

RESEARCH ARTICLE

Regression-based Bayesian estimation and structure learning for nonparanormal graphical models

Jami J. Mulgrave¹ | Subhashis Ghosal

Department of Statistics, North Carolina State University, Raleigh, North Carolina

Correspondence

Jami Mulgrave, Department of Statistics, North Carolina State University, Raleigh, NC 27695, USA.

Email: jnj2102@gmail.com

Funding information

National Institutes of Health, Grant/Award Number: GM081057; National Science Foundation, Grant/Award Numbers: DGE-1252376, DMS-1510238, DMS-1732842

Abstract

A nonparanormal graphical model is a semiparametric generalization of a Gaussian graphical model for continuous variables in which it is assumed that the variables follow a Gaussian graphical model only after some unknown smooth monotone transformations. We consider a Bayesian approach to inference in a nonparanormal graphical model in which we put priors on the unknown transformations through a random series based on B-splines. We use a regression formulation to construct the likelihood through the Cholesky decomposition on the underlying precision matrix of the transformed variables and put shrinkage priors on the regression coefficients. We apply a plug-in variational Bayesian algorithm for learning the sparse precision matrix and compare the performance to a posterior Gibbs sampling scheme in a simulation study. We finally apply the proposed methods to a microarray dataset. The proposed methods have better performance as the dimension increases, and in particular, the variational Bayesian approach has the potential to speed up the estimation in the Bayesian nonparanormal graphical model without the Gaussianity assumption while retaining the information to construct the graph.

KEYWORDS

Bayesian inference, Cholesky decomposition, continuous shrinkage prior, nonparanormal graphical models

1 | INTRODUCTION

The Gaussian graphical model (GGM) is a mathematical model commonly used to describe conditional independence relationships among normally distributed random variables. The estimation of the underlying graph in a GGM is known as structure learning. Zeros in the inverse covariance matrix, or the precision matrix, indicate that the corresponding variables in the dataset are

conditionally independent given the rest of the variables in the dataset, and this relationship is represented by the absence of an edge in the graph. Similarly, nonzero entries in the precision matrix are represented by edges in the graph and correspond to conditionally dependent variables in the dataset. Thus, an assumed sparsity condition is used to learn the conditional dependence structure in a GGM. An extension of the GGM is the nonparanormal graphical model [26] in which the random

variables are replaced by transformed variables that are assumed to be normally distributed. Liu et al. [26] use a truncated empirical distribution function to estimate the transformation functions and then estimate the precision matrix of the transformed variables using the graphical lasso. A Bayesian method for the nonparanormal graphical model [36] uses a random series B-splines prior to estimate the transformation functions and a Student- t spike-and-slab prior to estimate the resulting precision matrix. These extensions differ from the Gaussian copula graphical model [14, 25, 33, 43] in that the nonparanormal graphical model concurrently estimates the transformation functions and the precision matrices. Nonparanormal graphical model approaches have been applied to discrete data models of interactions between genes [38] and to test differential gene networks [57].

Estimation of a sparse precision matrix is necessary to learn the structure in GGMs and nonparanormal graphical models. For unstructured precision matrices, a commonly used algorithm in the frequentist literature is the graphical lasso [16]. Many algorithms have been proposed to solve this problem including [2, 13, 16, 28, 31, 32, 45, 46, 54, 56].

Analogous methods in the Bayesian literature use priors to aid the edge selection procedure. For instance, off-diagonal entries of the precision matrix may be set to zero by allowing a point mass at zero in the prior [3], but the posterior is harder to compute or sample from. A normal spike-and-slab prior [51] replaces the point mass at zero by a highly concentrated normal distribution around zero and similarly, a Laplace spike-and-slab prior [17] has been used. From a computational point of view, continuous shrinkage priors such as the horseshoe prior [9], the Dirichlet–Laplace prior [6], and generalized double exponential prior [1], bring in the effects of both a point mass and a thick tail by a single continuous distribution with an infinite spike at zero.

Ideally, we seek solutions that guarantee a sparse positive definite matrix using continuous shrinkage priors. Since continuous shrinkage priors do not assign exact zeros, a variable selection procedure needs to be used to determine which of the small and nonzero elements should be specified as exactly zero. Methods that use spike and slab priors naturally incorporate variable selection, whereas methods that use alternative priors need a thresholding procedure. However, post-hoc thresholding procedures do not guarantee a positive definite precision matrix. The methods in [50, 51] guarantee a positive definite matrix by way of the sampling algorithm. [42, 50] use the double exponential prior and improve on its use for sparsity by allowing each double exponential prior to have its own shrinkage parameter. More recent methods estimate the inverse covariance matrix by

using the normal spike and slab prior [22, 23, 41, 51] for variable selection in the graphical model context. Lastly, Williams et al. [53] constructs Gaussian graphical models by estimating the partial correlation matrix using a horseshoe prior for regularization and for sparsity, using projection predictive selection, a method that allows for variable exclusion based on predictive utility, with good results.

The purpose of this paper is to explore the use of a Cholesky decomposition, horseshoe prior, and variational Bayesian (VB) techniques to construct a Bayesian nonparanormal graphical model. Utilizing a Cholesky decomposition is an alternative way to incorporate the positive definiteness constraint on precision matrices, but is very dependent on the ordering of the variables [44]. We consider a prior based on Cholesky decomposition of the precision matrix that reduces this dependence. We derive a sparsity constraint that ensures a weak order invariance in that it maintains the same order of sparsity in the rows of the precision matrix by increasing the order of sparsity down the rows of the lower triangular matrix. We construct a pseudo-likelihood through regression of each variable on the preceding ones. The approach splits the very high dimensional original problem to several lower dimensional ones. The method in [55] is also based on Cholesky decomposition, but it uses a noninformative Jeffreys' prior and the ordering issue of the Cholesky decomposition is not addressed.

We consider two different priors, the horseshoe and the Bernoulli–Gaussian [47]. These priors have clear interpretations of the probability of nonzero elements [47, 48], which allows us to effectively calibrate sparsity. The strength of the Bernoulli–Gaussian prior is that it leads to a sparse positive definite precision matrix that does not require thresholding and the strength of the horseshoe prior is that it is a better model of sparsity than the Bernoulli–Gaussian prior due to its heavier tails. Horseshoe priors have not yet been considered for Bayesian nonparanormal graphical models that use transformation functions. We compare the performance of the methods using both a VB algorithm and a full Markov chain Monte Carlo (MCMC) sampling scheme. Mean field variational Bayes [18, 49] is an alternative to MCMC that allows for faster fitting by deterministic optimization. A VB method for Gaussian graphical models is developed in [10] and an expectation conditional-maximization approach is used by Li and McCormick [23] in Gaussian copula graphical models. This approach has not yet been explored in the setting of a nonparanormal graphical model. We wish to determine if we can retain the information learned in a Bayesian nonparanormal graphical model while speeding up the estimation process using VB techniques.

The paper is organized as follows. In the next section, we describe the model and the sparsity constraint. In Section 3, we describe the VB algorithm. In Sections 4 and 5, we discuss particular priors and their corresponding Markov Chain Monte Carlo algorithms. In Section 6, we describe a thresholding procedure and in Section 7, we detail the tuning procedure. In Section 8, we present a simulation study. In Section 9, we describe a real data application.

2 | MODEL AND PRIORS

2.1 | Nonparanormal transformation

Definition 1. A random vector $\mathbf{X} = (X_1, \dots, X_p)'$ has a nonparanormal distribution if there exist smooth monotone functions $\{f_d : d = 1, \dots, p\}$ such that $\mathbf{Y} = \mathbf{f}(\mathbf{X}) \sim N_p(\boldsymbol{\mu}, \Sigma)$, a normal distribution with mean $\boldsymbol{\mu}$, covariance matrix Σ , and dimension p , and where $\mathbf{f}(\mathbf{X}) = (f_1(X_1), \dots, f_p(X_p))'$. In this case, we shall write $\mathbf{X} \sim \text{NPN}(\boldsymbol{\mu}, \Sigma, \mathbf{f})$.

We put prior distributions on the unknown transformation functions through a random series based on B-splines. In [36], we have described the prior distributions, including the motivation and support for the choices made, in greater detail. We briefly describe the prior in this section. We represent the transformation functions $\mathbf{f}(\mathbf{x}) = (f_1(x_1), \dots, f_p(x_p))'$ in a nonparanormal model $\mathbf{X} \sim \text{NPN}(\boldsymbol{\mu}, \Sigma, \mathbf{f})$ through a basis expansion

$$f_d(x_d) = \sum_{j=1}^J \theta_{dj} B_j(x_d), \quad (1)$$

where each θ_{dj} are coefficients, $B_j(\cdot)$ are the B-spline basis functions, $d = 1, \dots, p$, $j = 1, \dots, J$, and J is the number of B-spline basis functions used in the expansion. We assume that the precision matrix $\Omega = \Sigma^{-1}$ is sparse, in that, most of its off-diagonal entries are zero. However, the model is not identifiable, since location-scale changes in the transformation functions and the normal distributions can be canceled by each other. To resolve the issue, one possibility is to fix the mean-vector to zero and assume that the covariance matrix is a correlation matrix, but putting a prior on such a matrix maintaining sparsity of its inverse appears inconvenient. Therefore, we let the mean and the precision matrix be free parameters while putting restrictions on the transformations. We begin with a normal prior on each of the coefficients of the B-splines, $\boldsymbol{\theta}_d = (\theta_{d1}, \dots, \theta_{dJ})'$, that is set to be $\boldsymbol{\theta}_d \sim N_J(\boldsymbol{\zeta}, o^2 \mathbf{I})$, where o^2 is some positive constant, $\boldsymbol{\zeta}$ is some vector of constants, and \mathbf{I} is the identity

matrix, and impose a monotonicity restriction on them to make the transformation f_d monotone (see below for details). We impose the following two linear constraints on the coefficients through function values of the transformations: $0 = f_d(1/2) = \sum_{j=1}^J \theta_{dj} B_j(1/2)$ and $1 = f_d(3/4) - f_d(1/4) = \sum_{j=1}^J \theta_{dj} [B_j(3/4) - B_j(1/4)]$. The linear constraints can be written in matrix/vector form as $\mathbf{A}\boldsymbol{\theta}_d = \mathbf{c}$ for each $d = 1, \dots, p$. The linear nature of the constraints allows us to retain the joint normality of the coefficient vectors before the monotonicity restriction, and hence a truncated joint normal after the restriction is imposed.

By the properties of a B-spline basis function, if the B-spline coefficients, θ_{dj} are increasing in j , then f_j is an increasing function. We thus impose the monotonicity constraint on the coefficients, which is equivalent with the series of inequalities $\theta_{d2} - \theta_{d1} > 0, \dots, \theta_{dJ} - \theta_{dJ-1} > 0$. The monotonicity constraint can be expressed in matrix/vector form as $\mathbf{F}\boldsymbol{\theta}_d > \mathbf{0}$ for each $d = 1, \dots, p$. Thus, the prior on the coefficients before the truncation is imposed is given by $\boldsymbol{\theta}_d | \{\mathbf{A}\boldsymbol{\theta}_d = \mathbf{c}\} \sim N_J(\boldsymbol{\xi}, \Gamma)$, where the prior mean and variance are $\boldsymbol{\xi} = \boldsymbol{\zeta} + \mathbf{A}'(\mathbf{A}\mathbf{A}')^{-1}(\mathbf{c} - \mathbf{A}\boldsymbol{\zeta})$ and $\Gamma = o^2 [\mathbf{I} - \mathbf{A}'(\mathbf{A}\mathbf{A}')^{-1}\mathbf{A}]$. To ensure we have a Lebesgue density on \mathbb{R}^{J-2} , we work with a dimension-reduced coefficient vector by removing two coefficients and we denote this reduction with a bar over the vector and matrix.

The final prior on the coefficients is given by a truncated normal prior distribution $\bar{\boldsymbol{\theta}}_d | \{\mathbf{A}\boldsymbol{\theta}_d = \mathbf{c}\} \sim \text{TN}_{J-2}(\bar{\boldsymbol{\xi}}, \bar{\Gamma}, \mathcal{T})$, where $\bar{\boldsymbol{\theta}}_d$ is the dimension-reduced coefficient vector with the dimension-reduced mean vector $\bar{\boldsymbol{\xi}}$, dimension-reduced covariance matrix $\bar{\Gamma}$, restriction $\mathcal{T} = \{\bar{\boldsymbol{\theta}}_d : \bar{\mathbf{F}}\bar{\boldsymbol{\theta}}_d + \bar{\mathbf{g}} > \mathbf{0}\}$. Additionally, $\bar{\mathbf{F}}$ is the dimension-reduced matrix of the monotonicity constraints and $\bar{\mathbf{g}}$ is a dimension-reduced vector of the constant pertaining to the monotonicity constraints. We denote the truncated normal distribution as $\text{TN}_p(\boldsymbol{\mu}, \Sigma, \mathcal{T})$ with mean $\boldsymbol{\mu}$, covariance matrix Σ , restriction \mathcal{T} , and dimension p . Any choice of $\boldsymbol{\zeta}$ is acceptable, but we use $\zeta_j = \nu + \tau \Phi^{-1}\left(\frac{j-0.375}{J-0.75+1}\right)$, $j = 1, \dots, J$, where ν is a constant, τ is a positive constant, and Φ^{-1} is the inverse of the cumulative distribution function of the standard normal distribution. The idea is that by increasing the original components of the mean vector $\boldsymbol{\zeta}$, the truncation set \mathcal{T} in the final prior of the B-spline coefficients will have a substantial prior probability.

Finally, we put an improper uniform prior on the mean $p(\boldsymbol{\mu}) = \prod_{d=1}^p p_d(\boldsymbol{\mu}_d) \propto 1$. The resulting transformed variables, $\mathbf{Z}_d = \mathbf{Y}_d - \boldsymbol{\mu}_d$, which are assumed to be distributed as $N(\mathbf{0}, \Omega^{-1})$ and $\mathbf{Y}_d = \sum_{j=1}^J \theta_{dj} B_j(\mathbf{X}_d)$, $d = 1, \dots, p$, are used to estimate the precision matrix and learn the structure of the underlying graph.

2.2 | Cholesky decomposition reformulated as regression problems

We learn the structure of the precision matrix using a Cholesky decomposition. Denote the Cholesky decomposition of Ω as $\Omega = \mathbf{L}\mathbf{L}'$, where \mathbf{L} is a lower triangular matrix with elements l_{kd} . Define the coefficients $\beta_{kd} = -l_{kd}/l_{dd}$ and the precision as $\phi_d = 1/\sigma_d^2 = l_{dd}^2$, where $d = 1, \dots, p$. Then, as described in [55], the lower triangular entries of Ω , denoted as ω_{kd} , are given by

$$\omega_{kd} = \sum_{m=1}^d l_{km}l_{dm} = \sum_{m=1}^d \beta_{km}\beta_{dm}\phi_m, \text{ for } k \geq d.$$

Accordingly, the multivariate Gaussian model $\mathbf{Z} \sim \mathbf{N}(\mathbf{0}, \Sigma)$ is equivalent to the set of independent regression problems

$$\mathbf{Z}_d = \sum_{k>d} \beta_{kd}\mathbf{Z}_k + \varepsilon_d, \quad \varepsilon_d \sim \mathbf{N}(0, \sigma_d^2), \quad d = 1, \dots, p,$$

where β_{kd} are the regression coefficients for $k = d + 1, \dots, p$ and $d = 1, \dots, p$, and \mathbf{Z}_d and \mathbf{Z}_k are, respectively, the d th column and k th columns selected from matrix \mathbf{Z} . We use the notation $k > d$ to indicate that the columns are greater than the d th column.

We use a standard conjugate noninformative prior on the variances. We consider two different continuous shrinkage priors on the regression coefficients, the horseshoe prior and the Bernoulli–Gaussian prior. Using these priors, we enforce a sparsity constraint along the rows of the lower triangular matrix. The sparsity constraint is one in which the global sparsity parameter of the continuous shrinkage prior is scaled by \sqrt{k} , where $k > d$ and $d = 1, \dots, p$. Using this constraint, we expect that the precision matrix will be sparse through weak order invariance. The sparsity constraint is derived in the next section.

2.3 | Sparsity constraint

In order to ensure that the probability that an entry is nonzero (i.e., sparsity) remains roughly the same over different rows we cannot simply impose the same degree of sparsity on the rows of the Cholesky factor \mathbf{L} , but need to change it over rows appropriately. Denote the probability as $P(\cdot)$. To see how the Cholesky factor \mathbf{L} depends on the row index, we observe that

$$\begin{aligned} P(\omega_{kd} \neq 0) &= P\left(\sum_m l_{km}l_{dm} \neq 0\right) \\ &= P(l_{km}l_{dm} \neq 0 \text{ for some } m) \end{aligned}$$

$$\begin{aligned} &= 1 - P(l_{km}l_{dm} = 0 \text{ for all } m) \\ &= 1 - P\left(\bigcap_{m=1}^{\min(k,d)} \{l_{km}l_{dm} = 0\}\right) \\ &= 1 - \prod_{m=1}^{\min(k,d)} P(l_{km}l_{dm} = 0) \\ &= 1 - \prod_{m=1}^{\min(k,d)} \{1 - P(l_{km}l_{dm} \neq 0)\} \\ &= 1 - \prod_{m=1}^{\min(k,d)} \{1 - P(l_{km} \neq 0)P(l_{dm} \neq 0)\} \\ &= 1 - \{1 - P(l_{km} \neq 0)P(l_{dm} \neq 0)\}^{\min(k,d)}. \end{aligned}$$

Let $\rho_k = P(\text{nonzero entry in the } k\text{th row of } \mathbf{L})$. Then

$$P(\omega_{kd} \neq 0) = 1 - (1 - \rho_k\rho_d)^{\min(k,d)}.$$

If $k \sim d$, the expression is roughly $1 - (1 - \rho_k^2)^k$, which remains stable in k if $\rho_k = c_p/\sqrt{k}$, where c_p depends on p but not on k . Then, we obtain the probability of nonzero to be $1 - \exp(-c_p^2)$. Furthermore, choosing c_p to be small for $p \rightarrow \infty$ makes the probability small, which is essential in higher dimension. We choose $\rho_k = P(\text{nonzero in } k\text{th row}) = c/(p\sqrt{k})$, and tune the value of $c \in \{0.1, 1, 10\}$ to cover a range of three orders of magnitude, that is $10^{-1}, 10^0, 10^1$.

3 | VARIATIONAL BAYES ESTIMATION

We observe n independent samples, $\mathbf{X}_1, \dots, \mathbf{X}_n$, from the nonparanormal model $\text{NPN}(\boldsymbol{\mu}, \Omega^{-1}, \mathbf{f})$ with a sparse Ω . Based on these observations and the prior described in Section 2.1, we intend to compute the posterior distribution to make inferences about Ω and its structure, using the transformations \mathbf{f} . Ideally, we would want to construct a complete VB algorithm in which the B-spline coefficients, mean, and inverse covariance matrix are estimated all in one setting. However, for our problem, there is no closed form solution for the truncated multivariate normal distribution, and closed form solutions are needed for the mean field VB algorithms. Instead, we use an exact Hamiltonian Monte Carlo within Gibbs scheme to sample the B-spline coefficients and the mean. We obtain the Bayes estimate of the B-spline coefficients, $\hat{\boldsymbol{\theta}}_d = E(\boldsymbol{\theta}_d | \mathbf{X}_1, \dots, \mathbf{X}_n)$, and the Bayes estimate of the mean, $\hat{\boldsymbol{\mu}}_d = E(\boldsymbol{\mu}_d | \mathbf{X}_1, \dots, \mathbf{X}_n)$, where $E(\cdot | \mathbf{X}_1, \dots, \mathbf{X}_n)$ is the posterior mean operator. We then apply the VB method on the synthetic data obtained by transforming the original observations using the estimated transformations. Thus we estimate the transformed

variables using

$$Z_{id} = \sum_{j=1}^J \hat{\theta}_{jd} \mathbf{B}_j(X_{id}) - \hat{\mu}_d.$$

Ideally, instead of plugging in, one can obtain samples from the posterior distributions of the transformations and draw samples from the variational distributions of the precision matrix for each generated sample and accumulate them. However, even in moderately high dimension, such an approach is extremely computationally intensive. Since the posterior distributions of the transformations are consistent [36], they concentrate near the Bayes estimate. As the main goal is structure learning, the inability of the plug-in to assess the posterior variability of the transformations is not a highly deterring issue. Thus, although the proposed algorithm is not fully Bayesian, it utilizes the strength of the VB approach to identify conditional independence relations in a nonparanormal graphical model within a manageable time. While the variational inference generally underestimates the posterior variance [8], the quality of uncertainty quantification is affected, but that of estimation is hardly compromised. Moreover, since the goal of structure learning is to identify zero or nearly nonzero elements in the precision matrix, the main purpose is not affected at all. We illustrate the variational method on the Bernoulli–Gaussian prior, following the strategy described in [39]. Let the Bernoulli distribution be denoted as Ber and the inverse gamma distribution be denoted as IG(A, B) with shape parameter A and scale parameter B . We can describe the joint distribution by

$$\begin{aligned} \mathbf{Z}_d \mid \boldsymbol{\beta}_{k>d}, \boldsymbol{\sigma}, Y_{k>d} &\sim \mathbf{N}(\mathbf{Z}_{k>d} Y_{k>d} \boldsymbol{\beta}_{k>d}, \boldsymbol{\sigma}_d^2 \mathbf{I}), \\ \beta_{kd} &\sim \mathbf{N}(0, g^2) \\ v_{kd} &\sim \text{Ber}(\rho_{kd}^*), \quad \sigma_d^2 \sim \text{IG}(A, B), \end{aligned} \quad (2)$$

for $d = 1, \dots, p$, where $\boldsymbol{\beta}_{k>d} = (\beta_{d+1}, \dots, \beta_p)$ is the vector of regression coefficients, $\mathbf{Z}_{k>d}$ is the matrix of transformations, and $Y_{k>d}$ is a binary indicator matrix of 0s and 1s that is modeled by the Bernoulli distribution with elements v_{kd} . The hyperparameters g^2 , A , and B , are fixed, and $\rho_{kd}^* \in [0, 1]$ controls the sparsity. This variant of the spike-and-slab prior indirectly models sparsity on the regression coefficients by putting a binary indicator on the regression coefficients in the likelihood, instead of directly modeling sparsity on the regression coefficients. As such, if $v_{kd} = 0$ for the Bernoulli–Gaussian prior, then $\beta_{kd} \mid v_{kd} \sim \mathbf{N}(0, g^2)$, unlike in usual spike-and-slab priors in which β_{kd} would be exactly equal to 0. We select ρ_{kd}^* using a tuning procedure that incorporates the sparsity constraint and is discussed in Section 3.1.

The joint posterior distribution that we aim to compute is

$$\begin{aligned} p(\boldsymbol{\beta}, Y, \boldsymbol{\sigma}^2 \mid \mathbf{Z}) &\propto \left(\prod_{i=1}^n \prod_{d=1}^{p-1} p(Z_{id} \mid \mathbf{Z}_{i,k>d}, \boldsymbol{\beta}_{k>d}, Y_{k>d}, \boldsymbol{\sigma}_d^2) \right. \\ &\quad \left. \times p(\boldsymbol{\beta}_{k>d}) p(Y_{k>d}) p(\boldsymbol{\sigma}_d^2) p(Z_{ip} \mid \sigma_p^2) p(\sigma_p^2) \right). \end{aligned}$$

By plugging in the estimated transformed variables, we use a VB algorithm to compute the posterior distribution of the sparse precision matrix. Mean field VB inference involves minimizing the Kullback–Leibler divergence between the true posterior distribution and a factorized approximation of the posterior. Let $\boldsymbol{\kappa}$ represent the set of parameters in the model and \mathbf{Z} represent the matrix of estimated transformed variables. Then, $p(\boldsymbol{\kappa} \mid \mathbf{Z})$ is approximated by $q(\boldsymbol{\kappa}) = \prod_{k=1}^K q_k(\boldsymbol{\kappa}_k)$, where $(\boldsymbol{\kappa}_1, \dots, \boldsymbol{\kappa}_K)$ is a partition of $\boldsymbol{\kappa}$. The optimal q_k densities satisfy

$$q_k(\boldsymbol{\kappa}_k) \propto \exp \left[\mathbb{E}_{\setminus q_k(\boldsymbol{\kappa}_k)} \{ \log p(\mathbf{Z}, \boldsymbol{\kappa}) \} \right],$$

where $\mathbb{E}_{\setminus q_k(\boldsymbol{\kappa}_k)}$ is the expectation with respect to all densities except $q_k(\boldsymbol{\kappa}_k)$ [7]. The variational lower bound (VLB) for the marginal likelihood for \mathbf{Z} is then given by

$$\text{VLB}(q) = \mathbb{E}_q[\log\{p(\mathbf{Z}, \boldsymbol{\kappa})/q(\boldsymbol{\kappa})\}],$$

where \mathbb{E}_q is the expectation with respect to the density $q_k(\boldsymbol{\kappa}_k)$. Using the coordinate ascent method, optimizing each q_k while holding the others fixed will result in the algorithm converging to a local maximum of the lower bound.

Following [39], the choice of factorization that we use for the VB approximation is

$$q(\boldsymbol{\beta}, \mathbf{v}, \boldsymbol{\sigma}^2) = q(\boldsymbol{\sigma}_p^2) \prod_{d=1}^{p-1} q(\boldsymbol{\beta}_d) q(\boldsymbol{\sigma}_d^2) \prod_{k=d+1}^p q(v_{kd}),$$

with, for some choice of parameters

$$\begin{aligned} q^*(\boldsymbol{\beta}_d) &\sim \mathbf{N}(\boldsymbol{\alpha}_d, \boldsymbol{\Sigma}_d), \quad q^*(\boldsymbol{\sigma}_d^2) \sim \text{IG}\left(A + \frac{n}{2}, S_d\right), \\ q^*(v_{kd}) &\sim \text{Ber}(w_{kd}). \end{aligned}$$

The parameters are obtained by the VLB with respect to them by coordinate ascents, called variational updates, which we can derive as in [39]. Introduce the notations $\text{expit}(x) = \exp(x)/\{1 + \exp(x)\}$, and $\text{logit}(x) = \log(x/(1 - x))$, and let the symbol \circ denote the Hadamard product between two matrices. Then, we obtain

$$\boldsymbol{\Sigma}_d = [\tau_d (\mathbf{Z}'_{k>d} \mathbf{Z}_{k>d}) \circ \boldsymbol{\Omega}_d + g^{-2} \mathbf{I}]^{-1},$$

$$\begin{aligned}
\alpha_d &= \tau_d (\tau_d \mathbf{W}_d \mathbf{Z}'_{k>d} \mathbf{Z}_{k>d} \mathbf{W}_d + \mathbf{D}_d)^{-1} \mathbf{W}_d \mathbf{Z}'_{k>d} \mathbf{Z}_d, \\
s_d &= B + \frac{1}{2} \left[\|\mathbf{Z}_d\|^2 - 2\mathbf{Z}'_d \mathbf{Z}_{k>d} \mathbf{W}_d \alpha_d \right. \\
&\quad \left. + \text{tr} \left\{ (\mathbf{Z}'_{k>d} \mathbf{Z}_{k>d} \circ \Omega_d) (\alpha_d \alpha'_d + \Sigma_d) \right\} \right], \\
\eta_{kd} &= \text{logit}(\rho_{kd}^*) - \frac{\tau_d}{2} (\alpha_{kd}^2 + \Sigma_{k,k}) \|\mathbf{Z}_k\|^2 \\
&\quad + \tau_d [\alpha_{kd} \mathbf{Z}'_k \mathbf{Z}_d - \mathbf{Z}'_k \mathbf{Z}_{l>k} \mathbf{W}_{l>k} (\alpha_{l>k} \alpha_{kd} + \Sigma_{l>k,k})], \\
s_p &= B + \frac{1}{2} \|\mathbf{Z}_p\|^2, \quad w_{kd} = \text{expit}(\eta_{kd}), \\
\tau_d &= \frac{2A + n}{2s_d},
\end{aligned}$$

for $l = k + 1, \dots, p$, and $k = d + 1, \dots, p$. Note that we use the notation $l > k$ to indicate that the columns are greater than the k th column and $\#(k > d)$ means the number of columns k higher than d . In addition, $\mathbf{W}_d = \text{diag}(\mathbf{w}_{k>d})$ where $\mathbf{w}_{k>d} = (w_{d+1}, \dots, w_p)$, $\Omega_d = \mathbf{w}_d \mathbf{w}'_d + \mathbf{W}_d (\mathbf{I} - \mathbf{W}_d)$, and $\mathbf{D}_d = \tau_d (\mathbf{Z}'_{k>d} \mathbf{Z}_{k>d}) \circ \mathbf{W}_d \circ (\mathbf{I} - \mathbf{W}_d) + g^{-2} \mathbf{I}$.

Using these optimal q_k densities, the VLB simplifies to

$$\begin{aligned}
\text{VLB}(\mathbf{Z}; \rho) &= -\frac{pn}{2} \log(2\pi) + pA \log(B) \\
&\quad - p \log \Gamma(A) - \left(A + \frac{n}{2}\right) \log s_p \\
&\quad + p \log \Gamma\left(A + \frac{n}{2}\right) \\
&\quad + \sum_{d=1}^{p-1} \left\{ \frac{\#(k > d)}{2} - \frac{\#(k > d)}{2} \log(g^2) \right. \\
&\quad - \left(A + \frac{n}{2}\right) \log(s_d) + \frac{1}{2} \log |\Sigma_d| \\
&\quad - \frac{1}{2g^2} \text{tr}(\alpha_d \alpha'_d + \Sigma_d) + \sum_{k=(d+1)}^p \left[w_{kd} \log\left(\frac{\rho_{kd}^*}{w_{kd}}\right) \right. \\
&\quad \left. + (1 - w_{kd}) \log\left(\frac{1 - \rho_{kd}^*}{1 - w_{kd}}\right) \right] \Big\}. \tag{3}
\end{aligned}$$

The VB algorithm (Algorithm 1) is detailed in Appendix C.

3.1 | Tuning procedure

For every $(p - 1)$ regression problem, we choose the parameter ρ_{kd}^* , used in the prior in Equation (2), by applying the tuning algorithm described in detail in [39, sec. 4] because the authors also describe a way to select the hyperparameter ρ , the tuning parameter that they use to control sparsity. In this section, we describe the changes that we made to add the sparsity constraint to our tuning parameter. We use the value of ρ discussed in [39] and multiply that value with $\rho_k = c/(p\sqrt{k})$ to incorporate the sparsity constraint discussed in Section 2.3. Thus, for the fixed ρ that was discussed in [39], for our work, that translates to $\rho_k^* = \text{expit}(-0.5\sqrt{n})/(p\sqrt{k})$. Note that, since

the dimension d is not changing for ρ_k^* , we do not need to include c for tuning. For a fixed \mathbf{w} , which was discussed in [39], for our work, which translates to the fixed $\mathbf{w}_{k>d}$, and we select $\rho_{kd}^* = (\text{expit}(\iota_j c_j) / (p\sqrt{k}))$, where c_j is taken from an equally spaced grid of 50 points between 0.1 and 10, and ι_j varies over an equally spaced grid of 50 points between -15 and 5 . We replace the c with c_j which leads a grid of 50 values of c_j between 0.1 and 10 instead of the three values of $c \in \{0.1, 1, 10\}$ that was discussed in Section 2.3. The variational lower bound for the tuning procedure is only based on the preceding $(p - 1)$ regressions and not the regression relations that involve Z_p and σ_p^2 .

4 | MCMC ESTIMATION THROUGH THE HORSESHOE PRIOR

4.1 | Horseshoe prior

We use the horseshoe prior described in [37], to shrink the β coefficients

$$\begin{aligned}
\mathbf{Z}_d \mid (\mathbf{Z}_{k>d}, \boldsymbol{\beta}_{k>d}, \sigma_d^2) &\sim \text{N}(\mathbf{Z}_{k>d} \boldsymbol{\beta}_{k>d}, \sigma_d^2 \mathbf{I}), \\
\boldsymbol{\beta}_{kd} \mid (\lambda_d^2, b_{kd}, \sigma_d^2) &\stackrel{\text{ind}}{\sim} \text{N}\left(0, \frac{\sigma_d^2 b_{kd} c^2 \lambda_d^2}{p^2 k}\right), \\
\lambda_d^2 \mid a_d &\sim \text{IG}\left(\frac{1}{2}, \frac{1}{a_d}\right), \quad a_d \sim \text{IG}\left(\frac{1}{2}, 1\right), \\
b_{kd} \mid h_{kd} &\stackrel{\text{ind}}{\sim} \text{IG}\left(\frac{1}{2}, \frac{1}{h_{kd}}\right), \quad h_{kd} \sim \text{IG}\left(\frac{1}{2}, 1\right), \\
\sigma_d^2 &\sim \text{IG}(A, B), \tag{4}
\end{aligned}$$

for $d = 1, \dots, p$, where $\boldsymbol{\beta}_{k>d} = (\beta_{d+1}, \dots, \beta_p)$, $\mathbf{Z}_{k>d}$ is the matrix of transformations, and A and B are fixed hyperparameters.

The global scale parameter λ is roughly equivalent to the probability of a nonzero element [48]. We enforce the sparsity constraint using, $(\lambda_d c) / (p\sqrt{k})$. Thus, since we are working with the squared parameter, the factor in the variance term for β_{kd} is $(\lambda^2 c^2) / (p^2 k)$, where $c \in \{0.1, 1, 10\}$.

The joint posterior distribution and the corresponding conditional posterior distributions are provided in Appendix A, and the sampling algorithm (Algorithm 2) is provided in Appendix C.

5 | MCMC ESTIMATION THROUGH THE BERNOULLI-GAUSSIAN PRIOR

5.1 | Bernoulli-Gaussian prior

We use the same Bernoulli-Gaussian prior described in Equation (2). The joint posterior distribution and

the corresponding conditional posterior distributions are provided in Appendix B and the sampling algorithm (Algorithm 3) is provided in Appendix C.

6 | THRESHOLDING

Since the horseshoe prior is a continuous shrinkage prior, it does not assign exact zeros to the elements of the inverse covariance matrix, so we must apply a thresholding procedure that determines which of the elements should be exactly zero. The resulting thresholded matrices are then used to construct the graphical model. The thresholding procedure that we consider for the method using the horseshoe prior (Equation 4) is based on a 0–1 loss function described in [50] for classification under absolutely continuous priors. Although this procedure is heuristic, it seems to perform well in practice. Other thresholding rules may be used, such as those based on posterior credible intervals [19], information criterion [20], clustering [24], posterior model probabilities [3, 34], and projection predictive selection [53], but we choose to focus on the 0–1 loss procedure for this study.

6.1 | 0–1 Loss procedure

We find the posterior partial correlation using the precision matrices from the Gibbs sampler of the horseshoe prior (Equation 4) and the posterior partial correlation using the standard conjugate Wishart prior. The posterior samples of the partial correlation using the precision matrices from the Gibbs sampler are defined as

$$e_{kd,m} = \frac{-\omega_{kd,m}}{\sqrt{\omega_{kd,m}\omega_{dd,m}}},$$

where $\omega_{kd,m}$ is an MCMC sample from the posterior distribution of Ω_m , where $m = 1, \dots, M$, M is the number of MCMC samples, and $k, d = 1, \dots, p$. The posterior partial correlation using the standard conjugate Wishart prior is found by starting with the latent observation, \mathbf{Z}_m , which is obtained from the MCMC output. We put the standard Wishart prior on the precision matrix, $\Omega_m \sim W_p(3, \mathbf{I})$, which was used in [50] for their thresholding procedure, where \mathbf{I} is the identity matrix. Note that this Wishart prior does not assume sparsity, but \mathbf{Z} is obtained from the MCMC output assuming sparsity of the precision matrix. Through conjugacy, the posterior distribution is $\Omega_m \sim W_p(n+3, (\mathbf{I} + \mathbf{S}_m)^{-1})$, where $\mathbf{S}_m = \mathbf{Z}'_m \mathbf{Z}_m$. We then calculate the mean of the posterior distribution, $\mathbf{H}_m = \mathbb{E}(\Omega_m | \mathbf{Z}_m) = (n+3)(\mathbf{I} + \mathbf{S}_m)^{-1}$. Finally, we compute the posterior samples of partial correlation coefficients by

conjugate Wishart prior as

$$j_{kd,m} = \frac{-h_{kd,m}}{\sqrt{h_{kd,m}h_{dd,m}}},$$

where $h_{kd,m}$ stands for the (k, d) th element of \mathbf{H}_m .

We link these two posterior partial correlations for the 0–1 loss method. We claim the event $\{\omega_{kd,m} \neq 0\}$ if and only if

$$\frac{e_{kd,m}}{j_{kd,m}} > 0.5, \quad (5)$$

for $k, d = 1, \dots, p$ and $m = 1, \dots, M$. The idea is that we are comparing the regularized precision matrix from the horseshoe prior to the nonregularized precision matrix from the Wishart prior. If the absolute value of the partial correlation coefficient from the regularized precision matrix is similar in size or larger than the absolute value of the partial correlation coefficient from the Wishart precision matrix, then there should be an edge in the edge matrix. If the absolute value of the partial correlation coefficient from the regularized precision matrix is much smaller than the absolute value of the coefficient from the Wishart matrix, then the entry should not appear in the edge matrix.

7 | CHOICE OF PRIOR PARAMETERS

For the precision matrix being estimated with a horseshoe prior (Equation 4), we need to select the value of the parameter c which controls the sparsity. We solve a convex constrained optimization problem in order to use the Bayesian Information Criterion (BIC), as described in [11, 12]. First, we find the Bayes estimate of the inverse covariance matrix, $\hat{\Omega} = \mathbb{E}(\Omega | \mathbf{Z})$. We also find the average of the transformed variables, $\bar{\mathbf{Z}} = M^{-1} \sum_{m=1}^M \mathbf{Z}_m$, where \mathbf{Z}_m , $m = 1, \dots, M$, are obtained from the MCMC output. Then, using the sum of squares matrix $\mathbf{S} = \bar{\mathbf{Z}}' \bar{\mathbf{Z}}$, we solve for $\hat{\Omega}_{MLE}$, the maximum likelihood estimate of the inverse covariance matrix

$$\text{minimize } -n \log \det \Omega + \text{tr}(\Omega \mathbf{S}), \quad \text{subject to } C(\hat{\Omega}),$$

where C represents the constraint that all elements of $\hat{\Omega}$ at the locations of the zeros of the estimated edge matrix from the MCMC sampler are zero. The estimated edge matrix from the MCMC sampler will be described in more detail in Section 8. For computational simplicity, in the code, we represent this problem as an unconstrained optimization problem as described in [11, 12].

Lastly, we calculate $\text{BIC} = -2\ell(\hat{\Omega}_{\text{MLE}}) + k \log n$, where k is the sum of the number of diagonal elements and the number of edges in the estimated edge matrix, $\hat{\Omega}$, and $-\ell(\hat{\Omega}_{\text{MLE}}) = -n \log \det \hat{\Omega}_{\text{MLE}} + \text{tr}(\hat{\Omega}_{\text{MLE}} \mathbf{S})$. We select the value of c that results in the smallest BIC.

8 | SIMULATION RESULTS

We conduct a simulation study to assess the performance of the proposed methods using the horseshoe MCMC, indicated as Horseshoe, Bernoulli–Gaussian MCMC, indicated as Bernoulli–Gaussian, and VB algorithm, indicated as variational Bayes. We choose not to include the Bayesian method for the nonparanormal graphical model described in [36] because we want to compare only the Cholesky decomposition-based Bayesian methods to the empirical method for the nonparanormal graphical model [26] and to a Bayesian Gaussian copula graphical model [33] based method. We indicate the Bayesian Gaussian copula graphical model by the Bayesian Copula, in which the rank likelihood is used to transform the random variables with a uniform prior on the graph, a G-Wishart prior on the inverse correlation matrix, and estimation is used with the birth-death MCMC [34]. These competing methods all utilize a transformation of the data to learn the graphical structure.

We assess the performance of these methods by calculating sensitivity, specificity, and the Matthews correlation coefficient (MCC). We assess the effect of the transformation functions of our proposed methods by calculating the scaled L_1 -loss. These metrics are detailed in Section 8.1. In this section, we describe the data generation process used to conduct the simulation study.

The random variables, Y_1, \dots, Y_p , are simulated from a multivariate normal distribution such that $Y_{1i}, \dots, Y_{ip} \stackrel{\text{i.i.d.}}{\sim} N(\boldsymbol{\mu}, \Omega^{-1})$ for $i = 1, \dots, n$. The means $\boldsymbol{\mu}$ are selected from an equally spaced grid between 0 and 2 with length p . We consider nine different combinations of n, p , and sparsity for Ω :

- $p = 25, n = 25$, sparsity = 10% nonzero entries in the off-diagonals;
- $p = 50, n = 100$, sparsity = 5% nonzero entries in the off-diagonals;
- $p = 100, n = 300$, sparsity = 2% nonzero entries in the off-diagonals;
- $p = 25, n = 25$, AR(2) model, sparsity $\approx 16\%$;
- $p = 50, n = 100$, AR(2) model, sparsity $\approx 8\%$;
- $p = 100, n = 300$, AR(2) model, sparsity $\approx 4\%$;
- $p = 25, n = 25$, circle model, sparsity = 8%;

- $p = 50, n = 100$, circle model, sparsity = 4%;
- $p = 100, n = 300$, circle model, sparsity = 2%;

where the circle model and the AR(2) model are described by the relations

- Circle model: $\omega_{ii} = 2, \omega_{i,i-1} = \omega_{i-1,i} = 1$, and $\omega_{1,p} = \omega_{p,1} = 0.9$;
- AR(2) model: $\omega_{i,i} = 1, \omega_{i,i-1} = \omega_{i-1,i} = 0.5$ and $\omega_{i,i-2} = \omega_{i-2,i} = 0.25$.

The percent sparsity levels for Ω are computed using lower triangular matrices that have diagonal entries normally distributed with $\mu_{\text{diag}} = 1$ and $\sigma_{\text{diag}} = 0.1$, and nonzero off-diagonal entries normally distributed with $\mu_{\setminus \text{diag}} = 0$ and $\sigma_{\setminus \text{diag}} = 1$, where \setminus denotes the complement of the set.

The observed variables $\mathbf{X} = (X_1, \dots, X_p)$ are constructed from the simulated variables Y_1, \dots, Y_p . The functions used to construct the observed variables are three cumulative distribution functions (c.d.f.s): asymmetric Laplace, extreme value, and stable. Any values of the parameters for the c.d.f.s could be chosen, but instead of selecting 25, 50, and 100 sets of parameters, we automatically choose the values of the parameters. The values are the maximum likelihood estimates of the corresponding distributions (asymmetric Laplace, extreme value, and stable) using the variables Y_1, \dots, Y_p , calculated with the mle function in MATLAB.

We follow the procedure in [36] to estimate the transformation functions. The hyperparameters for the normal prior are chosen to be $\nu = 1, \tau = 1$, and $\sigma^2 = 1$. To choose the number of basis functions, we use the Akaike Information Criterion as described in [36]. Samples from the truncated multivariate normal posterior distributions for the B-spline coefficients are obtained using the exact Hamiltonian Monte Carlo (exact HMC) algorithm [40]. The initial coefficient values, $\theta_{dj, \text{initial}}$, for the exact HMC algorithm are calculated using quadratic programming as described in [36]. After finding the initial coefficient values θ_d , we construct initial values for $Y_{d, \text{initial}} = \sum_{j=1}^J \theta_{dj, \text{initial}} B_j(X_d)$ using the observed variables. These initial values $\mathbf{Y}_{\text{initial}}$ are used to find the initial values for $\Sigma, \boldsymbol{\mu}$, and Ω for the algorithm, where $\Sigma_{\text{initial}} = \text{cov}(\mathbf{Y}_{\text{initial}})$, $\boldsymbol{\mu}_{\text{initial}} = \bar{\mathbf{Y}}_{\text{initial}}$, where $\bar{\mathbf{Y}}_{\text{initial}}$ is the average of $\mathbf{Y}_{\text{initial}}$, and $\Omega_{\text{initial}} = \Sigma_{\text{initial}}^{-1}$.

For the part of the simulation study in which we do not estimate the transformation functions, the initial values for the Horseshoe, Bernoulli–Gaussian, and variational Bayes algorithms are constructed from the observed variables, \mathbf{X} , with $\Sigma_{\text{initial}} = \text{cov}(\mathbf{X})$, $\boldsymbol{\mu}_{\text{initial}} = \bar{\mathbf{X}}$, where $\bar{\mathbf{X}}$ is the average of \mathbf{X} , and $\Omega_{\text{initial}} = \Sigma_{\text{initial}}^{-1}$. Afterward, the

mean μ and the precision matrix Ω are estimated using the algorithms as described in the previous sections.

The hyperparameter g^2 for the Bernoulli–Gaussian prior and the variational Bayes algorithm is fixed at 10. The hyperparameters A and B for the inverse gamma distribution for the Bernoulli–Gaussian prior, the variational Bayes algorithm, and the horseshoe prior, are fixed at $A = B = 0.01$. The initial value, τ^0 , where $t = 0$, for the variational Bayes algorithm is chosen to be 1000. The threshold ε for stopping the variational Bayes algorithm is set to $\varepsilon = 10^{-6}$. For the variational Bayes algorithm and the MCMC algorithm using the Bernoulli–Gaussian prior, the tuning procedure described in Section 3.1 is used to find the hyperparameter for the Bernoulli distribution, ρ_{kd}^* . Since the vector $\mathbf{w}_{k>d}$ from the tuning procedure consists of only 0 and 1 values, it is used as the initial indicator vector \mathbf{v}_d for the MCMC algorithm using the Bernoulli–Gaussian prior. The data matrix that is used as input for the tuning procedure is $\mathbf{Z}_{\text{initial}} = \mathbf{Y}_{\text{initial}} - \boldsymbol{\mu}_{\text{initial}}$, which was described in the previous paragraphs.

For the MCMC algorithm for the horseshoe prior, we consider three values of c that are a range of three orders of magnitude: $c \in \{0.1, 1, 10\}$. The value of c that yields the lowest BIC was selected for the final estimates of the precision matrix and edge matrix. The 0–1 loss procedure described in Section 6.1 was used to threshold the precision matrices and construct the edge matrices.

For the simulation study, we run 100 replications for each of the nine combinations and assess structure learning for each replication. We collect 10,000 MCMC samples for inference after discarding a burn-in of 5000. We do not apply thinning. The Bayesian copula method is implemented by the R package, `BDGraph` [35] using the option “`gcgm`.” Posterior graph selection is done using Bayesian model averaging, the default option in the `BDGraph` package, in which it selects the graph with links for which their estimated posterior probabilities are greater than 0.5. The nonparanormal graphical model is implemented by the R package `huge` [58] using the option “`truncation`.” The graphical lasso method is selected for the graph estimation and the default screening method, `lossless` [30, 54], is used. Three regularization selection methods are used to find the estimated precision matrix and select the nonparanormal graphical model: the Stability Approach for Regularization Selection (StARS) [27], the modified Rotation Information Criterion (RIC) [29], and the Extended Bayesian Information Criterion (EBIC) [15]. The default parameters in the `huge` package are used for each selection method. As in Liu et al. [26], the number of regularization parameters used is 50 and they are selected among an evenly spaced grid in the interval [0.16, 1.2].

The code for the proposed Bayesian methods is written in MATLAB and sparse representations of the

matrices are used when appropriate. For the variational Bayes algorithm, when calculating $w_{kd}^* = \text{expit}(\eta_{kd})$, it is set to 0 if $\exp(\eta_{kd})$ is below 2^{-52} , which is `eps`, the floating-point relative accuracy in MATLAB, while w_{kd}^* is set to 1 if $\exp(\eta_{kd})$ is equal to `infinity` in MATLAB for numerical stability. Infinity results from operations that lead to results too large to represent as conventional floating-point values. Similar adjustments are also applied for the Bernoulli–Gaussian MCMC. The code is given in Appendix C.

8.1 | Performance assessment

We compute the Bayes estimate of the precision matrix $\hat{\Omega} = \mathbb{E}(\Omega|\mathbf{Z})$ by averaging all MCMC samples after burn-in, or the variational Bayes estimate by averaging over 500 independent samples from the variational distribution. The median probability model [4] is used to obtain the Bayes estimate of the edge matrix. We find the estimated edge matrix by first using the 0–1 loss procedure discussed in Section 6.1 to threshold the MCMC precision matrix samples, and then we take the mean of the thresholded precision matrices. If each off-diagonal element of the mean of the thresholded matrices is greater than 0.5, the element is registered as an edge in the estimated edge matrix, and if each off-diagonal element of the mean is not greater than 0.5, it is registered as no edge. We use 0.5 as the cut-off since an average above 0.5 means on average, the matrices included an edge more than half of the time.

We compute specificity (SP), sensitivity (SE), and MCC to assess the performance of the graphical structure learning. They are defined as follows:

$$\begin{aligned} \text{Specificity} &= \frac{\text{TN}}{\text{TN} + \text{FP}}, & \text{Sensitivity} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{MCC} &= \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}, \end{aligned}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. For all three metrics, the higher the values are, the better is the classification. If there are models that are estimated to have no edges, they result in NaNs as MCC values.

We also look at the effect of the transformation functions on parameter estimation for our methods. We consider the scaled L_1 -loss function, the average absolute distance, as a measure of parameter estimation. Scaled L_1 -loss is defined as

$$\text{Scaled } L_1 - \text{loss} = \frac{1}{p^2} \sum_k \sum_d \left\| \hat{\Omega}_{kd} - \Omega_{\text{true},kd} \right\|$$

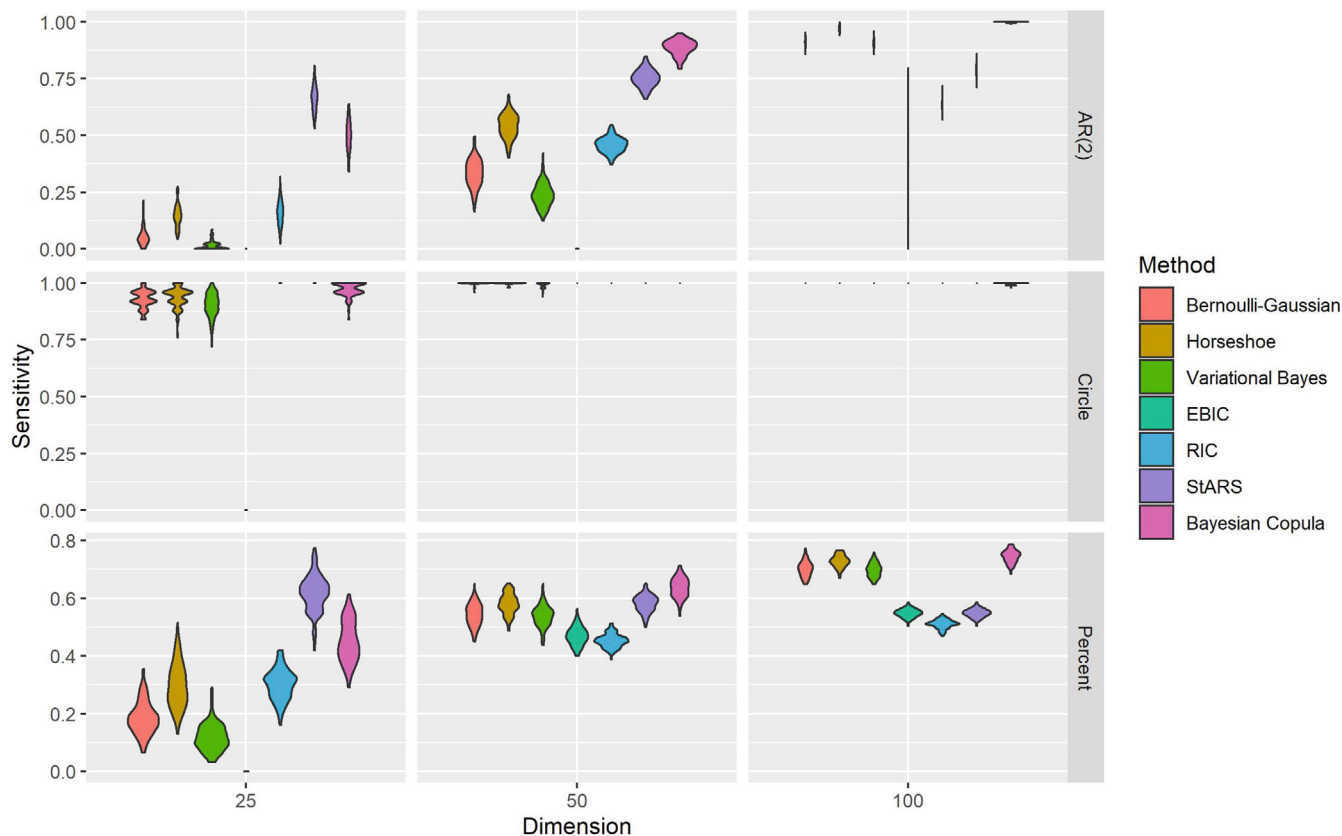


FIGURE 1 Violin plots of the sensitivity results for each of the methods for different structures of precision matrices. Percent refers to the 10% model for dimension $p = 25$, 5% model for dimension $p = 50$, and 2% model for dimension $p = 100$

where $\Omega_{\text{true},kd}$ stands for the true covariance matrix. Note that for the Bayesian Copula method, we use the estimated inverse correlation matrix and the true correlation matrix in place of the precision matrix for loss calculation.

We review the results of sensitivity, specificity, MCC, and the scaled L_1 -loss for each method using violin plots. In general, for sensitivity, specificity, and MCC, the closer the violin plots are to one and the tighter the violin plots, the better the performance of the method. For the scaled L_1 -loss, the closer the violin plots are to zero and the tighter the violin plots, the better performance.

First, we consider sensitivity. In Figure 1 for the $p = 25$ dimension and AR(2) model, the StARS model has the best sensitivity, followed with the Bayesian Copula model. For $p = 50$ and the AR(2) model, the Bayesian Copula performs the best, followed by the StARS model. Notably, the proposed methods perform better at the $p = 50$ dimension than at the $p = 25$ dimension, with the Horseshoe method performing the third best. Finally, for the $p = 100$ dimension and AR(2) model, the Bayesian Copula method performs the best and the proposed methods perform second best, with the Horseshoe method performing the best and the variational Bayes and Bernoulli–Gaussian methods performing third and fourth best. The Bayesian Copula

is the best, the Horseshoe is the second best, and the Bernoulli–Gaussian and variational Bayes methods are the third and fourth best, respectively. For the $p = 25$ dimension and the circle model, all methods are high performing, but the RIC and StARS methods perform the best and the Bayesian Copula method is the third best. For the $p = 50$ and $p = 100$ dimensions and the circle model, all methods perform similarly. For the $p = 25$ dimension and the 10% model, the StARS method is the best and the Bayesian Copula method is the second best. For the $p = 50$ dimension and 5% model, the Bayesian Copula method performs the best. The Horseshoe and StARS methods perform similarly and are the second best, while the variational Bayes and Bernoulli–Gaussian methods perform similarly and are the third best. For the $p = 100$ dimension and the 2% model, the Bayesian Copula slightly outperforms the Horseshoe model, and the Bernoulli–Gaussian and variational Bayes methods perform similarly at third best.

Next, we review how the methods perform when considering specificity. In Figure 2 for all dimensions and AR(2) model, the three proposed methods, Bernoulli–Gaussian, Horseshoe, and variational Bayes methods, as well as the EBIC method, perform the best.

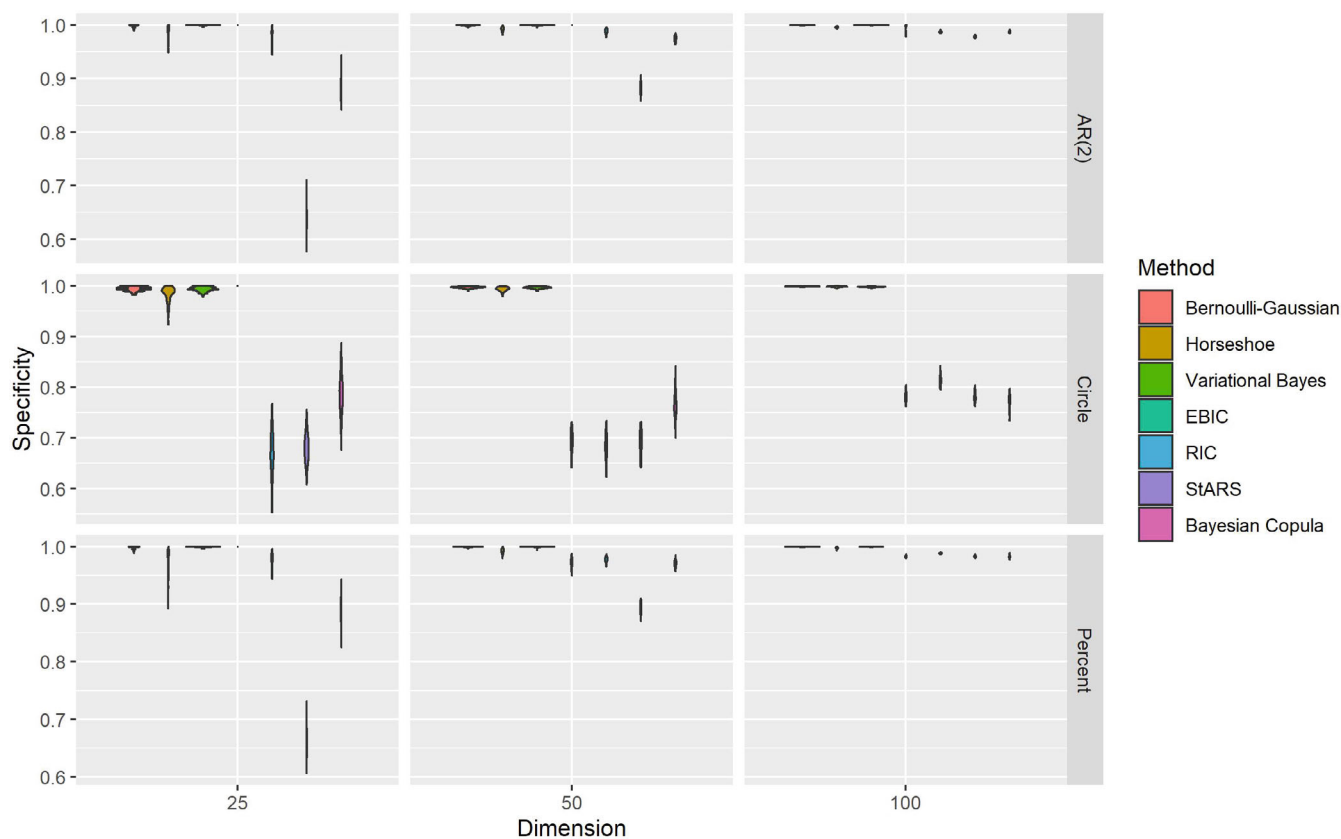


FIGURE 2 Violin plots of the specificity results for each of the methods for different structures of precision matrices. Percent refers to the 10% model for dimension $p = 25$, 5% model for dimension $p = 50$, and 2% model for dimension $p = 100$

For the $p = 25$ dimension and circle model, the three proposed methods, Bernoulli–Gaussian, Horseshoe, and variational Bayes methods, as well as the EBIC method, perform the best. For the $p = 50$ and $p = 100$ dimensions and the circle model, the three proposed methods, Bernoulli–Gaussian, Horseshoe, and variational Bayes methods, perform the best, outperforming all other methods. For the $p = 25$ dimension and the 10% model, the three proposed methods, Bernoulli–Gaussian, Horseshoe, and variational Bayes methods, as well as the EBIC method, perform the best. For the $p = 50$ dimension and 5% model and the $p = 100$ dimension and 2% model, the three proposed methods, Bernoulli–Gaussian, Horseshoe, and variational Bayes methods, perform the best.

We consider the MCC to compare the overall performance of structure learning. In Figure 3 for the $p = 25$ and $p = 50$ dimensions and the AR(2) model, the Bayesian Copula method performs the best and the Horseshoe method performs the second best. No edges were selected by the nonparanormal model using EBIC for the sparsity models of dimension $p = 25$ and for the $p = 50$ AR(2) model. For the $p = 100$ dimension and the AR(2) model, the three proposed methods, Horseshoe, Bernoulli–Gaussian, and variational Bayes

methods, perform the best. For all dimensions of the circle model, the three proposed methods, Horseshoe, Bernoulli–Gaussian, and variational Bayes methods, perform the best. Lastly, for the $p = 25$ dimension and 10% model, the Horseshoe method performs the best, and the Bernoulli–Gaussian and RIC methods perform similarly and are the second best. For the $p = 50$ and 5% model and $p = 100$ and 2% model, the three proposed methods, Horseshoe, Bernoulli–Gaussian, and variational Bayes methods, perform the best. Thus, when considering the overall structure learning, the proposed methods outperform all competing methods except in the cases of $p = 25$ and $p = 50$ and the AR(2) model.

Finally, in Figure 4, we review the results of parameter estimation, using the scaled L_1 -loss, for the three proposed methods. We consider whether or not the transformation decreases the scaled L_1 -loss. For all three methods, the transformation functions resulted in a smaller scaled L_1 -loss, implying an improvement in parameter estimation. Overall, the Horseshoe method had a higher scaled L_1 -loss than the Bernoulli–Gaussian and variational Bayes methods. In addition, overall, the variational Bayes method had a similar or lower scaled L_1 -loss compared to the Bernoulli–Gaussian method.

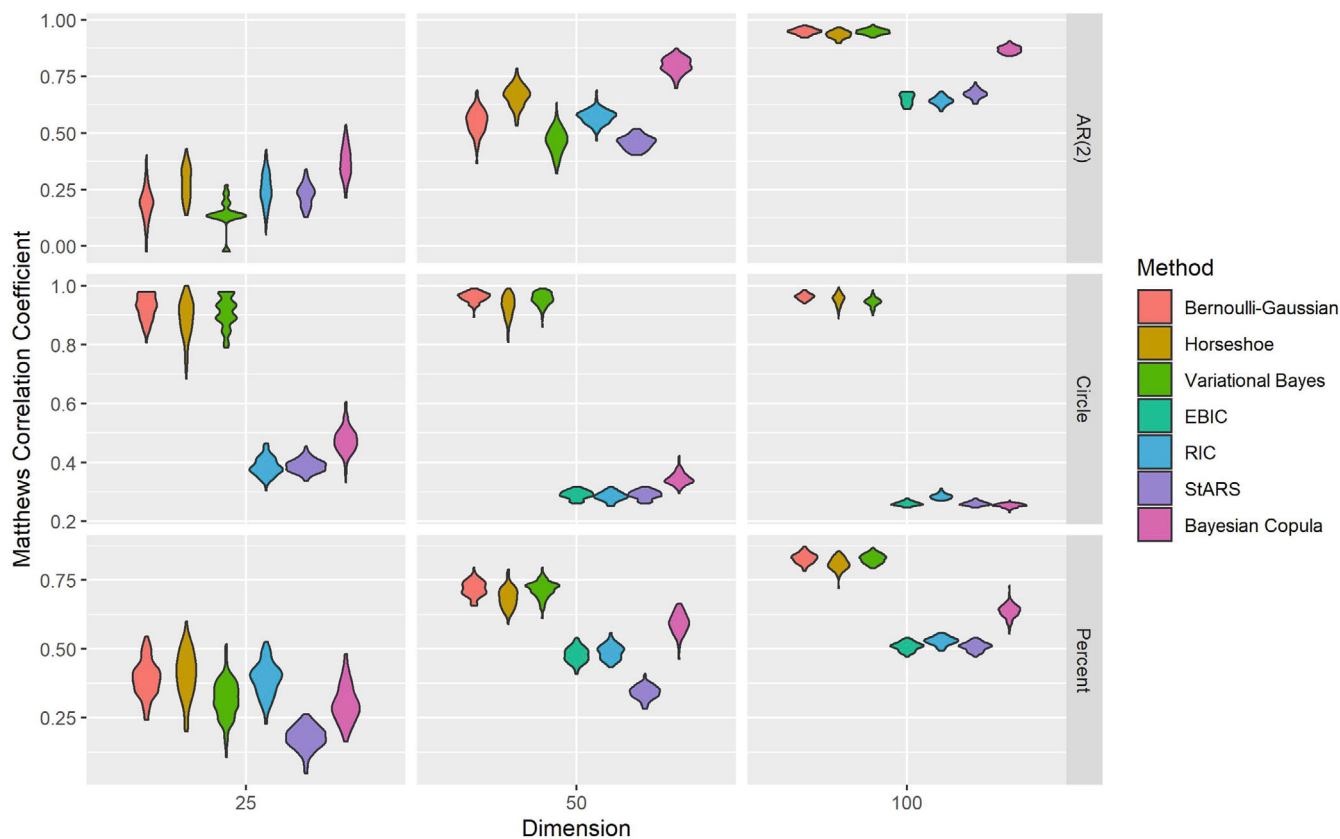


FIGURE 3 Violin plots of the Matthews correlation coefficient results for each of the methods for different structures of precision matrices. Percent refers to the 10% model for the dimension $p = 25$, 5% model for the dimension $p = 50$, and 2% model for the dimension $p = 100$

Figures 1–4 display the results. The first three violin plots in the figures are the three proposed methods, Bernoulli–Gaussian, Horseshoe, and variational Bayes, respectively. Note that Percent refers to the 10% model for dimension $p = 25$, 5% model for dimension $p = 50$, and 2% model for dimension $p = 100$.

9 | REAL DATA APPLICATION

For the real data application, we consider the dataset based on the GeneChip (Affymetrix) microarrays for the plant *Arabidopsis thaliana* originally referenced in Wille et al. [52]. This dataset features gene expression levels from isoprenoids. Isoprenoids serve a great many biochemical functions in plants, such as components of membranes (sterols) and photosynthetic pigments (carotenoids and chlorophylls). The cytosolic pathway, often described as the mevalonate or MVA pathway, is responsible for the synthesis of sterols and the plastidial (nonmevalonate or MEP) pathway is used for the synthesis of isoprenes, carotenoids and the side chains of chlorophyll. Although both pathways operate independently, interaction between them has

been discovered [21]. There are $n = 118$ microarrays and $p = 39$ genes from the isoprenoid pathway that are used. For pre-processing, the expression levels for each gene, x_i for $i = 1, \dots, 118$, are log-transformed. We study the associations among the genes using the Bayesian nonparanormal methods, the nonparanormal method of Liu et al. [26], and the method based on the Bayesian copula graphical model of [33]. These data are treated as multivariate Gaussian originally in Wille et al. [52].

Using the same set-up as in the simulation study, we fit the Bayesian copula graphical model using the BDGraph package and we fit the nonparanormal graphical model using the huge package. The BDGraph package selected 211 edges using Bayesian model averaging. The huge package using the RIC selection resulted in 140 edges and using the StARS method resulted in 209 edges. The EBIC-selected model results in no edges.

In order to construct the graphical models using our methods which use B-spline transformations, we converted the observations to be between 0 and 1 using the equation $(x - \min(x_i)) / (\max(x_i) - \min(x_i))$. The variational Bayes method results in 98 edges, the horseshoe prior based method results in 257 edges, and the

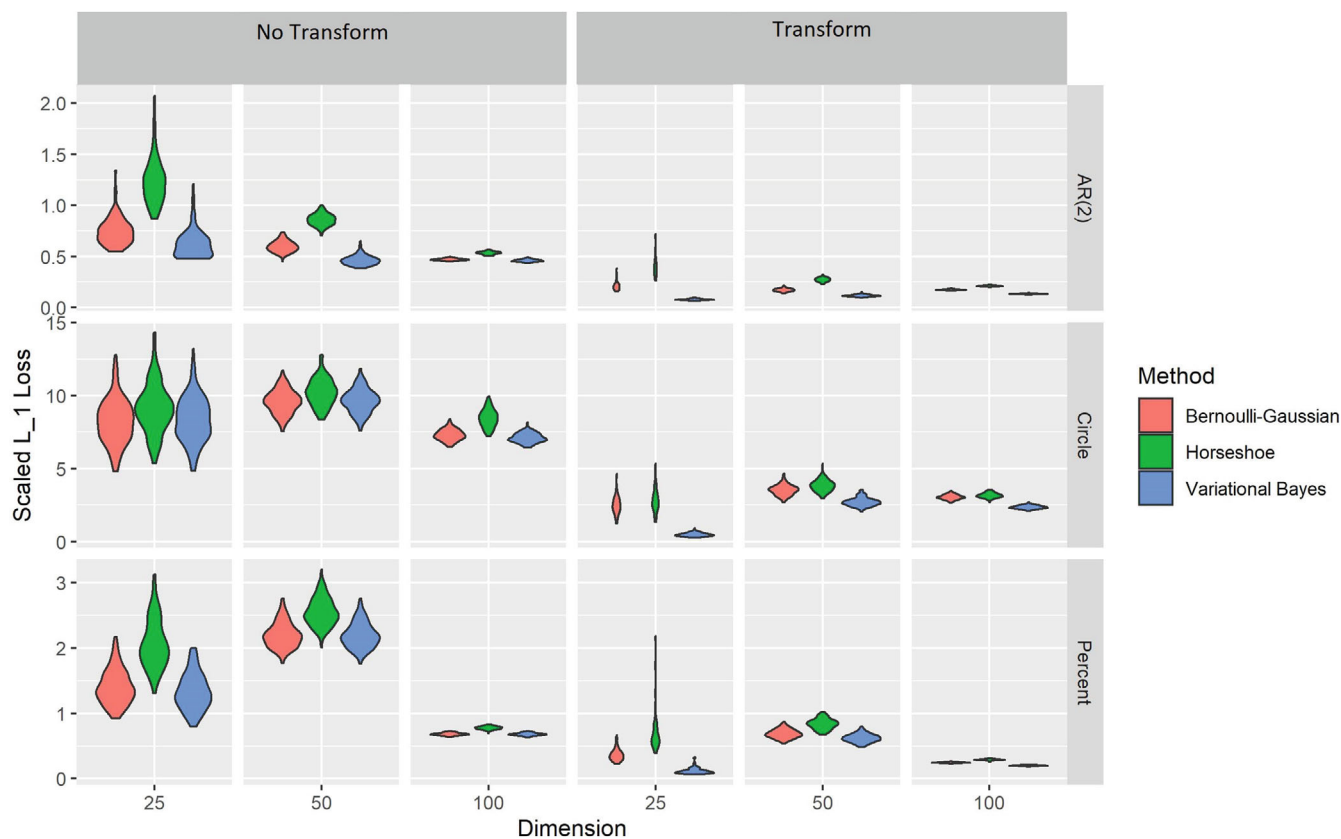


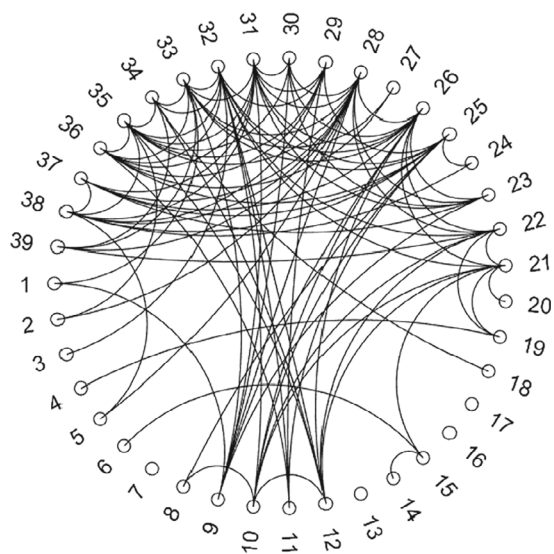
FIGURE 4 Violin plots of the scaled L_1 -loss, with and without transformation for different structures of precision matrices. Percent refers to the 10% model for the dimension $p = 25$, 5% model for the dimension $p = 50$, and 2% model for the dimension $p = 100$

Bernoulli–Gaussian prior based method results in 102 edges. For $p = 39$, convergence of the variational Bayes method can be achieved in about 26 min, the horseshoe prior based method in about 47 min for a given c , and the Bernoulli–Gaussian prior based method in about 52 min on a laptop computer with Windows operating system, 2.8 GHz of CPU, and 28 GB of RAM. Figure 5 shows the graphs of our proposed methods and Figure 6 shows the graphs of the existing methods.

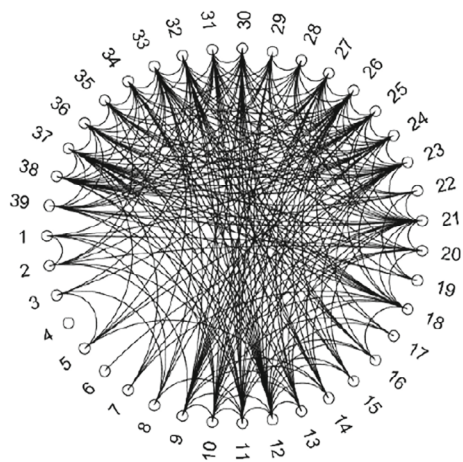
Since we use a sparsity inducing prior for each of the graphs, we consider the sparsity to compare the performance of the graphs. The variational Bayes and Bernoulli–Gaussian prior methods result in the sparsest graphs. The method based on the Horseshoe prior results in the densest graph. Out of the three proposed methods, this method is the most sensitive method, so it appears for this dataset, it is selecting more edges than the other models. The variational Bayes method is the fastest method out of the three proposed methods. The variational Bayes and Bernoulli–Gaussian prior methods proposed in this paper give sparser graphs than that based on the Gaussian copula graphical model, which uses a G-Wishart prior on the precision matrix. Sparse graphs can aid in simpler scientific

interpretation and could be used for further exploration, such as understanding the mechanisms involved in the isoprenoid pathway.

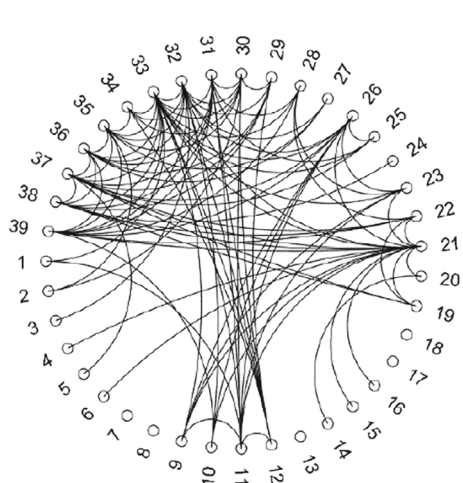
We also compare features related to the graphs. Wille et al. [52] found three subgroups in their GGM that were nearly or fully connected. They found that the genes DXR, MCT, CMK, and MECPS are nearly fully connected, the genes AACT2, HMGS, HMGR2, MK, MPDC1, FPPS1, and FPPS2 share many edges in the MVA pathway, and the subgroup AACT2, MK, MPDC1, and FPPS2 is completely interconnected [52]. We will refer to these subgroups as Subgroup 1, Subgroup 2, and Subgroup 3, respectively. The maximum number of edges in an undirected graph is $p(p - 1)/2$, where p is the number of nodes. The maximum number of edges for Subgroup 1, Subgroup 2, and Subgroup 3 is 6, 21, and 6, respectively. Table 1 shows the number of edges for each of the methods for the subgroups. The EBIC-selected method is not shown since it resulted in no edges. The RIC and StARS methods results in subgroups that have the highest number of edges. The Horseshow and Bayesian Copula methods have the next highest number of edges. The variational Bayes and Bernoulli–Gaussian have the least number of edges.



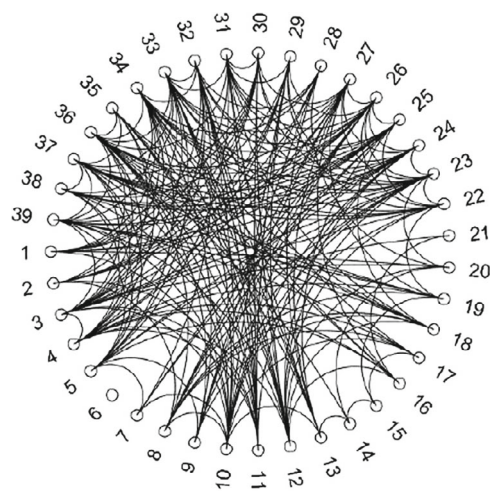
(A) Variational Bayesian method.



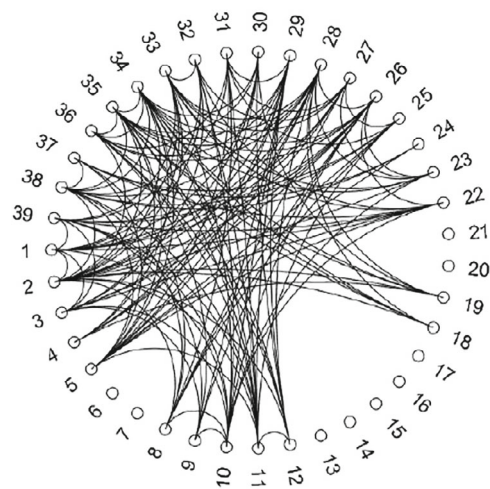
(B) Horseshoe method.



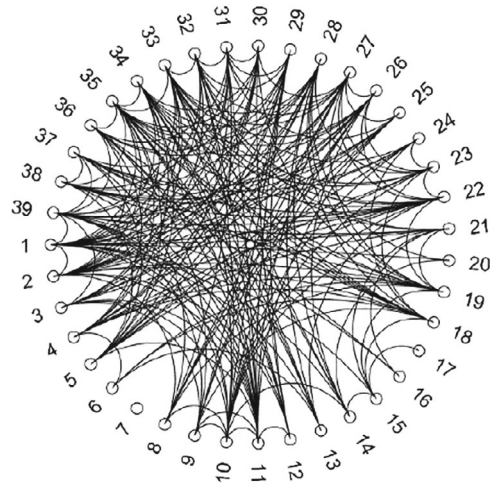
(C) Bernoulli-Gaussian method.



(A) Bayesian copula method.



(B) RIC method.



(C) StARS method.

FIGURE 5 Comparison of selected graphs from the proposed methods using gene expression data

FIGURE 6 Comparison of selected graphs from the existing methods using gene expression data

TABLE 1 Table showing the number of edges for each of the subgraphs for the methods

| Method | Subgroup 1 | Subgroup 2 | Subgroup 3 |
|--------------------|------------|------------|------------|
| Bernoulli–Gaussian | 3 | 5 | 3 |
| Horseshoe | 3 | 11 | 5 |
| Variational Bayes | 3 | 8 | 4 |
| RIC | 6 | 18 | 6 |
| StARS | 6 | 17 | 6 |
| Bayesian Copula | 6 | 10 | 3 |

10 | DISCUSSION

We have introduced a Bayesian regression method to construct graphical models for continuous data that do not rely on a normality assumption. The method assumes the nonparanormal structure, that under some unknown monotone transformations, the original observation vector reduces to a multivariate normal vector. The precision matrix of the transformed observations can be used to learn the graphical structure of conditional independence of the original observations. We use a prior distribution on the underlying transformations through a finite random series of B-splines with increasing coefficients that are given a multivariate truncated normal prior. We incorporate the positive definiteness constraint on the precision matrix of the transformed variables by utilizing the Cholesky decomposition. We consider two different priors based on the Cholesky decomposition, the Bernoulli–Gaussian prior and the horseshoe prior, and we impose a sparsity constraint. We use a VB algorithm to learn the conditional independence relations more efficiently as well as use a traditional Gibbs sampling approach. The VB approach and the approaches using Bernoulli–Gaussian and horseshoe priors result in most cases with better overall structure learning, measured using the Matthews correlation coefficient, than competing methods. The competing methods perform similarly or in some cases, better, than the proposed methods with smaller dimension. In addition, the VB algorithm performs similarly to the proposed methods in terms of overall structure learning and parameter estimation. It appears that information is not lost with the VB algorithm and we have the potential to speed up the estimation of the Bayesian nonparanormal graphical model. Lastly, when comparing the horseshoe to the Bernoulli–Gaussian prior, the horseshoe prior has higher sensitivity than the Bernoulli–Gaussian prior. Although the Bernoulli–Gaussian methods perform similarly to the horseshoe in terms of specificity and overall

structure learning, they do better parameter estimation. In summary, the proposed methods perform best at higher dimension ($p \geq 50$). Thus, for higher dimensional problems, we recommend using the VB algorithm to reduce the computational time while still maintaining good estimation properties.

Bayesian nonparanormal graphical models are flexible. They can be used to estimate the elements of the precision matrix directly or via a Cholesky decomposition. Researchers can try different sparsity inducing priors on the precision matrix based on their interests and needs. In addition, researchers can use a fully Bayesian approach to learn the graphical structure or employ a partially Bayesian approach to increase the speed in learning the structure without sacrificing much in quality. The Bernoulli–Gaussian prior, used in the VB method and the traditional Bayesian approach, resulted in the sparsest graphs using real data, which might be useful for researchers who would like greater variable reduction for data exploration.

ACKNOWLEDGMENTS

The work of Jami J. Mulgrave was supported by the National Science Foundation (NSF) Graduate Research Fellowship Program Grant no. DGE-1252376, the National Institutes of Health (NIH) training grant GM081057, and NSF grant DMS-1732842. The authors would like to acknowledge partial support for this project for Subhashis Ghosal by NSF grant DMS-1510238. Open access funding enabled and organized by Projekt DEAL.


CONFLICT OF INTERESTS

The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENT

The microarray dataset that support the findings of this study are openly available in Supplementary Materials at <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC545783/> [51].

ORCID

Jami J. Mulgrave  <https://orcid.org/0000-0003-3981-3917>

REFERENCES

1. A. Armagan, D. B. Dunson, and J. Lee, *Generalized double Pareto shrinkage*, *Stat. Sin.* 23 (2013), no. 1, 119–143. <https://doi.org/10.5705/ss.2011.048>
2. O. Banerjee, L. El Ghaoui, and A. d'Aspremont, *Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data*, *J. Mach. Learn. Res.* 9 (2008), 485–516. <http://dl.acm.org/citation.cfm?id=1390681.1390696>

3. S. Banerjee and S. Ghosal, *Bayesian structure learning in graphical models*, *J. Multivar. Anal.* 136 (2015), 147–162. <https://doi.org/10.1016/j.jmva.2015.01.015>
4. J. O. Berger and M. M. Barbieri, *Optimal predictive model selection*, *Ann. Stat.* 32 (2004), no. 3, 870–897. <https://doi.org/10.1214/009053604000000238>
5. A. Bhattacharya, A. Chakraborty, and B. K. Mallick, *Fast sampling with Gaussian scale-mixture priors in high-dimensional regression*, *Biometrika* 103 (2016), no. 4, 985–991. <https://doi.org/10.1093/biomet/asw042>
6. A. Bhattacharya, D. Pati, N. S. Pillai, and D. B. Dunson, *Dirichlet-Laplace priors for optimal shrinkage*, *J. Am. Stat. Assoc.* 110 (2015), no. 512, 1479–1490. <https://doi.org/10.1080/01621459.2014.960967>
7. C. M. Bishop, *Pattern recognition and machine learning (information science and statistics)*, Springer-Verlag, New York, 2006.
8. D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, *Variational inference: A review for statisticians*, *J. Am. Stat. Assoc.* 112 (2017), no. 518, 859–877. <https://doi.org/10.1080/01621459.2017.1285773>
9. C. M. Carvalho, N. G. Polson, and J. G. Scott, 2009: Handling sparsity via the horseshoe. *JMLR Workshop and Conference Proceedings*, 5, 73–80.
10. M. Chen, H. Wang, X. Liao, and L. Carin, 2011: Bayesian learning of sparse Gaussian graphical models. Working Paper.
11. J. Dahl, V. Roychowdhury, and L. Vandenberghe, 2005: Maximum likelihood estimation of Gaussian graphical models: Numerical implementation and topology selection. Technical report. University of California, Los Angeles. <http://www.seas.ucla.edu/vandenbe/publications/covsell.pdf>
12. J. Dahl, L. Vandenberghe, and V. Roychowdhury, *Covariance selection for nonchordal graphs via chordal embedding*, *Optimization Methods and Software* 23 (2008), no. 4, 501–520. <https://doi.org/10.1080/10556780802102693>
13. A. d'Aspremont, O. Banerjee, and L. El Ghaoui, *First-order methods for sparse covariance selection*, *SIAM Journal on Matrix Analysis and Applications* 30 (2008), no. 1, 56–66. <https://doi.org/10.1137/060670985>
14. A. Dobra and A. Lenkoski, *Copula Gaussian graphical models and their application to modeling functional disability data*, *Ann. Appl. Stat.* 5 (2011), no. 2A, 969–993. <https://doi.org/10.1214/10-AOAS397>
15. R. Foygel and M. Drton, *Extended Bayesian information criteria for Gaussian graphical models*, *Adv. Neural Inf. Proces. Syst.* 23 (2010), 604–612. http://books.nips.cc/papers/files/nips23/NIPS2010_0060.pdf
16. J. Friedman, T. Hastie, and R. Tibshirani, *Sparse inverse covariance estimation with the graphical lasso*, *Biostatistics* 9 (2008), no. 3, 432–441. <https://doi.org/10.1093/biostatistics/kxm045>
17. L. Gan, N. N. Narisetty, and F. Liang, *Bayesian regularization for graphical models with unequal shrinkage*, *J. Am. Stat. Assoc.* 114 (2019), no. 527, 1218–1231. <https://doi.org/10.1080/01621459.2018.1482755>
18. M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, *An introduction to variational methods for graphical models*, *Mach. Learn.* 37 (1999), no. 2, 183–233. <https://doi.org/10.1023/A:1007665907178>
19. Z. S. Khondker, H. Zhu, H. Chu, W. Lin, and J. G. Ibrahim, *The Bayesian covariance lasso*, *Statistics and Its Interface* 6 (2013), no. 2, 243–259.
20. M. Kuusimäki and M. J. Sillanpää, *Use of Wishart prior and simple extensions for sparse precision matrix estimation*, *PLoS One* 11 (2016), no. 2, e0148171. <https://doi.org/10.1371/journal.pone.0148171>
21. O. Laule, A. Furholz, H.-S. Chang, T. Zhu, X. Wang, P. B. Heifetz, W. Gruissem, and M. Lange, *Crosstalk between cytosolic and plastidial pathways of isoprenoid biosynthesis in Arabidopsis thaliana*, *Proc. Natl. Acad. Sci.* 100 (2003), no. 11, 6866–6871. <https://doi.org/10.1073/pnas.1031755100>
22. Z. R. Li, T. H. McComick, and S. J. Clark, *Using Bayesian latent Gaussian graphical models to infer symptom associations in verbal autopsies*, *Bayesian Anal.* 15 (2020), no. 3, 781–807. <https://doi.org/10.1214/19-BA1172>
23. Z. R. Li and T. H. McCormick, *An expectation conditional maximization approach for Gaussian graphical models*, *J. Comput. Graph. Stat.* 28 (2019), no. 4, 767–777. <https://doi.org/10.1080/10618600.2019.1609976>
24. H. Li and D. Pati, *Variable selection using shrinkage priors*, *Comput. Statist. Data Anal.* 107 (2017), 107–119. <https://doi.org/10.1016/j.csda.2016.10.008>
25. H. Liu, F. Han, M. Yuan, J. Lafferty, and L. Wasserman, *High-dimensional semiparametric Gaussian copula graphical models*, *Ann. Stat.* 40 (2012), no. 4, 2293–2326. <http://www.jstor.org/stable/41806536>
26. H. Liu, J. D. Lafferty, and L. A. Wasserman, *The nonparanormal: Semiparametric estimation of high dimensional undirected graphs*, *J. Mach. Learn. Res.* 10 (2009), 2295–2328. <http://jmlr.csail.mit.edu/papers/volume10/liu09a/liu09a.pdf>
27. H. Liu, K. Roeder, and L. Wasserman, *Stability approach to regularization selection (StARS) for high dimensional graphical models*, *Advances in Neural Information Processing Systems* 23 (2010), 1432–1440. <http://dl.acm.org/citation.cfm?id=2997046.2997056>
28. Z. Lu, *Smooth optimization approach for sparse covariance selection*, *SIAM J. Optim.* 19 (2009), no. 4, 1807–1827. <https://doi.org/10.1137/070695915>
29. S. Lysen, 2009: *Permuted inclusion criterion: A variable selection technique*. Ph.D. thesis, Publicly Accessible Penn Dissertations, 28. <http://repository.upenn.edu/edissertations/28>
30. R. Mazumder and T. Hastie, *Exact covariance thresholding into connected components for large-scale graphical lasso*, *J. Mach. Learn. Res.* 13 (2012), 781–794.
31. R. Mazumder and T. Hastie, *The graphical lasso: New insights and alternatives*, *Electronic Journal of Statistics* 6 (2012), 2125–2149. <https://doi.org/10.1214/12-EJS740>
32. N. Meinshausen and P. Bühlmann, *High-dimensional graphs and variable selection with the lasso*, *Ann. Stat.* 34 (2006), no. 3, 1436–1462. <https://doi.org/10.1214/009053606000000281>
33. A. Mohammadi, F. Abegaz, E. van den Heuvel, and E. C. Wit, *Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models*, *J. R. Stat. Soc.: Ser. C: Appl. Stat.* 66 (2017), no. 3, 629–645. <https://doi.org/10.1111/rssc.12171>
34. A. Mohammadi and E. C. Wit, *Bayesian structure learning in sparse Gaussian graphical models*, *Bayesian Anal.* 10 (2015), no. 1, 109–138. <https://doi.org/10.1214/14-BA889>
35. R. Mohammadi and E. C. Wit, *BDgraph: An R package for Bayesian structure learning in graphical models*, *J. Stat. Softw.* 89 (2019), no. 3, 1–30. <https://doi.org/10.18637/jss.v089.i03>

36. J. J. Mulgrave and S. Ghosal, *Bayesian inference in nonparanormal graphical models*, *Bayesian Anal.* 15 (2020), no. 2, 449–475. <https://doi.org/10.1214/19-BA1159>
37. S. E. Neville, J. T. Ormerod, and M. P. Wand, *Mean field variational Bayes for continuous sparse signal shrinkage: Pitfalls and remedies*, *Electron. J. Statist.* 8 (2014), no. 1, 1113–1151. <https://doi.org/10.1214/14-EJS910>
38. T. K. H. Nguyen and M. Chiogna, 2018: Structure learning of undirected graphical models for count data. arXiv:1810.10854. <http://arxiv.org/abs/1810.10854>
39. J. T. Ormerod, C. You, and S. Müller, *A variational Bayes approach to variable selection*, *Electron. J. Statist.* 11 (2017), no. 2, 3549–3594. <https://doi.org/10.1214/17-EJS1332>
40. A. Pakman and L. Paninski, *Exact Hamiltonian Monte Carlo for truncated multivariate Gaussians*, *J. Comput. Graph. Stat.* 23 (2014), no. 2, 518–542. <https://doi.org/10.1080/10618600.2013.788448>
41. C. B. Peterson, F. C. Stingo, and M. Vannucci, *Joint Bayesian variable and graph selection for regression models with network-structured predictors*, *Stat. Med.* 35 (2016), no. 7, 1017–1031. <https://doi.org/10.1002/sim.6792>
42. C. Peterson, M. Vannucci, C. Karakas, W. Choi, L. Ma, and M. Maletic-Savatic, *Inferring metabolic networks using the Bayesian adaptive graphical lasso with informative priors*, *Statist. Its Interface* 6 (2013), no. 4, 547–558. <https://doi.org/10.4310/SII.2013.v6.n4.a12>
43. M. Pitt, D. Chan, and R. Kohn, *Efficient Bayesian inference for Gaussian copula regression models*, *Biometrika* 93 (2006), no. 3, 537–554. <http://www.jstor.org/stable/20441306>
44. M. Pourahmadi, *Covariance estimation: The GLM and regularization perspectives*, *Stat. Sci.* 26 (2011), no. 3, 369–387. <https://doi.org/10.1214/11-STS358>
45. A. J. Rothman, P. J. Bickel, E. Levina, and J. Zhu, *Sparse permutation invariant covariance estimation*, *Electron. J. Statist.* 2 (2008), 494–515. <https://doi.org/10.1214/08-EJS176>
46. K. Scheinberg, S. Ma, and D. Goldfarb, 2010: *Sparse inverse covariance selection via alternating linearization methods*. Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2, Curran Associates Inc., 2101–2109. <http://dl.acm.org/citation.cfm?id=2997046.2997130>
47. C. Soussen, J. Idier, D. Brie, and J. Duan, *From Bernoulli-Gaussian deconvolution to sparse signal restoration*, *IEEE Trans. Signal Process.* 59 (2011), no. 10, 4572–4584. <https://doi.org/10.1109/TSP.2011.2160633>
48. S. L. van der Pas, B. J. K. Kleijn, and A. W. van der Vaart, *The horseshoe estimator: Posterior concentration around nearly black vectors*, *Electron. J. Statist.* 8 (2014), no. 2, 2585–2618. <https://doi.org/10.1214/14-EJS962>
49. M. J. Wainwright and M. I. Jordan, *Graphical models, exponential families, and variational inference*, *Found. Trends® Mach. Learn.* 1 (2007), no. 1–2, 1–305. <https://doi.org/10.1561/2200000001>
50. H. Wang, *Bayesian graphical lasso models and efficient posterior computation*, *Bayesian Anal.* 7 (2012), no. 4, 867–886. <https://doi.org/10.1214/12-BA729>
51. H. Wang, *Scaling it up: Stochastic search structure learning in graphical models*, *Bayesian Anal.* 10 (2015), no. 2, 351–377. <https://doi.org/10.1214/14-BA916>
52. A. Wille, P. Zimmermann, E. Vranová, A. Fürholz, O. Laule, S. Bleuler, L. Hennig, A. Prelic, P. von Rohr, L. Thiele, E. Zitzler, W. Gruissem, and P. Bühlmann, *Sparse graphical Gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana*, *Genome Biol.* 5 (2004), no. 11, R92–R92. <https://doi.org/10.1186/gb-2004-5-11-r92>
53. D. R. Williams, J. Piironen, A. Vehtari, and P. Rast, 2018: Bayesian estimation of Gaussian graphical models with projection predictive selection. arXiv:1801.05725. <https://arxiv.org/abs/1801.05725>
54. D. M. Witten, J. H. Friedman, and N. Simon, *New insights and faster computations for the graphical lasso*, *J. Comput. Graph. Stat.* 20 (2011), no. 4, 892–900. <https://doi.org/10.1198/jcgs.2011.11051a>
55. E. Wong, S. P. Awate, and P. T. Fletcher, 2013: *Adaptive sparsity in Gaussian graphical models*. Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, Atlanta, GA, USA. 311–319 <http://dl.acm.org/citation.cfm?id=3042817.3042854>
56. M. Yuan and Y. Lin, *Model selection and estimation in the Gaussian graphical model*, *Biometrika* 94 (2007), no. 1, 19–35. <https://doi.org/10.1093/biomet/asm018>
57. Q. Zhang, *Testing differential gene networks under nonparanormal graphical models with false discovery rate control*, *Genes* 11 (2020), no. 2, 167. <https://doi.org/10.3390/genes11020167>
58. T. Zhao, X. Li, H. Liu, K. Roeder, J. Lafferty, and L. Wasserman, 2015: *Huge: High-dimensional undirected graph estimation*. R Package Version 1.2.7. <http://CRAN.R-project.org/package=huge>

How to cite this article: J. J. Mulgrave, and S. Ghosal, *Regression-based Bayesian estimation and structure learning for nonparanormal graphical models*, *Stat. Anal. Data Min.: ASA Data Sci. J.* **15** (2022), 611–629. <https://doi.org/10.1002/sam.11576>

APPENDIX A. HORSESHOE POSTERIOR

The joint posterior distribution is

$$\begin{aligned}
 p(\boldsymbol{\beta}, \lambda^2, \sigma^2, \mathbf{a}, \mathbf{b}, \mathbf{h}, \boldsymbol{\theta}, \boldsymbol{\mu} \mid \mathbf{Z}) &\propto \prod_{i=1}^n \prod_{d=1}^{p-1} p\left(\sum_{j=1}^J \theta_{dj} \mathbf{B}_j(X_{id}) \mid \right. \\
 &\quad \times \left. \sum_{j=1}^J \theta_{k>d,j} \mathbf{B}_j(X_{i,k>d}), \boldsymbol{\beta}_{k>d}, \sigma_d^2\right) \\
 &\quad \times p(\boldsymbol{\beta}_{k>d}) p(\sigma_d^2) p(\boldsymbol{\theta}_d) p(\boldsymbol{\mu}_d) \\
 &\quad \times p(\mathbf{a}_d) p(\mathbf{b}_{k>d}) p(\mathbf{h}_{k>d}) p(\lambda_d^2) \\
 &\quad \times p\left(\sum_{j=1}^J \theta_{pj} \mathbf{B}_j(X_{ip}) \mid \sigma_p^2\right) \\
 &\quad p(\sigma_p^2) p(\boldsymbol{\theta}_p) p(\boldsymbol{\mu}_p).
 \end{aligned}$$

Then the corresponding conditional posterior distributions are given by

$$\begin{aligned} \beta_{k>d} &\sim \mathcal{N} \left[\left(\mathbf{Z}'_{k>d} \mathbf{Z}_{k>d} + \text{diag} \left(\frac{p^2 k}{\lambda_d^2 \mathbf{b}_{k>d} c^2} \right) \right)^{-1} \mathbf{Z}'_{k>d} \mathbf{Z}_d, \right. \\ &\quad \left. \times \sigma_d^2 \left(\mathbf{Z}'_{k>d} \mathbf{Z}_{k>d} + \text{diag} \left(\frac{p^2 k}{\lambda_d^2 \mathbf{b}_{k>d} c^2} \right) \right)^{-1} \right], \\ \lambda_d^2 &\sim \text{IG} \left(\frac{\#(k > d)}{2} + \frac{1}{2}, \frac{1}{2} \beta'_{k>d} \right. \\ &\quad \left. \text{diag} \left(\frac{p^2 k}{\sigma_d^2 \mathbf{b}_{k>d} c^2} \right) \beta_{k>d} + \frac{1}{a_d} \right), \\ a_d &\sim \text{IG} \left(1, \frac{1}{\lambda_d^2} + 1 \right), \\ b_{kd} &\sim \text{IG} \left(1, \frac{k \beta_{kd}^2 p^2}{2 \sigma_d^2 \lambda_d^2 c^2} + \frac{1}{h_{kd}} \right), \\ h_{kd} &\sim \text{IG} \left(1, \frac{1}{b_{kd}} + 1 \right), \\ \sigma_d^2 &\sim \text{IG} \left(\frac{n + \#(k > d)}{2} + A, \right. \\ &\quad \left. \frac{1}{2} \|\mathbf{Z}_d - \mathbf{Z}_{k>d} \beta_{k>d}\|^2 \right. \\ &\quad \left. + \frac{1}{2} \beta'_{k>d} \text{diag} \left(\frac{p^2 k}{\lambda_d^2 \mathbf{b}_{k>d} c^2} \right) \beta_{k>d} + B \right), \\ \sigma_p^2 &\sim \text{IG} \left(\frac{n}{2} + A, \frac{1}{2} \|\mathbf{Z}_p\|^2 + B \right). \end{aligned}$$

Since sampling the $\beta_{k>d}$ can be expensive for large p , we use an exact sampling algorithm for Gaussian priors based on data augmentation [5].

APPENDIX B. BERNOULLI-GAUSSIAN POSTERIOR

The joint posterior distribution is

$$\begin{aligned} p(\boldsymbol{\beta}, Y | \mathbf{Z}) &\propto \prod_{i=1}^n \prod_{d=1}^{p-1} p \left(\sum_{j=1}^J \theta_{dj} \mathbf{B}_j (X_{id}) \mid \right. \\ &\quad \left. \times \sum_{j=1}^J \theta_{k>d,j} \mathbf{B}_j (X_{i,k>d}), \boldsymbol{\beta}_{k>d}, Y_{k>d}, \sigma_d^2 \right) \\ &\quad \times p(\boldsymbol{\beta}_{k>d}) p(Y_{k>d}) \\ &\quad \times p(\sigma_d^2) p(\theta_d) p(\mu_d) p \left(\sum_{j=1}^J \theta_{pj} \mathbf{B}_j (X_{ip}) \mid \sigma_p^2 \right) \\ &\quad \times p(\sigma_p^2) p(\theta_p) p(\mu_p). \end{aligned}$$

Then the corresponding conditional posterior distributions are given by

$$\begin{aligned} \beta_{k>d} \mid \cdot &\sim \mathcal{N} \left[\left(Y_{k>d} \mathbf{Z}'_{k>d} \mathbf{Z}_{k>d} Y_{k>d} + \frac{\sigma_d^2}{g^2} \mathbf{I} \right)^{-1} Y_{k>d} \mathbf{Z}'_{k>d} \mathbf{Z}_d, \right. \\ &\quad \left. \times \left(Y_{k>d} \mathbf{Z}'_{k>d} \mathbf{Z}_{k>d} Y_{k>d} + \frac{\sigma_d^2}{g^2} \mathbf{I} \right)^{-1} \right], \\ v_k \mid \cdot &\sim \text{Ber} \left[\text{expit} \left\{ \text{logit}(\rho_{kd}^*) - \frac{1}{2\sigma_d^2} \|\mathbf{Z}_k\|^2 \beta_k^2 \right. \right. \\ &\quad \left. \left. + \frac{1}{\sigma_d^2} \beta_k \mathbf{Z}'_{k'} (\mathbf{Z}_d - \mathbf{Z}_{l>k} Y_{l>k} \beta_{l>k}) \right\} \right], \\ \sigma_d^2 &\sim \text{IG} \left(\frac{n}{2} + A, \frac{1}{2} \|\mathbf{Z}_d - \mathbf{Z}_{k>d} Y_{k>d} \beta_{k>d}\|^2 + B \right), \\ p(\sigma_p^2) &\sim \text{IG} \left(\frac{n}{2} + A, \frac{1}{2} \|\mathbf{Z}_p\|^2 + B \right), \end{aligned}$$

where $k = d + 1, \dots, p$, and $d = 1, \dots, p - 1$.

Again, to sample $\beta_{k>d}$, we used an exact sampling algorithm for Gaussian priors that invokes data augmentation [5].

APPENDIX C. GITHUB REPOSITORY

The code used to run the methods described in this paper are available on GitHub: <https://github.com/jnj2102/BayesianRegressionApproach>.

Algorithm 1. Variational Bayesian Algorithm

- 1: Gibbs Sampler: Estimate $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$
- 2: **for** $d = 1 : p$ **do**
 - (a) Sample $\bar{\boldsymbol{\theta}}_d \mid (\bar{\boldsymbol{\Theta}}_{-d}, \mathbf{Y}, \boldsymbol{\mu}, \boldsymbol{\Omega}) \sim \text{TN}(\boldsymbol{\gamma}, \boldsymbol{\Psi}, \{\bar{\mathbf{F}}_d \bar{\boldsymbol{\theta}}_d + \bar{\boldsymbol{g}}_d > \mathbf{0}\})$, where $\boldsymbol{\gamma}$ and $\boldsymbol{\Psi}$ are defined in Section 3.1 of [36].
- 3: **end for**
- 4: Repeat Step 2 until convergence.
- 5: Compute $\hat{\boldsymbol{\theta}}_d = \sum_{m=1}^M \boldsymbol{\theta}_{dm}$ and $\hat{\mu}_d = \sum_{m=1}^M \mu_{dm}$, where M is the number of Markov Chain Monte Carlo samples.
- 6: Compute $Z_{id} = \sum_{j=1}^J \hat{\theta}_{jd} B_j(X_{id}) - \hat{\mu}_d$.
- 7: Using \mathbf{Z} , tune ρ_{kd}^* and find the initial values for $\mathbf{w}_{k>d}$ using the tuning procedure described in Subsection 3.1.
- 8: Coordinate Ascent Variational Inference: To compute $\boldsymbol{\Omega}$
 - (a) Initialize with $t = 1, \mathbf{Z}_d, \mathbf{Z}_{k>d}, g^2, A, B, \tau_0, \rho_d^*, \mathbf{w}_{k>d}$ where $\mathbf{w}_{k>d}^{(1)} \in [0, 1]^{\#(k>d)}$
 - (b) **for** $d = 1 : p - 1$
 - $\mathbf{W}_d^{(t)} = \text{diag}(\mathbf{w}_{k>d}^{(t)})$

- $\Omega_d = \mathbf{w}_d^{(t)} \mathbf{w}_d^{(t)'} + \mathbf{W}_d^{(t)} (\mathbf{I} - \mathbf{W}_d^{(t)})$
- $\Sigma_d^{(t)} = [\tau_d^{(t-1)} (\mathbf{Z}_{k>d} \mathbf{Z}_{k>d}' \circ \Omega_d^{(t)} + \mathbf{g}^{-2} \mathbf{I})]^{-1}$
- $\mu_d^{(t)} = \tau_d^{(t-1)} \Sigma_d^{(t)} \mathbf{W}_d^{(t)} \mathbf{Z}_{k>d} \mathbf{Z}_d$
- $s_d = B + \frac{1}{2} [\|\mathbf{Z}_d\|^2 - 2\mathbf{Z}_d' \mathbf{Z}_{k>d} \mathbf{W}_d^{(t)} \mu_d^{(t)} + \text{tr}\{(\mathbf{Z}_{k>d} \mathbf{Z}_{k>d}' \circ \Omega_d^{(t)}) (\mu_d^{(t)} \mu_d^{(t)'} + \Sigma_d^{(t)})\}]$
- $\tau_d^{(t)} = \frac{2A+n}{2s_d}$
- $\mathbf{w}_d^* = \mathbf{w}_d^{(t)}$
- **for** $k = (d+1) : p$ **do**
 - $\eta_{kd} = \text{logit}(\rho_{kd}^*) - \frac{\tau_d^{(t)}}{2} ((\mu_k^{(t)})^2 + \Sigma_{k,k}^{(t)}) \|\mathbf{Z}_k\|^2 + \tau_d^{(t)} [\mu_k^{(t)} \mathbf{Z}_k' \mathbf{Z}_d - \mathbf{Z}_k' \mathbf{Z}_l \mathbf{W}_l^{(t)} (\mu_l^{(t)} \mu_k^{(t)} + \Sigma_{l,k}^{(t)})]$
 - $w_{kd}^* = \text{expit}(\eta_{kd})$
- **end for**
- $\mathbf{w}_d^{(t+1)} = \mathbf{w}_d^*$
- (c) **end for**
- (d) $s_p^{(t)} = B + \frac{1}{2} [\|\mathbf{Z}_p\|^2]$
- (e) Repeat (b)–(d) until $|\text{VLB}(\mathbf{Z}, \rho)^{(t)} - \text{VLB}(\mathbf{Z}, \rho)^{(t-1)}| < \epsilon$.
- 9: Sample $\beta_d \sim N(\mu_d, \Sigma_d)$, $v_{kd} \sim \text{Ber}(w_{kd})$, $\sigma_d \sim \text{IG}(A + n/2, s_d)$, and $\sigma_p \sim \text{IG}(A + n/2, s_p)$
- 10: Compute $l_{kd} = -v_{kd} \beta_{kd} / \sigma_d$ and $l_{dd} = 1 / \sigma_d$.
- 11: Compute $\Omega = \mathbf{L}\mathbf{L}'$.

Algorithm 2. Horseshoe Gibbs Algorithm

- 1: Gibbs Sampler: Estimate θ , μ , and Ω :
- 2: **for** $\bar{d} = 1 : p$ **do**
 - (a) $\bar{\theta}_d | (\bar{\Theta}_{-d}, \mathbf{Y}, \mu, \Omega) \sim \text{TN}(\gamma, \Psi, \{\bar{\mathbf{F}}_d \bar{\theta}_d + \bar{\mathbf{g}}_d > \mathbf{0}\})$ where γ and Ψ are defined in Section 3.1 of [36].
- 3: **end for**
- 4: Compute $Y_{id} = \sum_{j=1}^J \theta_{dj} B_j(X_{id})$.
- 5: Sample $\mu | (\mathbf{Y}, \Omega) \sim N_p(\bar{\mathbf{Y}}, \frac{1}{n} \Omega^{-1})$.
- 6: Compute $Z_{id} = Y_{id} - \mu_d$.
- 7: **for** $\bar{d} = 1 : p-1$ **do**
 - (a) Sample $\beta_{k>d} | \sigma_d, \mathbf{b}_{k>d}, \lambda_d^2 \sim N(\mathbf{A}^{-1} \mathbf{Z}_{k>d}' \mathbf{Z}_d, \sigma_d^2 \mathbf{A}^{-1})$, where $\mathbf{A} = (\mathbf{Z}_{k>d}' \mathbf{Z}_{k>d} + \text{diag}(p^2 k / (\lambda_d^2 \mathbf{b}_{k>d} c^2)))$:
 - (i) Sample $t \sim N(\mathbf{0}, \mathbf{D})$ and $\delta \sim \text{Normal}(0, I_n)$, where $\mathbf{D} = \sigma_d^2 \text{diag}(\lambda_d^2 \mathbf{b}_{k>d} c^2 / (p^2 k))$.
 - (ii) set $v = \Phi t + \delta$, where $\Phi = \mathbf{Z}_{k>d} / \sigma_d$.
 - (iii) solve for w in $(\Phi \mathbf{D} \Phi' + I_n) w = (\alpha - v)$, where $\alpha = \mathbf{Z}_d / \sigma_d$.
 - (iv) set $\beta = t + \mathbf{D} \Phi' w$.
 - (b) Sample $\lambda_d^2 \sim \text{IG}\left(\frac{\#(k > d)}{2} + \frac{1}{2}, \frac{1}{2} \beta_{k>d}' \text{diag}\left(\frac{p^2 k}{\sigma_d^2 \mathbf{b}_{k>d} c^2}\right) \beta_{k>d} + \frac{1}{\alpha_d}\right)$.
 - (c) Sample $a_d \sim \text{IG}(1, \lambda_d^{-2} + 1)$.

- (d) Sample $b_{kd} \sim \text{IG}\left(1, \frac{p^2 k \beta_{kd}^2}{2\sigma_d^2 \lambda_d^2 c^2} + \frac{1}{h_{kd}}\right)$.
- (e) Sample $h_{kd} \sim \text{IG}(1, b_{kd}^{-1} + 1)$.
- (f) Sample $\sigma_d^2 \sim \text{IG}\left(\frac{n + \#(k > d)}{2} + A, \frac{1}{2} \|\mathbf{Z}_d - \mathbf{Z}_{k>d} \beta_{k>d}\|^2 + \frac{1}{2} \beta_{k>d}' \text{diag}\left(\frac{p^2 k}{\lambda_d^2 \mathbf{b}_{k>d} c^2}\right) \beta_{k>d} + B\right)$.

8: **end for**

9: Sample

$$\sigma_p^2 \sim \text{IG}\left(\frac{n}{2} + A, \frac{1}{2} \|\mathbf{Z}_p\|^2 + B\right). \quad (\text{B1})$$

10: Compute $l_{kd} = -\beta_{kd} / \sigma_d$ and $l_{dd} = 1 / \sigma_d$.11: Compute $\Omega = \mathbf{L}\mathbf{L}'$.

12: These steps are repeated until convergence.

Algorithm 3. Bernoulli–Gaussian Gibbs Algorithm

1: Gibbs Sampler: Estimate θ , μ , and Ω :2: **for** $\bar{d} = 1 : p$ **do**

- (a) Sample $\bar{\theta}_d | (\bar{\Theta}_{-d}, \mathbf{Y}, \mu, \Omega) \sim \text{TN}(\gamma, \Psi, \{\bar{\mathbf{F}}_d \bar{\theta}_d + \bar{\mathbf{g}}_d > \mathbf{0}\})$, where γ and Ψ are defined in Section 3.1 of [36].

3: **end for**4: Compute $Y_{id} = \sum_{j=1}^J \theta_{dj} B_j(X_{id})$.5: Sample $\mu | (\mathbf{Y}, \Omega) \sim N_p(\bar{\mathbf{Y}}, \frac{1}{n} \Omega^{-1})$.6: Compute $Z_{id} = Y_{id} - \mu_d$.7: **for** $\bar{d} = 1 : p-1$ **do**

- (a) Sample $\beta_{k>d} | \sigma_d, \mathbf{Y}_{k>d} \sim N(\mathbf{A}^{-1} \mathbf{Y}_{k>d} \mathbf{Z}_{k>d}' \mathbf{Z}_d, \sigma_d^2 \mathbf{A}^{-1})$, where $\mathbf{A} = (\mathbf{Y}_{k>d} \mathbf{Z}_{k>d}' \mathbf{Z}_{k>d} \mathbf{Y}_{k>d} + \frac{\sigma_d^2}{g^2} \mathbf{I})$.
 - (i) Sample $t \sim N(\mathbf{0}, \mathbf{D})$ and $\delta \sim N(\mathbf{0}, I_n)$, where $\mathbf{D} = g^2 \mathbf{I}$;
 - (ii) set $v = \Phi t + \delta$, where $\Phi = \mathbf{Z}_{k>d} \mathbf{Y}_{k>d} / \sigma_d$;
 - (iii) solve for q in $(\Phi \mathbf{D} \Phi' + I_n) q = (\alpha - v)$, where $\alpha = \mathbf{Z}_d / \sigma_d$;
 - (iv) set $\beta = t + \mathbf{D} \Phi' q$.
- (b) Sample $v_k | \beta_k, \sigma_d \sim \text{Ber}[\text{expit}\{\text{logit}(\rho_{kd}^*) - \frac{1}{2\sigma_d^2} \|\mathbf{Z}_k\|^2 \beta_k^2 + \frac{1}{\sigma_d^2} \beta_k \mathbf{Z}_k' (\mathbf{Z}_d - \mathbf{Z}_{l>k} \mathbf{Y}_{l>k} \beta_{l>k})\}]$.
- (c) Sample $\sigma_d^2 | \beta_{k>d}, \mathbf{Y}_{k>d} \sim \text{IG}\left(\frac{n}{2} + A, \frac{1}{2} \|\mathbf{Z}_d - \mathbf{Z}_{k>d} \mathbf{Y}_{k>d} \beta_{k>d}\|^2 + B\right)$.

8: **end for**9: Sample $\sigma_p^2 | \mathbf{Z}_p \sim \text{IG}(\frac{n}{2} + A, \frac{1}{2} \|\mathbf{Z}_p\|^2 + B)$.10: Compute $l_{kd} = -v_{kd} \beta_{kd} / \sigma_d$ and $l_{dd} = 1 / \sigma_d$.11: Compute $\Omega = \mathbf{L}\mathbf{L}'$.

12: These steps are repeated until convergence.