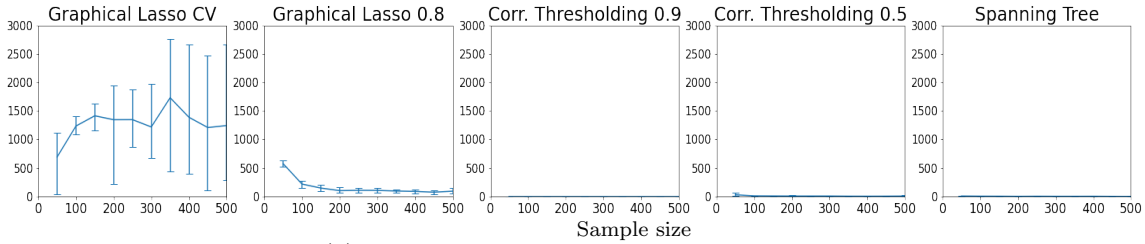
(a) Convergence diagnostics on the degrees of the spanning tree.



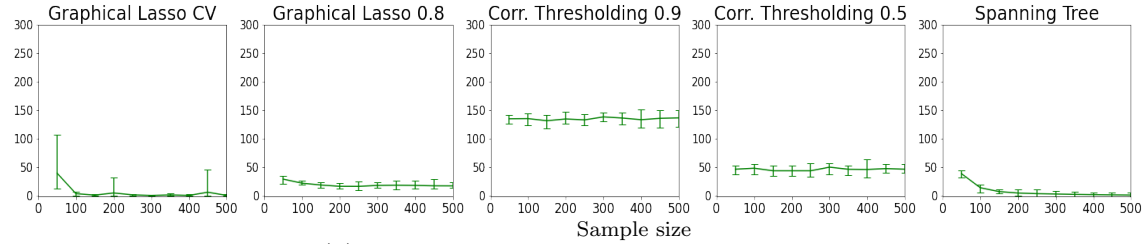
(b) Convergence diagnostics on the scale parameter τ.

Figure 11: The Gibbs sampler shows a rapid mixing of the Markov chains. Using the MCMC sample collected from the two-moon manifold experiments, we show the traceplot of the degree for one node / the scale parameter in one experiment, and autocorrelation plot computed based on 30 repeated experiments, and their effective sample sizes per iteration.

A.11 Additional Simulation Details when the Oracle is a Sparse Graph



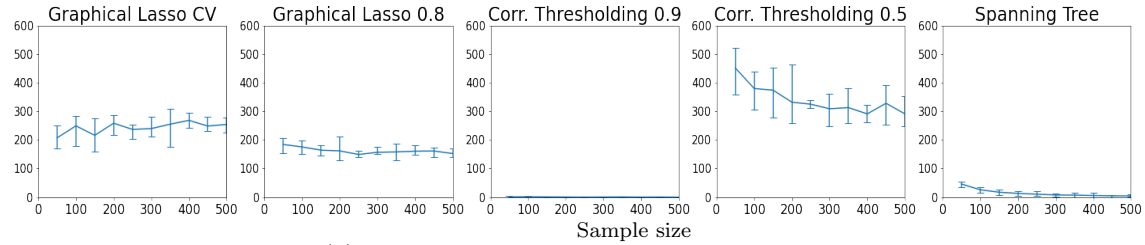
(a) False positive edges comparing with G_0.



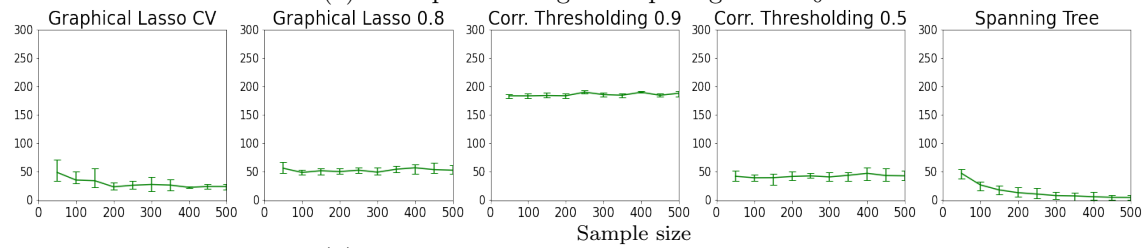
(b) False negative edges comparing with T_0.

Figure 12: The finite sample performance of graph estimation with the oracle being a sparse graph with about 600 edges over 200 nodes.

A.12 Additional Simulation Details when the Oracle is a Relatively Dense Graph



(a) False positive edges comparing with G_0.



(b) False negative edges comparing with T_0.

Figure 13: The finite sample performance of graph estimation with the oracle being a relatively dense graph with about $4,000$ edges over 200 nodes.

We choose to present $|T_0 \setminus \hat{G}| + |\hat{G} \setminus G_0|$ as error, because we know the false positive rates will always have $|\hat{G} \setminus G_0| \leq |\hat{G} \setminus T_0|$. Therefore, using $|\hat{G} \setminus T_0|$ would make the other estimators

showing even higher errors than the ones shown in the main text. On the other hand, for tree estimate, we can find out $|T_0 \setminus \hat{T}| = |\hat{T} \setminus T_0|$.

A.13 Sensitivity of Using Empirical Precision Matrix Thresholding for Graph Estimation

We use a simulation to empirically show the sensitivity in solely relying on comparing the magnitude of empirical precision matrix elements for graph estimation. We use a toy example, with $\Omega_0 = \begin{bmatrix} 4.75 & -1.64 & -2.66 \\ -1.64 & 2.72 & 0 \\ -2.66 & 0 & 2.75 \end{bmatrix}$, then we generate n Gaussian vectors from $\mathrm{N}(0, \Omega_0^{-1})$, and compute the empirical precision matrix $\hat{\Omega}$. As the ground-truth $\Omega_{0:2,3} = 0$, we record the event that $|\hat{\Omega}_{2,3}| > |\hat{\Omega}_{1,2}|$ or $|\hat{\Omega}_{1,3}|$, which would lead to an erroneous graph estimate if one uses magnitude thresholding (including soft-thresholding) on $\hat{\Omega}$. We repeat each experiment at each n for 100 times, and compute the error rate and quantify the error rate uncertainty. As shown in Figure 14, even at $n \approx p^2$), the error rate is still non-trivially large.

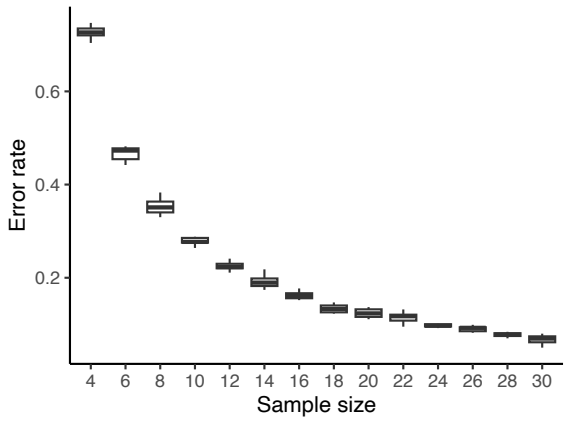
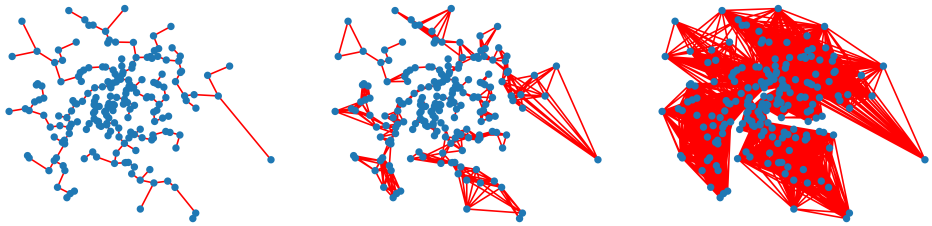


Figure 14: The error rate of using magnitude thresholding on empirical precision matrix for a simple graph estimation.

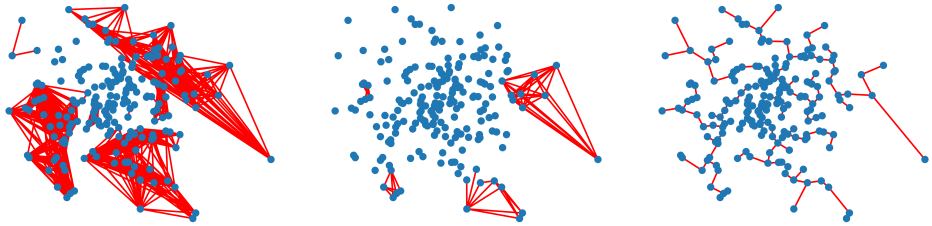
A.14 Graph Estimation When the Oracle is a Spanning Tree



(a) Oracle graph.

(b) Thresholding absolute correlation at 0.9.

(c) Thresholding absolute correlation at 0.5.



(d) Graphical lasso with α chosen by cross-validation.

(e) Graphical lasso using $\alpha = 20$.

(f) Bayesian spanning tree (posterior mode shown).

Figure 15: Simulated experiments of recovering a graph with $p = 200$ nodes, where the oracle graph is a spanning tree (Panel a). Panels (b-f) are plotted for $n = 50$. Starting around $n = 50$, the Bayesian spanning tree successfully recovers the ground truth with almost no errors, whereas the other approaches show many false positives.

We consider data generated from a spanning tree [Figure (15)(a)]. The thresholding estimator and graphical lasso show many false positives (Panels b-d). Empirically tuning the graphical lasso to $\alpha = 20$ does somewhat reduce false positives, however, it leads to edge loss and more false negatives (Panel e). The thresholding estimator has a similar sensitivity issue: thresholding at 0.5, as a common "default" choice in practice, leads to a severe overestimation of the graph edges, while 0.9 reduces this problem to some extent. On the other hand, coherent with the generative model, the Bayesian spanning tree shows good performance (Panel f).

References

David J Aldous. The Random Walk Construction of Uniform Spanning Trees and Uniform Labelled Trees. *SIAM Journal on Discrete Mathematics*, 3(4):450–465, 1990.

Artin Armagan, David B Dunson, and Jaeyong Lee. Generalized Double Pareto Shrinkage. *Statistica Sinica*, 23(1):119, 2013.

Francis Bach and Michael Jordan. Thin Junction Trees. In *Advances in Neural Information Processing Systems*, volume 14, 2001.

Ravindra B Bapat. *Graphs and Matrices*, volume 27. Springer, 2010.

Sumanta Basu and George Michailidis. Regularized Estimation in Sparse High-Dimensional Time Series Models. *The Annals of Statistics*, 43(4):1535–1567, 2015.

Peter J Bickel and Elizaveta Levina. Covariance Regularization by Thresholding. *The Annals of Statistics*, 36(6):2577–2604, 2008a.

Peter J Bickel and Elizaveta Levina. Regularized Estimation of Large Covariance Matrices. *The Annals of Statistics*, 36(1):199–227, 2008b.

Francis Buekenhout and Monique Parker. The Number of Nets of the Regular Convex Polytopes. *Discrete Mathematics*, 186(1-3):69–94, 1998.

V. V. Buldygin and Yu. V. Kozachenko. *Metric Characterization of Random Variables and Random Processes*. American Mathematical Society, 2000. ISBN 0821805339.

Simon Byrne and A Philip Dawid. Structural Markov Graph Laws for Bayesian Model Uncertainty. *The Annals of Statistics*, 43(4):1647–1681, 2015.

Diana Cai, Trevor Campbell, and Tamara Broderick. Edge-Exchangeable Graphs and Sparsity. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4249–4257, 2016a.

T Tony Cai, Weidong Liu, and Harrison H Zhou. Estimating Sparse Precision Matrix: Optimal Rates of Convergence and Adaptive Estimation. *The Annals of Statistics*, 44 (2):455–488, 2016b.

T Tony Cai, Zhao Ren, and Harrison H Zhou. Estimating Structured High-Dimensional Covariance and Precision Matrices: Optimal Rates and Adaptive Estimation. *Electronic Journal of Statistics*, 10(1):1–59, 2016c.

Xuan Cao, Kshitij Khare, and Malay Ghosh. Posterior Graph Selection and Estimation Consistency for High-Dimensional Bayesian DAG Models. *The Annals of Statistics*, 47 (1):319–348, 2019.

Seth Chaiken and Daniel J Kleitman. Matrix Tree Theorems. *Journal of Combinatorial Theory, Series A*, 24(3):377–381, 1978.

C Chow and Cong Liu. Approximating Discrete Probability Distributions With Dependence Trees. *IEEE transactions on Information Theory*, 14(3):462–467, 1968.

C Chow and T Wagner. Consistency of an Estimate of Tree-Dependent Probability Distributions. *IEEE Transactions on Information Theory*, 19(3):369–371, 1973.

David Roxbee Cox and Nanny Wermuth. *Multivariate Dependencies: Models, Analysis and Interpretation*, volume 67. CRC Press, 1996.

Hongsheng Dai. Perfect Sampling Methods for Random Forests. *Advances in Applied Probability*, 40(3):897–917, 2008.

Arthur P Dempster. Covariance Selection. *Biometrics*, 28(1):157–175, 1972.

Edsger W Dijkstra. A Note on Two Problems in Connexion With Graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

Adrian Dobra and Alex Lenkoski. Copula Gaussian Graphical Models and Their Application to Modeling Functional Disability Data. *The Annals of Applied Statistics*, 5(2A):969–993, 2011.

Adrian Dobra, Chris Hans, Beatrix Jones, Joseph R Nevins, Guang Yao, and Mike West. Sparse Graphical Models for Exploring Gene Expression Data. *Journal of Multivariate Analysis*, 90(1):196–212, 2004.

David L Donoho, Matan Gavish, and Iain M Johnstone. Optimal shrinkage of eigenvalues in the spiked covariance model. *The Annals of Statistics*, 46(4):1742, 2018.

Leo L Duan and Arkaprava Roy. Spectral Clustering, Bayesian Spanning Forest, and Forest Process. *Journal of the American Statistical Association*, in press:1–14, 2023.

David Durfee, Rasmus Kyng, John Peebles, Anup B Rao, and Sushant Sachdeva. Sampling Random Spanning Trees Faster Than Matrix Multiplication. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 730–742, 2017.

David Edwards, Gabriel CG De Abreu, and Rodrigo Labouriau. Selecting High-Dimensional Mixed Graphical Models Using Minimal AIC or BIC Forests. *BMC Bioinformatics*, 11 (1):1–13, 2010.

Gal Elidan and Stephen Gould. Learning Bounded Treewidth Bayesian Networks. In *Advances in Neural Information Processing Systems*, volume 21, 2008.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse Inverse Covariance Estimation With the Graphical Lasso. *Biostatistics*, 9(3):432–441, 2008.

Jerome H Friedman and Lawrence C Rafsky. Multivariate Generalizations of the Wald-Wolfowitz and Smirnov Two-Sample Tests. *The Annals of Statistics*, 7(4):697–717, 1979.

Edward I George and Robert E McCulloch. Stochastic Search Variable Selection. *Markov chain Monte Carlo in practice*, 68:203–214, 1995.

CJ Goh. *Duality in Optimization and Variational Inequalities*, volume 2. Taylor & Francis, 2002.

Peter J Green and Alun Thomas. Sampling Decomposable Graphs Using a Markov Chain on Junction Trees. *Biometrika*, 100(1):91–110, 2013.

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.

Søren Højsgaard, David Edwards, and Steffen Lauritzen. *Graphical Models With R*. Springer Science & Business Media, 2012.

Piotr Juszczak, David MJ Tax, Elżbieta Pe, and Robert PW Duin. Minimum Spanning Tree Based One-Class Classifier. *Neurocomputing*, 72(7-9):1859–1869, 2009.

David R Karger, Philip N Klein, and Robert E Tarjan. A Randomized Linear-Time Algorithm to Find Minimum Spanning Trees. *Journal of the ACM (JACM)*, 42(2):321–328, 1995.

Joseph B Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.

Richard A Levine and George Casella. Optimizing Random Scan Gibbs Samplers. *Journal of Multivariate Analysis*, 97(10):2071–2100, 2006.

Zhao Tang Luo, Huiyan Sang, and Bani Mallick. BAST: Bayesian Additive Regression Spanning Trees for Complex Constrained Domain. In *Advances in Neural Information Processing Systems*, volume 34, pages 90–102. Curran Associates, Inc., 2021.

Zhao Tang Luo, Huiyan Sang, and Bani Mallick. BAMDT: Bayesian Additive Semi-Multivariate Decision Trees for Nonparametric Regression. In *Proceedings of the 39th International Conference on Machine Learning*, pages 14509–14526. PMLR, June 2022. ISSN: 2640-3498.

Zhao Tang Luo, Huiyan Sang, and Bani Mallick. A Nonstationary Soft Partitioned Gaussian Process Model via Random Spanning Trees. *Journal of the American Statistical Association*, 0(0):1–12, 2023. ISSN 0162-1459.

Marina Meilă and Tommi Jaakkola. Tractable Bayesian Learning of Tree Belief Networks. *Statistics and Computing*, 16(1):77–92, 2006.

Marina Meilă and Michael I Jordan. Learning with Mixtures of Trees. *Journal of Machine Learning Research*, 1(Oct):1–48, 2000.

Mohamed Mosbah and Nasser Saheb. Non-Uniform Random Spanning Trees on Weighted Graphs. *Theoretical computer science*, 218(2):263–271, 1999.

Kazuo Murota. Discrete Convex Analysis. *Mathematical Programming*, 83(1-3):313–371, 1998.

Abhinav Natarajan, Willem van den Boom, Kristoforus Bryant Odang, and Maria de Iorio. On a Wider Class of Prior Distributions for Graphical Models. *Journal of Applied Probability*, pages 1–14, June 2023. ISSN 0021-9002, 1475-6072. Publisher: Cambridge University Press.

Nicholas G Polson and James G Scott. Shrink Globally, Act Locally: Sparse Bayesian Regularization and Prediction. *Bayesian Statistics*, 9(501-538):105, 2010.

Robert Clay Prim. Shortest Connection Networks and Some Generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957.

Pradeep Ravikumar, Martin J. Wainwright, Garvesh Raskutti, and Bin Yu. High-Dimensional Covariance Estimation by Minimizing L1-Penalized Log-Determinant Divergence. *Electronic Journal of Statistics*, 5:935–980, 2011. ISSN 19357524. doi: 10.1214/11-EJS631.

Alberto Roverato. Hyper Inverse Wishart Distribution for Non-Decomposable Graphs and Its Application to Bayesian Inference for Gaussian Graphical Models. *Scandinavian Journal of Statistics*, 29(3):391–411, 2002.

Philipp Rütimann and Peter Bühlmann. High Dimensional Sparse Covariance Estimation via Directed Acyclic Graphs. *Electronic Journal of Statistics*, 3:1133–1160, 2009.

Loïc Schwaller, Stéphane Robin, and Michael Stumpf. Closed-Form Bayesian Inference of Graphical Model Structures by Averaging Over Trees. *Journal de la Société Française de Statistique*, 160(2):1–23, 2019.

Kean Ming Tan, Daniela Witten, and Ali Shojaie. The Cluster Graphical Lasso for Improved Estimation of Gaussian Graphical Models. *Computational Statistics & Data Analysis*, 85: 23–36, 2015.

Vincent YF Tan, Animashree Anandkumar, Lang Tong, and Alan S Willsky. A Large-Deviation Analysis of the Maximum-Likelihood Learning of Markov Tree Structures. *IEEE Transactions on Information Theory*, 57(3):1714–1735, 2011.

Leonardo V. Teixeira, Renato M. Assunção, and Rosangela H. Loschi. Bayesian Space-Time Partitioning by Sampling and Pruning Spanning Trees. *Journal of Machine Learning Research*, 20(85):1–35, 2019.

Prejaas Tewarie, Edwin van Dellen, Arjan Hillebrand, and Cornelis J Stam. The Minimum Spanning Tree: An Unbiased Method for Brain Network Analysis. *Neuroimage*, 104: 177–188, 2015.

Hao Wang. Bayesian Graphical Lasso Models and Efficient Posterior Computation. *Bayesian Analysis*, 7(4):867–886, 2012.

Mike West. On Scale Mixtures of Normal Distributions. *Biometrika*, 74(3):646–648, September 1987.

Halbert White. Maximum Likelihood Estimation of Misspecified Models. *Econometrica: Journal of the Econometric Society*, 50(1):1–25, 1982.

David Bruce Wilson. Generating Random Spanning Trees More Quickly Than the Cover Time. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 296–303, 1996.

Xiaofan Xu and Malay Ghosh. Bayesian Variable Selection and Estimation for Group Lasso. *Bayesian Analysis*, 10(4):909–936, 2015.

Ming Yuan and Yi Lin. Model Selection and Estimation in the Gaussian Graphical Model. *Biometrika*, 94(1):19–35, 2007.