Università di Firenze, Università di Perugia, INdAM consorziate nel CIAFM

DOTTORATO DI RICERCA
IN MATEMATICA, INFORMATICA, STATISTICA

CURRICULUM IN STATISTICA
CICLO XXXV

Sede amministrativa Università degli Studi di Firenze
Coordinatore Prof. Matteo Focardi

Bayesian methods for high-dimensional applications

Settore Scientifico Disciplinare SECS-S/01

Dottorando:
Claudio Busatto

Tutore
Prof. Francesco Claudio Stingo

Coordinatore
Prof. Matteo Focardi

Anni 2019/2022

Contents

List of Figures

List of Tables

List of Algorithms

Introduction

This thesis deals with Bayesian methods for different high-dimensional applications and faces the difficult challenges of prediction and variable selection when the number of covariates is much greater than the number of observations. Chapter 1 aims to explain the usefulness of these methods in a general framework. It introduces the main controversial topics behind high-dimensional data from both a statistical and computational point of view and gives an overview of the statistical methods used throughout this dissertation. The presented methods rely on different types of shrinkage priors for sparse models: Chapter 2 discusses a new class of fast Bayesian spike-and-slab algorithms for continuous outcome, which relies on a group of efficient updating methods based on the thinQR decomposition; Chapter 3 introduces a novel multivariate shrinkage prior for modelling multiple correlated networks; Chapter 4 presents a flexible way to include prior information in the estimation process improving prediction and variable selection. Final discussions and comments are presented in Chapter 5.

Chapter 1

Analysis of high-dimensional data

In many scientific fields where new data are collected with automated technologies, the main interest relies on the analysis of datasets with a large number of features. High-dimensional data are defined as data with a large number of observed variables, p, and a small number of observations, n. Note that the ratio between n and p must be small for the data to be high-dimensional, that is, a dataset with 10000 features and 100000 observations is considered as low-dimensional. The reasons behind a small sample size are mainly due to time and budget limitations or practical restrictions (for example the study of rare diseases involving restricted populations).

The analysis of high-dimensional data requires the application of non-standard approaches, as common methods such as linear regression can not be estimated when $n < p$. Even when the number of samples is slightly greater than the number of variables, classical methods incur the so-called *curse of dimensionality* (Bellman, 1961) and the quality of their estimates deteriorates. Indeed, in order for linear regression's results to be reliable, the needed number of observations grows exponentially with the dimensionality of the problem (Hastie et al., 2009).

Different statistical and computational problems arise from large amount of observed variables and a small number of available observations:

- Two practical problems with high-dimensional data are data visualization and exploration, as it becomes impossible to plot the response variable against each predictor in order to to identify the factors with a more likely significant effect on the outcome;

- Few observations are associated to low degree of information. When the sample size is not large enough, some of the variables are falsely selected, as the effect on the outcome happens by chance and can not be generalized to the whole population;

- When the number of covariates increases, the correlations among the predictors are more likely and the interactions with the response variable become complex and difficult to model. Under these circumstances, classical methods fail to provide an accurate variable selection procedure;

- The high number of variables makes the exploration of all subsets computationally infeasible, as the computational time increases exponentially with the dimensionality (Saeys et al., 2007);

- In *large p / small n* analysis it is easy to face the problem of *overfitting* (Raudys and Jain, 1991), which happens when the model fits (almost) perfectly the training data and fails to generalize to the whole population, resulting in poor prediction performances.

A common approach to overcome the statistical problems above is given by penalized regression methods, which are deterministic extensions of the ordinary linear regression. These models have been widely used due to their ability of dealing with high correlations among the predictors. They add a constraint on the dimension of the regression model and minimize a loss function (usually the residual sum of squares) which includes one or more penalty parameters, with the goal of decreasing the collinearity between variable by penalizing the inclusion of a predictor in the model. The penalty parameter(s) plays a key role by shrinking the estimates towards zero and reducing the dimensionality of the problem. Such improvements, however, come with a cost: penalized regressions introduce bias in the estimates in order to reduce their variance. That is, it is better to be slightly wrong all the time than to be perfectly correct sometimes and completely wrong some others. This concept is known as the *bias-variance trade-off* and usually the decrease of the variance is greater than the increase of the bias. This way the estimated model is more generalizable and the prediction outside training data becomes more accurate. Some examples of penalized regression are Ridge regression (Hoerl and Kennard, 1970) and LASSO (Tibshirani, 1996), which minimize the squared norm of the regression parameters (l_2 penalization) and the sum of their absolute values (l_1 penalization), respectively. A compromise between these two approaches is given by the Elastic-Net regression (Zou and Hastie, 2005), which attempts to overcome their limitations by combining the l_1 and l_2 penalizations. Another common method is the LARS algorithm of Efron et al. (2004).

These methods represent the *golden standard* techniques, but they are too simplistic and often fail when the dimensionality of the problem is huge. Therefore, modern developments focus on the extension of such methods. Within the penalization context, Bayesian inference has become a widely applied tool. As for penalized regression, Bayesian methods introduce bias in the model in order to improve the overall performances. On the other hand, they provide a much more

flexible approach and a natural way to include external information. In Bayesian statistics the parameters are treated as random variables and the inclusion of prior knowledge in the model is allowed by assuming a prior distribution. This approach provides a probabilistic process for the update prior beliefs in light of observed data. When compared to their deterministic counterpart, Bayesian models present several advantages. Above all, parameters become interpretable and they are not abstract numbers anymore. Indeed, Bayesian models provide a posterior distribution for the parameters rather than a point estimate and allow to quantify the uncertainty in the estimates following the posterior standard deviations. In penalized regression, the evaluation of the standard deviations, especially for the penalty parameter, can be troublesome, with unreliable and unstable results in the case of sandwich and bootstrap estimates (Kyung et al., 2010). The Bayesian posterior distribution also allows to retrieve credible intervals for each parameter, providing a useful tool for posterior inference which can be of great interest in many scientific fields such as biology and genomics. Other advantages concerning Bayesian estimation methods are: first, when dealing with multiple penalty parameters, these can be evaluated jointly with the other parameters of interest, thus avoiding the need of cross-validation procedures; second, most of these methods rely on Markov Chain Monte Carlo (MCMC) sampling algorithms, which provide a more flexible tool than optimization when facing non-convex penalties. The main drawback of Bayesian inference is the computational efficiency, as the implementation of iterative sampling procedures until convergence negatively affect the computational performances.

The following section provides a brief introduction to Bayesian inference and an overview of Bayesian shrinkage methods for the analysis of high-dimensional data.

1.1 Bayesian inference

As opposed to the *frequentist* perspective, where the model parameters are considered fixed quantities to be estimated, Bayesian methods treat the parameters as random variables and require the specification of a prior distribution alongside the likelihood function. The parameters of the prior distributions are called hyperparameters and their choice guides the amount prior knowledge to be included in the estimation process. When no prior evidence is available, non-informative specifications for the prior distributions, such as Uniform distributions, can be implemented and, therefore, the estimates are guided only by the data. The final goal is the analysis of the posterior distribution conditionally on the observed data, which can be retrieved with Bayes' theorem, and the parameters are usually estimated by selecting the posterior mean or mode. The posterior distribution is

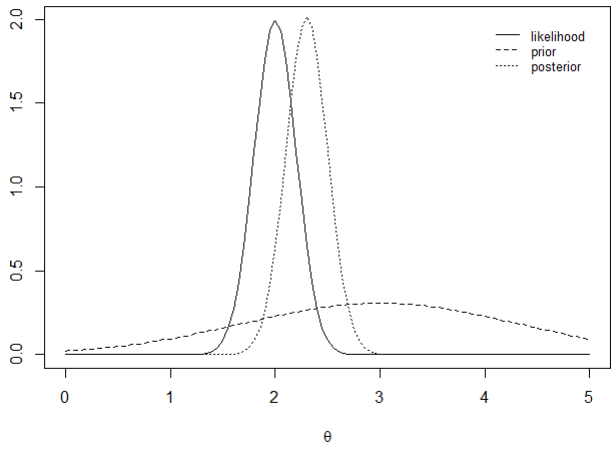


Figure 1.1: Bayes' theorem with $\mathbf{y} \mid \boldsymbol{\theta} \sim \mathcal{N}(2, 0.2^2)$ and $\boldsymbol{\theta} \sim \mathcal{N}(3, 1.3^2)$.

a combination between the likelihood function and the prior density. When data are low-dimensional, the choice of the hyperparameters has a small influence on the final results. However, when dealing with high-dimensional data, the posterior distribution becomes more sensitive to the specification of the prior. In this case, selecting a good prior becomes particularly important. Many attempts at selecting the best hyperparameters have been made, however there is not a prevalent approach and is still an open subject of research. Among others, Empirical Bayes (Casella, 1985) estimates the hyperparameters from the data, whereas modern developments include external information in the estimation process and model the hyperparameters as a function of *complementary data* (co-data; Neuenschwander et al., 2016; Van Nee et al., 2021).

Let $\mathbf{y} = [y_1, \ldots, y_n]^\intercal$ be the n-dimensional response vector and $\boldsymbol{\theta}$ the (possibly) multivariate vector of parameters of interest. From Bayes' rule, the posterior distribution $\pi(\boldsymbol{\theta} \mid \mathbf{y})$ can be evaluated as

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{l(\mathbf{y} \mid \boldsymbol{\theta})\, \pi(\boldsymbol{\theta})}{p(\mathbf{y})},$$

where $l(\mathbf{y} \mid \boldsymbol{\theta})$ is the likelihood function, $\pi(\boldsymbol{\theta})$ is the prior distribution and $p(\mathbf{y})$ is the normalizing constant. The prior's and likelihood's effect on the posterior density is shown in Figure 1.1.

Density $p(\mathbf{y})$ usually involves multiple integrals and is unknown. For this reason, exact posterior inference becomes intractable. One way to overcome this issue is to assume a conjugate prior, which allows the posterior to follow the

same distribution as the prior. Well-known distributions are usually implemented and $p(\mathbf{y})$ is available in closed form, therefore the posterior inference can be easily achieved. However, this can only be done with the most trivial probabilistic models and the posterior quantities generally can not be directly inferred, requiring some form of approximation in most cases.

1.1.1 MCMC methods

Markov Chain Monte Carlo (MCMC) methods represent a class of sampling algorithms for approximating intractable integrals. They combine Markov chain methodologies to randomly sample from high-dimensional distributions and Monte Carlo integration. A detailed overview of MCMC methods can be found in Robert and Casella (2004).

Their main goal is to overcome Monte Carlo problems. Typically, Bayesian inference aims at estimating a function of the parameters of the form

$$\mathbb{E}_{\pi(\boldsymbol{\theta}|\mathbf{y})}\left[g\left(\boldsymbol{\theta}\right)\right] = \int_{S_\theta} g\left(\boldsymbol{\theta}\right)\pi\left(\boldsymbol{\theta} \mid \mathbf{y}\right)d\boldsymbol{\theta}, \tag{1.1}$$

which usually does not admit an analytical solution. Monte Carlo integration draws random values from the target distribution $\pi\left(\boldsymbol{\theta} \mid \mathbf{y}\right)$ and approximates the integral in (1.1) as

$$\mathbb{E}_{\pi(\boldsymbol{\theta}|\mathbf{y})}\left[g\left(\boldsymbol{\theta}\right)\right] \approx \frac{1}{B}\sum_{b=1}^{B} g\left(\boldsymbol{\theta}^{(b)}\right), \tag{1.2}$$

where B is the number of samples. When B is sufficiently large, estimate (1.2) provides a consistent, unbiased and asymptotically normal estimator for $g\left(\boldsymbol{\theta}\right)$. This method, however, is not suited for high-dimensional problems: it assumes the independence between the samples drawn from the target distribution and requires techniques to easily generate these values, which is usually unrealistic given the high dimensionality of the problem. For these reasons, MCMC methods rely on Markov chains to randomly generate values from the target density in order to achieve a Monte Carlo approximation of the required integral. Markov chains provide a sampling scheme to sample from a distribution when its density is known up to a normalizing constant. They represent a stochastic process where each value only depends on the current state and not on the previous ones. More details about Markov properties are discussed in Meyn and Tweedie (1993) and Robert and Casella (2004), where the authors establish the results for the convergence of a Markov chain to its target density.

There exist many MCMC approaches for sampling from a distribution without directly requiring it. The most applied are the following two methods, which represent the baseline for modern generalizations introduced in literature:

- **Metropolis-Hasting algorithm** (Hastings, 1970): a flexible approach for constructing a Markov chain is the Metropolis-Hastings algorithm (MH). Sampling from the target density is achieved by proposing a new state of the chain at each iteration and evaluating the transition between states with a MH acceptance probability. Specifically, this method requires the introduction of a proposal density $q\left(\boldsymbol{\theta} \rightarrow \boldsymbol{\theta}'\right)$ to sample new values $\boldsymbol{\theta}'$. Let $\pi\left(\boldsymbol{\theta} \mid \mathbf{y}\right)$ be the target density, new state $\boldsymbol{\theta}'$ is sampled from $q\left(\boldsymbol{\theta} \rightarrow \boldsymbol{\theta}'\right)$ and accepted with the following probabilistic acceptance criterion;

$$\alpha = \min\left\{1, \frac{\pi\left(\boldsymbol{\theta}' \mid \mathbf{y}\right) q\left(\boldsymbol{\theta}' \rightarrow \boldsymbol{\theta}\right)}{\pi\left(\boldsymbol{\theta} \mid \mathbf{y}\right) q\left(\boldsymbol{\theta} \rightarrow \boldsymbol{\theta}'\right)}\right\}.$$

  In order for the MH algorithm to converge to the target density, proposal density q must be able to generate all the values belonging to the support of π, that is, $q\left(\boldsymbol{\theta} \rightarrow \boldsymbol{\theta}'\right) > 0$ for every $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{S}_{\pi}$ (Roberts and Smith, 1994). A good proposal distribution leads to faster convergence of the algorithm. The tuning of q, however, is not straightforward, as many different choices can be made and each of them leads to different results. Ideally, a good proposal should provide high acceptance probabilities for the proposed states and its covariance structure should reflect that of the target density. A main advantage of MH algorithm is its ability to avoid getting stuck at a local mode of the target density by occasionally accepting new values with lower acceptance probability. This method is mainly used when the conditional distributions of the parameters are not available or are either tricky or inefficient to sample from;

- **Gibbs sampler** (Geman and Geman, 1984; Casella and George, 1992): Gibbs sampling provides a sampling approach to construct a Markov chain by iteratively updating one component θ_k at a time. Specifically, each component is sampled from its full-conditional distribution $\pi\left(\theta_k \mid \boldsymbol{\theta}_{(-k)}, \mathbf{y}\right)$. This approach can be seen as particular case of the MH algorithm, where the proposal density q is the full-conditional distribution. For this reason, there is not need of tuning of the proposal density and each new state of the chain is accepted with probability equal to 1. Contrary to the MH algorithm, however, Gibbs sampling is more prone to being stuck at local modes and typically suffers from low convergence rate because of the local updates of the parameters. This method is mainly applied in a conjugated framework, where the full-conditionals are known and easy to sample from. More details on Gibbs sampling and its convergence properties can be found in Casella and George (1992); Roberts and Polson (1994); Roberts and Smith (1994).

1.1.2 Variational Bayes approximation

When the number of covariates is huge MCMC sampling methods become computationally infeasible. *Variational inference* (VI) is a deterministic optimization approach to approximate the target density $\pi(\boldsymbol{\theta} \mid \mathbf{y})$ with a variational distribution $q(\boldsymbol{\theta})$ and considers Bayesian inference as an optimization problem (Salimans et al., 2015; Lee, 2022). Note that this approach, contrary to MCMC methods that provide samples from the target distribution, gives mean point estimates of the quantity of interest. Moreover, the standard deviations are usually underestimated (MacKay et al., 2003; Wang and Titterington, 2005; Turner and Sahani, 2011; Giordano et al., 2017), leading to a trickier and less accurate posterior inference. This lack of accuracy, however, does not necessarily affect the performance of this methodology (Blei and Jordan, 2006).

The goal is to find $q(\boldsymbol{\theta})$ that minimizes the Kullback-Leibler divergence (KL) between the target density and the variational distribution. Taking the expectation with respect to q, the KL divergence is

$$\begin{aligned}
\mathrm{KL}(q\|\pi) &= \mathbb{E}_q\left[\log \frac{q(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta} \mid \mathbf{y})}\right] \\
&= \mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right] - \mathbb{E}_q\left[\log \pi(\boldsymbol{\theta} \mid \mathbf{y})\right] \\
&= \mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right] - \mathbb{E}_q\left[\log \pi(\boldsymbol{\theta}, \mathbf{y})\right] + \log p(\mathbf{y}), \qquad (1.3)
\end{aligned}$$

which depends on $p(\mathbf{y})$, the (usually) unknown marginal distribution of $\mathbf{y}$. The minimization problem in (1.3) is eventually reduced to the maximization of the Variational lower bound, which is defined as $\mathcal{L} = \mathbb{E}_q\left[\log \pi(\boldsymbol{\theta}, \mathbf{y})\right] - \mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right]$. Following the non-negativity property of the KL divergence, it yields $\log p(\mathbf{y}) \geq \mathcal{L}$. Thus, minimizing the KL divergence between q and p is equivalent to maximizing the lower bound $\mathcal{L}$.

A common factorization for $q(\boldsymbol{\theta})$ is the so-called *mean-field Variational approximation* (Jordan et al., 1999; Beal, 2003), which is a compromise between computational tractability and accuracy of the performances. The variational family $q(\boldsymbol{\theta})$ is assumed to be the product of independent marginal variational factors $q_k(\theta_k)$, $k = 1, \ldots, K$, and is defined as

$$q(\boldsymbol{\theta}) = \prod_{k=1}^{K} q_k(\theta_k).$$

The *Coordinate Ascent Variational Inference* algorithm (CAVI) (Bishop and Nasrabadi, 2006; Blei et al., 2017) is a useful tool for efficiently solving the optimization problem explained above. Until convergence of the lower bound $\mathcal{L}$, the CAVI algorithm iteratively updates the parameters of the variational factors $q_k(\theta_k)$, $k = 1, \ldots, K$,

based on prior distributions' hyperparameters and the current expectations of factor $q_{-k}(\theta_{-k})$, considered fixed. This way the model is able to account for non-linear dependencies among the parameters. Formally, the variational factors are updated as

$$q^{\star}(\theta_k) = \text{argmin}_q \, \text{KL}\left(q(\theta_k) \cdot \prod_{h \neq k} q^{\star}(\theta_h) \,\|\pi(\boldsymbol{\theta} \mid \mathbf{y})\right), \tag{1.4}$$

where the superscript $\star$ indicates that the corresponding factor has been updated (Lee, 2022). Under the mean field approximation, where the components are assumed to be independent, the optimal solution of (1.4) is

$$q^{\star}(\theta_k) \propto \exp\left\{\mathbb{E}_{q_{-k}}[\log \pi(\theta_k \mid \theta_{-k}, \mathbf{y})]\right\}.$$

While the assumption of independence between factors is particularly strict, the CAVI algorithm provides a flexible approach and ensures the convergence to a local optimum (Blei et al., 2017). Note that, when working with exponential families in a conjugated framework, variational factor $q(\theta_k)$ has the same kernel of the full-conditional distribution $\pi(\theta_k \mid \theta_{-k}, \mathbf{y})$.

1.2 Bayesian model selection

This thesis deals with Bayesian shrinkage models under two different frameworks: generalized linear regression and graphical models. In this section, a brief explanation is given of how variable selection with shrinkage priors is achieved under these circumstances, alongside a short introduction to the most common prior assumptions.

Generalized Linear Models (GLM). These models represent a generalization of the linear regression and allow a linear model to be related to the response variable through a *link function*. This way, different types of outcomes (continuous, binary, count data) can be modelled. Let $\mathbf{y}$ be the n-dimensional response vector and $\mathbf{X}$ the $n \times p$ design matrix. The expected value of $\mathbf{y}$ is related to the linear predictor $\mathbf{X}\boldsymbol{\beta}$, where $\boldsymbol{\beta}$ is the p-dimensional regression parameter vector, through an invertible link function h. Shrinkage is applied to vector $\boldsymbol{\beta}$ element-wise by decomposing the prior variances in a global scale τ^2 and a local scale λ_j^2, $j = 1, \ldots, p$. The goal is to find a sparse solution for vector $\boldsymbol{\beta}$. The general hypotheses

of the model are

$$\begin{aligned} Y_i \mid \mathbf{x_i}, \boldsymbol{\beta} &\overset{ind}{\sim} p\left(Y_i \mid \mathbf{x_i}, \boldsymbol{\beta}\right), \\ \mathbb{E}_{Y_i|\mathbf{x_i},\boldsymbol{\beta}}\left[Y_i\right] &= h^{-1}\left(\mathbf{x_i}^\intercal \boldsymbol{\beta}\right), \quad i = 1, \ldots, n, \\ \beta_j \mid \tau^2, \lambda_j^2 &\sim \mathcal{N}\left(0, \tau^2\lambda_j^2\right), \\ \lambda_j^2 &\sim \pi\left(\lambda_j^2\right), \quad j = 1, \ldots, p, \\ \tau^2 &\sim \pi\left(\tau^2\right). \end{aligned}$$

When h is the identity function the model reduces to the ordinary linear regression. Parameter λ_j^2 guides the amount of shrinkage for regression parameter β_j: when $\lambda_j^2 \to \infty$ then no shrinkage is applied, whereas the coefficient β_j is shrunk towards 0 when $\lambda_j^2 \to 0$.

Graphical Models. Graphical models are a useful tool for network analysis and their goal is to infer the dependencies between a set of variables. A network (or graph) represents a collection of variables $\mathbf{x} = [x_1, \ldots, x_p]^\intercal$ with a set of vertex $\mathcal{V} = \{1, \ldots, p\}$ and it encodes conditional dependencies by a set of edges $\mathcal{E} \coloneqq \left\{(s,k) \in \mathcal{E} \leftrightarrow x_s \not\perp\!\!\!\perp x_k \mid \mathbf{x}_{\mathcal{V}\setminus\{s,k\}}\right\}$. That is, if pair (s,k) does not belong to $\mathcal{E}$ then x_s and x_k are conditionally independent with respect to the other variables. There exist many different types of graphs, however this thesis only focuses on undirected networks. Graphical models use graph structure to model the dependencies between variables. A common class of graphical models is the so-called Gaussian Graphical models (GGM; Wang, 2012, 2015; Li et al., 2019), which relies on the multivariate Gaussian likelihood

$$\mathbf{x}_i \mid \boldsymbol{\Omega} \sim \mathcal{N}_p\left(\mathbf{0}_p, \boldsymbol{\Omega}^{-1}\right), \quad i = 1, \ldots, n,$$

where $\boldsymbol{\Omega} \equiv \{\omega_{sk}\}_{(p\times p)}$ denotes the $p \times p$ inverse covariance matrix, also called *precision matrix*. There is a one-to-one correspondence between the zero pattern in a precision matrix and an undirected graph. This property can be exploited to learn conditional independencies between variables. Specifically, it can be shown that under the Gaussian assumption, it yields $\omega_{sk} = 0$ if and only if variables s and k are conditionally independent with respect to the other variables (Dempster, 1972). Therefore, the goal is the estimation of non-zero entries in $\boldsymbol{\Omega}$ under the assumption of sparseness. Edge selection is performed by assuming a Gaussian prior distribution for entries ω_{sk} and decomposing their prior variance in a global scale τ^2 and a local scale λ_{sk}^2. Specifically,

$$\begin{aligned} \omega_{sk} \mid \tau^2, \lambda_{sk}^2 &\sim \mathcal{N}\left(0, \tau^2\lambda_{sk}^2\right), \\ \lambda_{sk}^2 &\sim \pi\left(\lambda_{sk}^2\right), \quad s < k, \quad k = 1, \ldots, p, \\ \tau^2 &\sim \pi\left(\tau^2\right). \end{aligned}$$

As for the shrinkage in GLM, the local variances λ_{sk}^2 guide the amount of shrinkage imposed on edge ω_{sk}. Graphical models estimation is computationally hard, as the number of parameters to be estimated is of order $O(p^2)$, which becomes particularly challenging in high-dimensional settings. Moreover, the estimation of precision matrices is particularly difficult since they are constrained to the cone of symmetric positive-definite matrices, which implies restrictive conditions on the sampling scheme.

1.2.1 Shrinkage and selection priors

Different types of shrinkage can be applied based on the prior assumptions on local scale parameters λ_j^2. Typically, most shrinkage priors share common properties such as zero-mean and symmetry around zero. The main difference is the induced amount of shrinkage on parameter β_j, which can be inferred by integrating out local scale λ_j^2: the resulting mass probability around zero and the tails of the induced prior distribution reflect the imposed level of shrinkage. Below an overview of the most famous shrinkage priors is presented. See Van Erp et al. (2019) for a detailed list of the existing shrinkage methods.

- **Ridge penalty**
  One of the first and most simplistic attempts at shrinkage variable selection is the Ridge penalty (Hsiang, 1975), which corresponds to the l_2 penalization in Hoerl and Kennard (1970). This shrinkage method was originally introduced to deal with multicollinearity between variables. It assumes the same common prior variance for all the regression parameters β_j. Specifically,

  $$\beta_j \mid \lambda^2 \sim \mathcal{N}\left(0, \lambda^2\right), \quad j = 1, \ldots, p.$$

  Prior variance λ^2 can be treated either as a fixed parameter or as an unknown quantity. For this latter case, common choices for $\pi(\lambda^2)$ are the inverse-Gamma (Tipping, 2001) and the scale inverse-χ^2 (De los Campos et al., 2009; Montesinos López et al., 2022).

- **Spike-and-slab prior**
  The so-called spike-and-slab prior (Mitchell and Beauchamp, 1988; George and McCulloch, 1993) is a mixture of two components, a spike component with mass concentrated around zero and a slab component with high variance v_1^2. The predictors included in the model are assigned the slab prior component, whereas the variables excluded are assigned the spike component. It is different from the continuous scale mixture of Normal priors, however a formulation that connects this prior to the other shown here is

presented in Ishwaran and Rao (2005). A common specification is the Dirac spike-and-slab prior, with prior assumptions

$$\begin{aligned}\beta_j \mid \gamma_j, v_1^2 &\sim \gamma_j \mathcal{N}\left(0, v_1^2\right) + (1-\gamma_j)\,\delta_0\left(\beta_j\right), \\ \gamma_j \mid \phi_j &\sim \mathsf{Bern}\left(\phi_j\right), \\ \phi_j &\sim \mathsf{Beta}(a,b), \quad j = 1, \ldots, p,\end{aligned}$$

where $\delta_0(\cdot)$ denotes a Dirac's Delta distribution with mass probability at 0. Integrating γ_j out yields

$$\beta_j \mid \phi_j, v_1^2 \sim \phi_j \mathcal{N}\left(0, v_1^2\right) + (1-\phi_j)\,\delta_0\left(\beta_j\right), \quad j = 1, \ldots, p,$$

which is a mixture distribution with mixing probabilities ϕ_j and $1-\phi_j$. Another common choice for the spike component is a Normal distribution with low variance $\mathcal{N}(0, v_0^2)$, with $v_0^2 << v_1^2$ (George and McCulloch, 1993; Van Erp et al., 2019). Prior variances v_0^2 and v_1^2 can be either treated as fixed parameters or considered unknown and assigned an Inverse-Gamma prior distribution.

- **t-Student prior**
  An extension of the Ridge shrinkage prior is to assume a specific local variance for each regression parameter β_j (Meuwissen et al., 2001; Griffin and Brown, 2005). The prior assumptions are

$$\begin{aligned}\beta_j \mid \tau^2, \lambda_j^2 &\sim \mathcal{N}\left(0, \tau^2\lambda_j^2\right), \\ \lambda_j^2 &\sim \mathcal{IG}\left(\frac{\nu}{2}, \frac{\nu}{2\zeta}\right), \quad j = 1, \ldots, p.\end{aligned}$$

  The induced prior distribution for β_j after integrating λ_j^2 out is

$$\beta_j \mid \tau^2, \nu, \zeta \sim t_\nu\left(0, \frac{\tau^2}{\zeta}\right), \quad j = 1, \ldots, p,$$

  where $t_\nu\left(0, \tau^2/\zeta\right)$ denotes the Student-t distribution with ν degrees of freedom, centered at 0 and scale parameter τ^2/ζ. When $\nu = 1$ the induced prior reduces to a Cauchy distribution. Compared to Ridge penalization, the Student-t distribution shows heavier tails, thus providing a sparser solution for $\boldsymbol{\beta}$.

- **LASSO penalty**
  The Bayesian version of the LASSO regression (l_1 penalty) was proposed

by Park and Casella (2008). The model assumes the following hierarchical structure

$$\begin{aligned}\beta_j \mid \tau^2, \lambda_j^2 &\sim \mathcal{N}\left(0, \tau^2\lambda_j^2\right), \\ \lambda_j^2 &\sim \mathsf{Exp}\left(\frac{\nu^2}{2}\right), \quad j = 1, \ldots, p.\end{aligned}$$

Integrating λ_j^2 out results in the following induced prior

$$\beta_j \mid \tau^2, \nu^2 \sim \mathsf{DE}\left(0, \frac{\tau}{\nu}\right), \quad j = 1, \ldots, p,$$

where DE denotes the double-exponential (or Laplace) distribution. Bayesian LASSO presents some differences when compared to its penalized counterpart (Tibshirani, 1996): first, the latter provides a variable selection method, whereas the former does not set coefficients to zero, requiring a posterior selection process; second, penalized LASSO can not select more predictors than observations, which can be problematic when $n > p$; the Bayesian version, instead, is able to overcome this issue; third, Bayesian LASSO does not follow the oracle property (Polson et al., 2011), whereas the penalized version follows it under some stringent conditions (Fan and Li, 2001; Zou, 2006). Finally, both methods suffer from oversrhinkage of large effects (Polson and Scott, 2011; Polson et al., 2011).

- **Elastic-net penalty**
  The Bayesian elastic-net was introduced by Li and Lin (2010). It relies on the following scale mixture of Normals assumption:

$$\begin{aligned}\beta_j \mid \tau_2, \lambda_j &\sim \mathcal{N}\left(0, \left(\tau_2 \frac{\lambda_j}{\lambda_j - 1}\right)^{-1}\right), \\ \lambda_j \mid \tau_2, \tau_1 &\sim \mathcal{G}_{(1,\infty)}\left(\frac{1}{2}, \frac{8\tau_2}{\tau_1^2}\right), \quad j = 1, \ldots, p,\end{aligned}$$

  where $\mathcal{G}_{(1,\infty)}$ denotes a Gamma distribution left-truncated at 1. The induced prior distribution on parameter β_j is

$$\beta_j \mid \tau_2, \tau_1 \propto \exp\left\{-\frac{1}{2}\left(\tau_1|\beta_j| + \tau_2\beta_j^2\right)\right\}, \quad j = 1, \ldots, p.$$

  Penalty parameters τ_1 and τ_2 determine the amount of LASSO and Ridge shrinkage, respectively. Although the estimation of these parameters leads to overshrinkage in the penalized version, the Bayesian elastic-net is able to overcome such issue by estimating them simultaneously.

- **Horseshoe prior**
  A modern shrinkage prior is the Horseshoe prior proposed by Carvalho et al. (2010), which assumes a positive Half-Cauchy distribution for the local variances. Formally,

$$\beta_j \mid \tau^2, \lambda_j \sim \mathcal{N}\left(0, \tau^2\lambda_j^2\right),$$
$$\lambda_j \sim \mathcal{C}^+(0,1), \quad j = 1, \ldots, p.$$

  The induced prior distribution on β_j is not analytically tractable. The shrinkage behaviour of this prior can be deduced by observing the posterior distribution of shrinkage coefficient $\kappa_j^2 = \left(1 + \lambda_j^2\right)^{-1}$, which shows a horseshoe form. This leads to large effects assuming values similar to their OLS estimates, whereas small effects are heavily shrunk towards zero. Since the local variances λ_j^2 can not be easily sampled from their full-conditional distributions, a Gibbs sampler can be implemented by augmenting the model as shown in Makalic and Schmidt (2016). An improved version of the Horseshoe prior is the regularized Horseshoe in Piironen and Vehtari (2017), where the authors give insights on the choice of the global scale prior distribution and overcome the problem related to the amount of regularization for the largest coefficients, which can be problematic with weakly identified parameters in the ordinary Horseshoe setting.

1.3 Outline and contributions

The main goal of this thesis is to provide efficient and reliable Bayesian statistical methods for the analysis of high-dimensional data. Different frameworks and hypothesis are considered, resulting in three independent projects. In order to provide efficient tools, the algorithms are based on fast computational approaches and are all implemented in C++ with Rcpp package for R software.

Chapter 2 addresses the problem of variable selection for sparse high-dimensional linear regression with Gaussian errors. A new class of trans-dimensional MCMC algorithms (Green, 1995; Fan et al., 2009) is introduced. In particular, a multiple-try MH scheme (Liu et al., 2000; Martino et al., 2012; Casarin et al., 2013) based on adaptive mixture of proposal distributions is discussed. The model relies on a Dirac spike-and-slab prior (George and McCulloch, 1993) where at each iteration a new model is proposed and accepted with a generalized MH step. The target density of the algorithm is efficiently updated by exploiting a new class of computational methods based on the thinQR decomposition.

Chapter 3 introduces a novel multivariate shrinkage prior for the estimation of multiple similar networks. The model combines the approach of Peterson et al.

(2020) and the Horseshoe prior (Carvalho et al., 2010) with the goal of inferring correlated (sparse) precision matrices. This approach represents an extension of the Graphical Horseshoe (Li et al., 2019) and scales well up to hundreds of variables. Finally, a novel approach for posterior edge selection based on model cuts (Zigler et al., 2013; Plummer, 2015) is proposed.

Chapter 4 presents a flexible way to handle co-data variables in high-dimensional regression for both binary and continuous outcome. The method relies on an informative version of the Horseshoe prior (Carvalho et al., 2010) based on the regression of the local variances on the co-data, following Van Nee et al. (2021). Both Gibbs sampler and Variational approximation are implemented for the model estimation. In particular, the former makes use of the method in Bhattacharya et al. (2016) for sampling the regression parameters from a multivariate Gaussian distribution and the latter relies on the computational methods presented in Münch et al. (2019). Therefore, both provide algorithms with $O(n^2p)$ operations, suited for high-dimensional problems.

Chapter 5 ends the thesis with several final discussions and comments. In particular, insights on future extensions of the presented models are debated.

Chapter 2

Fast Bayesian model selection for high-dimensional linear regression models

2.1 Introduction

Mixture priors for Bayesian variable selection in univariate linear regression models with Gaussian errors were originally proposed by Leamer (1978) and Mitchell and Beauchamp (1988) and made popular by the spike-and-slab approach of George and McCulloch (1993, 1997). Similar approaches have been proposed by Carlin and Chib (1995), Clyde et al. (1996), Geweke (1996), Smith and Kohn (1996), Raftery et al. (1997), Liang et al. (2001) and Dellaportas et al. (2002). Model and variable selection methods have seen a renewed interest nowadays due to the availability of huge datasets. Ročková and George (2014) propose the Expectation-Maximization algorithm for variable selection computationally faster than the Gibbs sampler, while Ročková and George (2018) extend the spike-and-slab approach to Laplace mixture components, to allow variable selection and shrinkage. Computationally efficient methods for the exploration of the space of competing models have been introduced by the shotgun procedure of Hans et al. (2007). Hans (2009, 2011) further extend the Bayesian model selection via Dirac spike-and-slab prior to the case of Laplace mixture components and the Elastic-net prior of Li and Lin (2010). Reviews of special features of the selection priors and on computational aspects can be found in Chipman et al. (2001), Clyde and George (2004), Ishwaran and Rao (2005), O'Hara and Sillanpää (2009), Heinze et al. (2018), Narisetty (2020), Forte et al. (2018) and in the recent book of Tadesse and Vannucci (2021).

However, when the number of covariates is large, the complete model enumer-

ation prevents the full exploration of the space of competing, Markov chain Monte Carlo (MCMC) methods provides a viable and feasible alternative to the Gibbs sampler. In the context of variable selection, trans-dimensional MCMC methods (see, e.g. Green, 1995; Fan and Sisson, 2011; Hastie and Green, 2012) quickly and efficiently explore the space of competing models looking for optimal solutions, i.e. models with high posterior probability, see George and McCulloch (1997). A popular approach is the Metropolis scheme (MC3), proposed by Madigan et al. (1995) in the context of model selection for discrete graphical models and subsequently adapted to variable selection, see Raftery et al. (1997) and Brown et al. (1998, 2002), among others. Improved MCMC schemes have been proposed to achieve an even faster exploration of the posterior space, see, for example, the shotgun algorithm of Hans et al. (2007) and the evolutionary Monte Carlo schemes combined with parallel tempering proposed by Bottolo and Richardson (2010), Bottolo et al. (2011).

Within the regression context, reversible jump (RJ, hereafter) algorithms have been previously proposed, for example, by Petralias and Dellaportas (2013) and for generalised linear models by Papathomas et al. (2011). As any other Metropolis schemes, the RJ-type proposals has the major disadvantage of performing a good "local" exploration of the posterior distribution, thereby slowing down the convergence speed as the dimension of the problem increases. Improving the mixing and the rate of convergence of the chain can be achieved by means of multiple-try Metropolis MCMC (MTM, hereafter) introduced by Liu et al. (2000) as an extended version of the classical Metropolis-Hastings scheme that allows to select the new state of the chain among several alternatives. MTM methods have been widely studied and generalized, with different versions based mainly on different choices for the trial proposals. The basic approach allows to propose multiple states of the chain independently from the same distribution Liu et al. (2000), whereas more complex versions involve correlated trials (Craiu and Lemieux, 2007; Bédard et al., 2012) or different independent proposals (Casarin et al., 2013). All of the cited papers assume the number of trials to be fixed in advance. In Martino and Louzada (2017) the authors study the mixing properties of MTM algorithm when the proposal distribution is a random walk: they state that large values of K do not always improve the rate of convergence of the MCMC and propose different solutions. To this aim, Chang et al. (2022) try to calibrate the optimal number of trials. For a general overview of MTM methods, we refer to Martino (2018). A reversible jump MTM method for Bayesian model selection framework is proposed in Pandolfi et al. (2010, 2014), where the authors rely on multiple trans-dimensional moves to efficiently explore the space of models.

Other interesting developments have focused on adaptive methods for the optimization of parametric transition probabilities of MCMC algorithms, with the

aim of improving efficiency and mixing of the Metropolis schemes. Early adaptation schemes can be found in Gilks et al. (1998) and Haario et al. (2001), where the authors propose the tuning of the transition kernel based on the previous states visited by the chain. However, adaptation can lead to the loss of ergodicity of the chain (see, e.g. Andrieu and Moulines, 2006; Roberts and Rosenthal, 2007; Andrieu and Thoms, 2008; Craiu et al., 2015, for the theoretical properties of adaptive MCMC algorithms). In Andrieu and Moulines (2006) and Andrieu and Thoms (2008) the authors provide a general guidances on building adaptive MCMC schemes and discuss a Metropolis scheme where the proposal density is a mixture of distributions that belong to the family of exponential distributions. Other applications dealing with adaptive proposal of Gaussian mixture can be found in Douc et al. (2007), Ji and Schmidler (2013), Feng and Li (2015) and Maire et al. (2019). Within the MTM framework, Yang et al. (2019) and Fontaine and Bédard (2022) propose adaptive versions of the MTM algorithm.

Here, we introduce a novel trans-dimensional adaptive MTM algorithm that exhaustively explores the target distribution. In particular, our approach considers parallel jumps between models that include different predictors, while Lamnisos et al. (2009) and Pandolfi et al. (2010, 2014) only consider jump between models that differ only by a single variable. Similarly to the shotgun stochastic algorithm of Hans et al. (2007), our model forces the chain to explore the model space in the neighborhood of high-probability models. We rely on a mixture of proposal distributions, where each component is related to a different degree of divergence from the current model, i.e. the number of included or excluded predictors. The importance of each component is calibrated to achieve optimal jumps, that is, the mixing probabilities of the mixture associated to the different proposals are estimated adaptively in order to ensure that the algorithm explores models that provide high scores of the target density.

Finally, in Appendix 2.D, we present a new class of algebraic algorithms based on the thinQR decomposition for the efficient update of the posterior covariance matrix under Dirac's spike-and-slab priors. These updating algorithms, alongside the methods discussed in Section 2.4 for the efficient evaluation of the target density with $\mathcal{O}(p)$ operations, make our model one of the fastest Bayesian approaches for model selection in high-dimensional linear regression with Gaussian errors.

The rest of this chapter is organized as follows. In Section 2.2, we introduce the model and prior specifications, while in Section 2.3, we outline the MCMC algorithm and discusses the posterior sampling details. In Section 2.4, we assess the problem of efficiently evaluating the target distribution of the algorithm. Simulation studies and applications to real datasets are presented in Section 2.5 and 2.6, respectively. Final discussions and comments are in Section 2.7.

2.2 Model specification

Let $\mathbf{y} \in \mathbb{R}^n$ be the n-dimensional response vector and $\mathbf{X} \in \mathbb{R}^{n \times p}$ be the $n \times p$ design matrix. We consider the following univariate Gaussian linear regression model for the continuous outcome y_i, $i = 1, \ldots, n$

$$\begin{aligned} y_i &= \mathbf{x}_i^\intercal \boldsymbol{\beta} + \varepsilon_i, \\ \varepsilon_i &\sim \mathsf{N}\big(0, \sigma^2\big), \quad i = 1, \ldots, n, \end{aligned} \tag{2.1}$$

where $\mathbf{x}_i = \big(x_{i,1}, \ldots, x_{i,p}\big)^\intercal \in \mathbb{R}^p$, is the set of p covariates related to the i-th observation and $\boldsymbol{\beta} = \big(\beta_1, \ldots, \beta_p\big)^\intercal \in \mathbb{R}^p$ is the p-dimensional vector of regression parameters. To induce sparse solutions for $\boldsymbol{\beta}$, we assume a Dirac spike-and-slab prior with the slab component's prior variance $v_1^2 \gg 0$ considered as a fixed hyperparameter (George and McCulloch, 1997). This approach relies on an auxiliary latent p-dimensional selection vector $\boldsymbol{\gamma} = \big(\gamma_1, \ldots, \gamma_p\big)^\intercal$, where $\gamma_j = 1$, if the j-th regressor is included in the model, and $\gamma_j = 0$ otherwise. Note that the complexity of regression model $\boldsymbol{\gamma}$ can be retrieved as $p_\gamma = \sum_{j=1}^p \gamma_j$. Let $\boldsymbol{\beta}_\gamma \in \mathbb{R}^{p_\gamma}$ be the vector consisting of all elements β_j for which $\gamma_j = 1$, $j = 1, \ldots, p$, and $\boldsymbol{\beta}_{-\gamma} = \boldsymbol{\beta} \setminus \boldsymbol{\beta}_\gamma$. Then the Dirac spike-and-slab hierarchical prior for the regression model in (2.1) is

$$\begin{aligned} \boldsymbol{\beta}_\gamma | \boldsymbol{\gamma}, \sigma^2 &\sim \mathsf{N}_{p_\gamma}\big(\beta_j | 0, \sigma^2 \boldsymbol{\Sigma}_{\beta_\gamma}\big), \\ \pi\big(\boldsymbol{\beta}_{-\gamma} | \boldsymbol{\gamma}\big) &= \prod_{j=1}^p \delta\big(\beta_j, 0\big)^{1-\gamma_j}, \\ \gamma_j &\sim \mathsf{Ber}\big(\phi\big), \quad j = 1, \ldots, p, \\ \phi &\sim \mathsf{Beta}\big(\xi, \varphi\big), \\ \sigma^2 &\sim \mathsf{IG}(\nu, \lambda), \end{aligned} \tag{2.2}$$

where $\delta\big(x, 0\big) = \mathbb{I}_{(0)}(x)$ denotes the Dirac function evaluated at zero and $\boldsymbol{\Sigma}_{\beta_\gamma} = v_1^2 \mathbf{I}_{p_\gamma}$ is the prior covariance matrix of $\boldsymbol{\beta}_\gamma$, with $\mathbf{I}_{p_\gamma}$ denoting the identity matrix of dimension p_γ. Independent Bernoulli priors on the γ_j's as specified in (2.2) with a Beta hyper-prior are used, for example, by Brown et al. (1998). As argued by Scott and Berger (2010), an attractive feature of these priors is that appropriate choices of ϕ, that depend on the number of covariates p, impose an a-priori multiplicity penalty.

2.3 Posterior inference

For the Gaussian linear regression model defined in (2.1), under the hierarchical Dirac spike-and-slab prior defined in (2.2), the joint posterior distribution is:

$$\pi\big(\boldsymbol{\beta},\boldsymbol{\gamma},\sigma^2,\phi|\mathbf{y},\mathbf{X}\big) \propto \sigma^{-(n+p_\gamma+2\nu+2)}|\boldsymbol{\Sigma}_{\beta_\gamma}|^{-1/2}\exp\left(-\frac{\widetilde{\boldsymbol{\varepsilon}}_\gamma^\intercal\widetilde{\boldsymbol{\varepsilon}}_\gamma+2\lambda}{2\sigma^2}\right)\pi(\boldsymbol{\gamma})\pi(\phi), \quad (2.3)$$

where $\widetilde{\boldsymbol{\varepsilon}}_\gamma$ is the n-dimensional vector of residuals for model $\boldsymbol{\gamma}$ defined as $\widetilde{\boldsymbol{\varepsilon}}_\gamma = \widetilde{\mathbf{y}} - \widetilde{\mathbf{X}}_\gamma\boldsymbol{\beta}_\gamma$, with

$$\widetilde{\mathbf{y}} = \begin{pmatrix}\mathbf{y}\\ 0_{p_\gamma}\end{pmatrix} \in \mathbb{R}^{n+p_\gamma} \quad \text{and} \quad \widetilde{\mathbf{X}}_\gamma = \begin{pmatrix}\mathbf{X}_\gamma\\ \boldsymbol{\Sigma}_{\beta_\gamma}^{-1/2}\end{pmatrix} \in \mathbb{R}^{(n+p_\gamma)\times p_\gamma}, \quad (2.4)$$

and $\mathbf{X}_\gamma \in \mathbb{R}^{n\times p_\gamma}$ is the $n\times p_\gamma$ matrix whose columns correspond to the components of $\boldsymbol{\beta}_\gamma$. The set of full-conditional distributions for the update of parameters $\boldsymbol{\beta}$, σ^2 and ϕ is

$$\begin{aligned}\boldsymbol{\beta}_\gamma|\mathbf{y},\mathbf{X},\boldsymbol{\gamma},\sigma^2 &\sim \mathsf{N}_{p_\gamma}\big(\boldsymbol{\Sigma}^\star_{\beta_\gamma}\mathbf{X}_\gamma^\intercal\mathbf{y},\sigma^2\boldsymbol{\Sigma}^\star_{\beta_\gamma}\big)\\ \sigma^2|\mathbf{y},\mathbf{X},\boldsymbol{\beta},\boldsymbol{\gamma} &\sim \mathsf{IG}\big(\nu+\frac{n+p_\gamma}{2},\lambda+\frac{\widetilde{\boldsymbol{\varepsilon}}_\gamma^\intercal\widetilde{\boldsymbol{\varepsilon}}_\gamma}{2}\big)\\ \phi|\boldsymbol{\gamma} &\sim \mathsf{Beta}\big(\xi+p_\gamma,\varphi+p-p_\gamma\big),\end{aligned} \quad (2.5)$$

where $\boldsymbol{\Sigma}^\star_{\beta_\gamma} = \big(\widetilde{\mathbf{X}}_\gamma^\intercal\widetilde{\mathbf{X}}_\gamma\big)^{-1} = \big(\mathbf{X}_\gamma^\intercal\mathbf{X}_\gamma+\boldsymbol{\Sigma}_{\beta_\gamma}^{-1}\big)^{-1}$. Integrating out $\boldsymbol{\beta}_\gamma$ and σ^2 from (2.3) yields the marginal posterior distribution of model indicator $\boldsymbol{\gamma}$ which is proportional to

$$\begin{aligned}m(\boldsymbol{\gamma}|\mathbf{y},\mathbf{X}) &\propto \ell(\boldsymbol{\gamma}|\mathbf{y},\mathbf{X})\pi(\boldsymbol{\gamma})\\ \ell(\boldsymbol{\gamma}|\mathbf{y},\mathbf{X}) &\propto |\widetilde{\mathbf{X}}_\gamma^\intercal\widetilde{\mathbf{X}}_\gamma|^{-1/2}|\boldsymbol{\Sigma}_{\beta_\gamma}|^{-1/2}\big(\lambda+\frac{S_\gamma^2}{2}\big)^{-(\nu+n/2)}\\ \pi(\boldsymbol{\gamma}) &= \binom{p}{p_\gamma}\phi^{p_\gamma}\big(1-\phi\big)^{p-p_\gamma},\end{aligned} \quad (2.6)$$

where $S_\gamma^2 = \mathbf{y}^\intercal\mathbf{y}-\mathbf{y}^\intercal\mathbf{X}_\gamma\big(\mathbf{X}_\gamma^\intercal\mathbf{X}_\gamma+\boldsymbol{\Sigma}_{\beta_\gamma}^{-1}\big)^{-1}\mathbf{X}_\gamma^\intercal\mathbf{y}$. Here, the goal is the investigation of the marginal posterior distribution $m\big(\boldsymbol{\gamma}|\mathbf{y},\mathbf{X}\big)$ defined in equation (2.6). However, the full exploration of the space of competing models and the complete model enumeration become infeasible when the number of covariates is moderately large. Therefore, we rely on trans-dimensional MCMC methods (Green, 1995; Fan and Sisson, 2011) to sample the model indicators from the target distribution. Specifically, in this Section, we present different MH schemes to explore $m\big(\boldsymbol{\gamma}|\mathbf{y},\mathbf{X}\big)$. We rely on the thinQR updating methods introduced in Section 2.4 and Appendix 2.D to efficiently update the design matrix and the target distribution of model indicator $\boldsymbol{\gamma}$ when one or more predictors are included or excluded from the set defining the current regression model.

2.3.1 Reversible jump

The first and most straightforward trans-dimensional method is an ordinary reversible-jump MCMC. At each iteration the transition from current model $\boldsymbol{\gamma}$ to a new model $\boldsymbol{\gamma}'$ is evaluated with a MH acceptance probability. Specifically, we consider a new model $\boldsymbol{\gamma}'$ that differs from current model $\boldsymbol{\gamma}$ by the inclusion/exclusion of one covariate. This can be achieved by sampling $\boldsymbol{\gamma}'$ from the following proposal distribution (Lamnisos et al., 2009)

$$q(\boldsymbol{\gamma}'|\boldsymbol{\gamma}) = \frac{1}{p}, \quad \text{if} \quad \sum_{j=1}^{p} |\gamma'_j - \gamma_j| = 1, \tag{2.7}$$

which is symmetric in $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}'$. This way all the models with dimensionality $(p_\gamma - 1)$ or $(p_\gamma + 1)$ are taken into account with the same probability. Sampling a new model $\boldsymbol{\gamma}'$ from (2.7) can be done by randomly selecting a predictor ι from the set $\{1, \ldots, p\}$ with uniform probabilities; variable $\mathbf{x}_\iota$ is then added to $\boldsymbol{\gamma}'$ if not included in $\boldsymbol{\gamma}$ (i.e. $\gamma_\iota = 0$), whereas $\boldsymbol{\gamma}'$ is constructed by deleting $\mathbf{x}_\iota$ from the current regression model, otherwise (i.e. $\gamma_\iota = 1$). Because of the symmetry of proposal distribution $q(\boldsymbol{\gamma}'|\boldsymbol{\gamma})$, i.e. $q(\boldsymbol{\gamma}|\boldsymbol{\gamma}')/q(\boldsymbol{\gamma}'|\boldsymbol{\gamma}) = 1$, the MH acceptance probability for new model indicator $\boldsymbol{\gamma}'$ is

$$\alpha_{\mathsf{RJ}}(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = \min\left\{1, \frac{m(\boldsymbol{\gamma}'|\mathbf{y}, \mathbf{X})}{m(\boldsymbol{\gamma}|\mathbf{y}, \mathbf{X})}\right\}.$$

The transition from model $\boldsymbol{\gamma}$ to $\boldsymbol{\gamma}'$ means updating the target marginal posterior distribution after the addition or deletion of a column in the design matrix. Specifically, the method requires the computation of the posterior of variance-covariance matrix $\boldsymbol{\Sigma}^{\star}_{\beta_{\gamma'}} = \left(\widetilde{\mathbf{X}}^{\intercal}_{\gamma'}\widetilde{\mathbf{X}}_{\gamma'}\right)^{-1}$, with $\widetilde{\mathbf{X}}_{\gamma'} \in \mathbb{R}^{(n+p_{\gamma'})\times p_{\gamma'}}$ and $p_{\gamma'} = p_\gamma \pm 1$, and the quantity $S^2_{\gamma'} = \mathbf{y}^{\intercal}\mathbf{y} - \mathbf{y}^{\intercal}\mathbf{X}_{\gamma'}\left(\mathbf{X}^{\intercal}_{\gamma'}\mathbf{X}_{\gamma'} + \boldsymbol{\Sigma}^{-1}_{\beta_{\gamma'}}\right)^{-1}\mathbf{X}^{\intercal}_{\gamma'}\mathbf{y}$. The thinQR updating methods discussed in Section 2.4 can be applied to efficiently achieve this. The RJ algorithm is shown in Algorithm 1.

The jumps to models that only differ from the current model by the inclusion or exclusion of one predictor do not allow a fast and efficient global exploration of the space of competing models (see, e.g. Hans et al., 2007; Lamnisos et al., 2009). In order to overcome this issue, in the following subsections, we discuss two generalizations of the RJ method based on multiple-try approaches (Liu et al., 2000) to allow the transition to models that differ by more than one variable and further improve the flexibility of the model.

Algorithm 1: RJ algorithm

Input: $B \in \mathbb{N}$, $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{X} \in \mathbb{R}^{n \times p}$, $(\nu, \lambda) \in \mathbb{R}_+^2$, $(\xi, \varphi) \in \mathbb{R}_+^2$ and $\upsilon_1 \in \mathbb{R}^+$;

Initialization: sample $\phi^{(0)}$ and $\boldsymbol{\gamma}^{(0)}$ from their prior distributions;

for $(b = 1, 2, \ldots, B)$ **do**

- Sample $\iota \sim \mathsf{U}(\{1, 2, \ldots, p\})$;
- Set $\boldsymbol{\gamma}'_\iota = 1 - \boldsymbol{\gamma}_\iota^{(b-1)}$ and compute $m(\boldsymbol{\gamma}'|\mathbf{y}, \mathbf{X})$;
- Compute the MH acceptance probability:

$$\alpha(\boldsymbol{\gamma}^{(b-1)}, \boldsymbol{\gamma}') = \min\left\{1, \frac{m(\boldsymbol{\gamma}'|\mathbf{y}, \mathbf{X})}{m(\boldsymbol{\gamma}^{(b-1)}|\mathbf{y}, \mathbf{X})}\right\},$$

  where $m(\boldsymbol{\gamma}^{(b-1)}|\mathbf{y}, \mathbf{X})$ is defined in (2.6);
- With probability $\alpha(\boldsymbol{\gamma}^{(b-1)}, \boldsymbol{\gamma}')$ set $\boldsymbol{\gamma}^{(b)} = \boldsymbol{\gamma}'$, otherwise set $\boldsymbol{\gamma}^{(b)} = \boldsymbol{\gamma}^{(b-1)}$;
- Sample $\phi^{(b)} \sim \mathsf{Beta}(\xi + p_\gamma, \varphi + p - p_\gamma)$;
- **Optional:** Sample $\boldsymbol{\beta}^{(b)}$ and $(\sigma^2)^{(b)}$ from the corresponding full-conditional distributions defined in (2.5).

end

2.3.2 Multiple-try

Here, we present a novel method to sample from (2.6) which relies on a multiple-try approach of Liu et al. (2000), which we refer to as MTM algorithm. At each iteration, a new state of the chain $\boldsymbol{\gamma}^\star$ is selected among $K \in \mathbb{N}^+$ independent alternatives (Casarin et al., 2013) and the trans-dimensional jump is evaluated with a generalised MH step.

The k-th proposal is sampled according to the following distribution

$$q_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}\big) = \frac{1}{\binom{p}{d_k}}, \quad \text{if} \quad \sum_{j=1}^{p} |\gamma_j^{(k)} - \gamma_j| = d_k, \quad k = 1, \ldots, K, \tag{2.8}$$

which is symmetric in $\boldsymbol{\gamma}^{(k)}$ and $\boldsymbol{\gamma}$ (see the results in Appendix 2.A for a theoretical justification of equation (2.8)). This way, all model indicators $\boldsymbol{\gamma}^{(k)}$ that differ by the inclusion/exclusion of d_k covariates from the current model indicator $\boldsymbol{\gamma}$ are taken into account with the same probability. The MTM proposal distribution (2.8) gains flexibility when compared to the proposal distribution in (2.7), as it allows the chain to jump to any other possible model in the space, improving the ability of avoiding local modes. When $K = 1$ and $d_K = 1$ the chain reduces to RJ algorithm introduced in Section 2.3.1.

Let $\mathcal{D}_K = \{d_1, \ldots, d_K\}$ denote the complete set of divergences between the dimension of current and proposed models, the simple and straightforward MTM extension of the RJ approach reduces to fixing $d_k = 1$ for all $k = 1, \ldots, K$. In the same spirit of Casarin et al. (2013), to allow for more flexibility, we propose a generalized MH step that accounts for multiple independent proposals. This approach allows several alternative sampling schemes that differ by the specification of the set $\mathcal{D}_K$. An example is the specification $\mathcal{D}_K = \{1, \ldots, K\}$, which allows the transition to models that differ at most by K predictors. Of course, several alternative specifications of $\mathcal{D}_K$ are possible. Motivated by the empirical evidence that jumps to large spaces usually have low acceptance rate, the number of proposals K can increase with the dimension of the explored space.

Let $\boldsymbol{\iota}_k = (\iota_{k,1}, \ldots, \iota_{k,d_k})^\intercal$ be an indexing vector of dimension d_k, sampling from the MTM proposal distribution $q_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}\big)$ defined in (2.8) can be achieved by sampling without replacement a set of d_k variables from a discrete Uniform distribution $\mathsf{U}\big(\{1, \ldots, p\}\big)$ defined over the set $\{1, \ldots, p\}$. More specifically, the h-th predictor of the k-th model indicator $\iota_{k,h}$ is sampled from $\iota_{k,h} \sim \mathsf{U}\big(\{1, \ldots, p\} \setminus \{\iota_{k,1}, \ldots, \iota_{k,h-1}\}\big)$ in such a way that $\mathbb{P}\big(\iota_{k,h}\big) = 1/(p - h + 1)$ for $h = 1, \ldots, d_k$, with $\iota_1 \sim \mathsf{U}\big(\{1, \ldots, p\}\big)$. Then, we set $\{\gamma_h^{(k)}\}_{h \in \boldsymbol{\iota}_k} = 1 - \{\gamma_h\}_{h \in \boldsymbol{\iota}_k}$. This way, variables are included in the model if they were not and excluded, otherwise. The new proposal $\boldsymbol{\gamma}^\star$ is selected among the K alternatives according to the following discrete probability density function:

$$\bar{w}_k(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}) = \frac{w_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}\big)}{\sum_{k=1}^K w_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}\big)}, \quad k = 1, \ldots, K,$$

where a common choice for $w_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}\big)$ is that of the importance weights defined as:

$$w_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}\big) = \frac{m(\boldsymbol{\gamma}^{(k)}|\mathbf{y}, \mathbf{X})}{q_k(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma})}, \quad k = 1, \ldots, K,$$

where $m\big(\boldsymbol{\gamma}^{(k)}|\mathbf{y}, \mathbf{X}\big)$ is the marginal posterior distribution of model $\boldsymbol{\gamma}^{(k)}$ defined in (2.6) and $q_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}\big)$ is the MTM proposal defined in (2.8). Thus, new proposal is then selected as $\boldsymbol{\gamma}^\star = \boldsymbol{\gamma}^{(j)}$, $j \in \{1, \ldots, K\}$. Without defying the detailed balance condition (Liu et al., 2000), the jump between models $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}^*$ is accepted with the generalized MH acceptance probability equal to

$$\alpha_{\mathsf{MTM}} = \min\left\{1, \frac{\sum_{k=1}^K w_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}\big)}{\sum_{k=1}^K w_k\big(\mathbf{v}^{(k)}|\boldsymbol{\gamma}^\star\big)}\right\}, \tag{2.9}$$

where $\mathbf{v}^{(k)}$, $k = 1, \ldots, j-1, j+1, \ldots, K$, are $K - 1$ auxiliary values sampled from distribution $q_k\big(\mathbf{v}^{(k)}|\boldsymbol{\gamma}^\star\big)$, i.e. $\mathbf{v}^{(k)} \sim q_k\big(\mathbf{v}^{(k)}|\boldsymbol{\gamma}^\star\big)$, and $\mathbf{v}^{(j)} = \boldsymbol{\gamma}$. See Algorithm 2 for the implementation of the MTM algorithm.

It is worth recognizing that the importance weights in (2.9) depend on the evaluation of (2.6) for each proposed model. Therefore, as for the RJ algorithm, the needed quantities $\boldsymbol{\Sigma}^{\star}_{\beta_{\boldsymbol{\gamma}^{(k)}}} = \big(\widetilde{\mathbf{X}}^{\intercal}_{\boldsymbol{\gamma}^{(k)}}\widetilde{\mathbf{X}}_{\boldsymbol{\gamma}^{(k)}}\big)^{-1}$, with $\widetilde{\mathbf{X}}_{\boldsymbol{\gamma}^{(k)}} \in \mathbb{R}^{(n+p_{\boldsymbol{\gamma}^{(k)}})\times p_{\boldsymbol{\gamma}^{(k)}}}$ and $p_{\boldsymbol{\gamma}^{(k)}}$ equal to the number of variables included in $\boldsymbol{\gamma}^{(k)}$, and $S^2_{\boldsymbol{\gamma}^{(k)}} = \mathbf{y}^{\intercal}\mathbf{y} - \mathbf{y}^{\intercal}\mathbf{X}_{\boldsymbol{\gamma}^{(k)}}\big(\mathbf{X}^{\intercal}_{\boldsymbol{\gamma}^{(k)}}\mathbf{X}_{\boldsymbol{\gamma}^{(k)}} + \boldsymbol{\Sigma}^{-1}_{\beta_{\boldsymbol{\gamma}^{(k)}}}\big)^{-1}\mathbf{X}^{\intercal}_{\boldsymbol{\gamma}^{(k)}}\mathbf{y}$, $k = 1, \ldots, K$, can be computed exploiting the updating methods introduced in Section 2.4 after the inclusion/exclusion of d_k predictors.

2.3.3 Adaptive multiple-try

At each iteration, the MTM algorithm defined in Section 2.3.2 proposes K different candidate models by sampling from the proposals $\boldsymbol{\gamma}^{(k)} \sim q_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}\big)$, $k = 1, 2, \ldots, K$, where $q_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}\big)$ is defined in (2.8). The idea, however, is that larger jumps, i.e. the transition to models that differ from the current model by a large number of variables, are more useful at the beginning of the chain, whereas they should be avoided once the algorithm has converged to the true model. For this reason, here we describe a novel adaptive MTM approach which relies on a mixture of proposal kernels and adapt the mixing probabilities in order to minimise the Kullback-Leibler (KL) divergence from the target distribution. Ji and Schmidler (2013) have proposed a closely related method for adapting a mixture of exponential proposal distributions based on the KL divergence in the context of adaptive MCMC samplers. We refer to this type of algorithm as adaptive MTM (adaMTM) algorithm (see Algorithm 3 for details on the implementation).

Let $\boldsymbol{\zeta}^{(k)} \sim \mathsf{Multin}\big(1, \boldsymbol{\theta}\big)$, $\boldsymbol{\zeta}^{(k)} = \big(\zeta^{(k)}_1 \ \ldots \ \zeta^{(k)}_M\big)^{\intercal}$ and $k = 1, \ldots, K$, $\boldsymbol{\theta} = \big(\theta_1 \ \ldots \ \theta_M\big)^{\intercal}$ and $M \in \mathbb{N}_+ \setminus \{0, 1\}$. The stochastic representation of the proposal distribution for the adaptive MTM algorithm can be written as

$$
\begin{aligned}
q_a\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}, \boldsymbol{\zeta}^{(k)}\big) &= \sum_{m=1}^{M} \zeta^{(k)}_m q_m\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}\big), \quad \sum_{m=1}^{M} \zeta^{(k)}_m = 1, \\
\pi\big(\boldsymbol{\zeta}^{(k)}|\boldsymbol{\theta}\big) &= \prod_{m=1}^{M} \theta_m^{\zeta^{(k)}_m}, \quad \sum_{m=1}^{M} \theta_m = 1, \quad k = 1, \ldots, K,
\end{aligned} \tag{2.10}
$$

with marginal density

$$
q_a\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}, \boldsymbol{\theta}\big) = \sum_{m=1}^{M} \theta_m q_m\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}\big), \quad k = 1, \ldots, K, \tag{2.11}
$$

which is a mixture of M proposals $q_m\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}\big)$ defined in (2.8), with mixing probabilities $\boldsymbol{\theta}$. The sampling scheme of the adaptive MTM algorithm consists to

Algorithm 2: MTM algorithm

Input: $B \in \mathbb{N}$, $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{X} \in \mathbb{R}^{n \times p}$, $K \in \mathbb{N}$, $\mathcal{D}_K = \{d_1, \ldots, d_K\}$, $(\nu, \lambda) \in \mathbb{R}_+^2$, $(\xi, \varphi) \in \mathbb{R}_+^2$ and $\upsilon_1 \in \mathbb{R}^+$;

Initialization: sample $\phi^{(0)}$ and $\boldsymbol{\gamma}^{(0)}$ from their prior distributions;

for $(b = 1, \ldots, B)$ **do**

  for $(k = 1, \ldots, K)$ **do**

    Let $\boldsymbol{\iota}_k = [\iota_1, \ldots, \iota_{d_k}]^\intercal$, sample $\iota_{k,1} \sim \mathsf{U}(\{1, \ldots, p\})$ and $\iota_{k,h} \sim \mathsf{U}(\{1, \ldots, p\} \setminus \{\iota_{k,1}, \ldots, \iota_{k,h-1}\})$ for $h > 1$;

    Set $\left\{\boldsymbol{\gamma}_h^{(k)}\right\}_{h \in \boldsymbol{\iota}_k} = 1 - \left\{\boldsymbol{\gamma}_h^{(b-1)}\right\}_{h \in \boldsymbol{\iota}_k}$ and compute the weights $w_k(\boldsymbol{\gamma}^{(k)} | \boldsymbol{\gamma}^{(b-1)})$;

  end

  Select $\boldsymbol{\gamma}^\star = \boldsymbol{\gamma}^{(j)}$ according to the probability density

$$\bar{w}_j = \frac{w_j(\boldsymbol{\gamma}^{(j)} | \boldsymbol{\gamma}^{(b-1)})}{\sum_{k=1}^K w_k(\boldsymbol{\gamma}^{(k)} | \boldsymbol{\gamma}^{(b-1)})}, \quad \text{for} \quad j = 1, \ldots, K;$$

  Set $\mathbf{v}^{(j)} = \boldsymbol{\gamma}^{(b-1)}$ and sample $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(j-1)}, \mathbf{v}^{(j+1)}, \ldots, \mathbf{v}^{(K)}$ auxiliary values:

  for $(k = 1, \ldots, j-1, j+1, \ldots, K)$ **do**

    Let $\boldsymbol{\iota}_k = [\iota_1, \ldots, \iota_{d_k}]^\intercal$, sample $\iota_{k,1} \sim \mathsf{U}(\{1, \ldots, p\})$ and $\iota_{k,h} \sim \mathsf{U}(\{1, \ldots, p\} \setminus \{\iota_{k,1}, \ldots, \iota_{k,h-1}\})$ for $h > 1$;

    Set $\left\{\mathbf{v}_h^{(k)}\right\}_{h \in \boldsymbol{\iota}_k} = 1 - \left\{\boldsymbol{\gamma}_h^{(j)}\right\}_{h \in \boldsymbol{\iota}_k}$ and compute the weights $w_k(\mathbf{v}^{(k)} | \boldsymbol{\gamma}^{(j)})$;

  end

  Compute the MH acceptance probability:

$$\alpha(\boldsymbol{\gamma}^{(b-1)}, \boldsymbol{\gamma}^{(j)}) = \min\left\{1, \frac{\sum_{k=1}^K w_k(\boldsymbol{\gamma}^{(k)} | \boldsymbol{\gamma}^{(b-1)})}{\sum_{k=1}^K w_k(\mathbf{v}^{(k)} | \boldsymbol{\gamma}^{(j)})}\right\};$$

  With probability $\alpha(\boldsymbol{\gamma}^{(b-1)}, \boldsymbol{\gamma}^{(j)})$ set $\boldsymbol{\gamma}^{(b)} = \boldsymbol{\gamma}^{(j)}$, otherwise set $\boldsymbol{\gamma}^{(b)} = \boldsymbol{\gamma}^{(b-1)}$;

  Sample $\phi^{(b)} \sim \mathsf{Beta}(\xi + p_\gamma, \varphi + p - p_\gamma)$;

  Optional: Sample $\boldsymbol{\beta}^{(b)}$ and $(\sigma^2)^{(b)}$ from the corresponding full-conditional distributions defined in (2.5).

end

sample k proposal indicators $\boldsymbol{\zeta}^{(1)}, \ldots, \boldsymbol{\zeta}^{(K)}$ from $\boldsymbol{\zeta}^{(k)} \sim \mathsf{Multin}\big(1, \boldsymbol{\theta}\big)$, $k = 1, \ldots, K$, and then sampling the model indicator $\boldsymbol{\gamma}^{(k)}$ from the selected proposal distribution $\boldsymbol{\gamma}^{(k)} | \boldsymbol{\gamma} \sim \sum_{m=1}^M \zeta_m^{(k)} q_m(\boldsymbol{\gamma}^{(k)} | \boldsymbol{\gamma})$. The approach learns which proposal is more

reliable based on the current value of mixing probabilities vector $\boldsymbol{\theta}$. Therefore, the goal is to adapt the vector of probabilities $\boldsymbol{\theta}$ to automatically select the most promising degree of divergence from the current model, i.e. number of variables to modify. In order to ensure that all the proposals are used at the early stages, the number of proposals M should not greatly exceed the number of trials K. Therefore, we assume $M = K$ for the analysis in Section 2.5 and 2.6.

The importance weights for the adaptive MTM algorithm with the mixture proposal defined in (2.11) are

$$w_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}\big) = \frac{m\big(\boldsymbol{\gamma}^{(k)}|\mathbf{y},\mathbf{X}\big)}{q_a\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}\big)}, \quad k = 1,\dots,K, \tag{2.12}$$

where $m\big(\boldsymbol{\gamma}^{(k)}|\mathbf{y},\mathbf{X}\big)$ is the target density defined in (2.6). The optimal model $\boldsymbol{\gamma}^{\star}$ is selected among the K alternatives according to the following probabilities:

$$\begin{aligned}\bar{w}_k(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}) &= \frac{w_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}\big)}{\sum_{k=1}^{K} w_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}\big)}, \\ &= \frac{m\big(\boldsymbol{\gamma}^{(k)}|\mathbf{y},\mathbf{X}\big)}{\sum_{k=1}^{K} m\big(\boldsymbol{\gamma}^{(k)}|\mathbf{y},\mathbf{X}\big)}, \quad k = 1,\dots,K, \end{aligned} \tag{2.13}$$

where last equality holds for the symmetry of $q_a\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}\big)$ with respect to $\boldsymbol{\gamma}^{(k)}$ and $\boldsymbol{\gamma}$. New proposal is then selected as $\boldsymbol{\gamma}^{\star} = \boldsymbol{\gamma}^{(j)}$, $j \in \{1,\dots,K\}$. Without defying the detailed balance condition (see Appendix 2.A), the jump between models $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}^{*}$ is accepted with the generalized MH acceptance probability equal to

$$\alpha_{\mathsf{adaMTM}} = \min\left\{1, \frac{\sum_{k=1}^{K} m\big(\boldsymbol{\gamma}^{(k)}|\mathbf{y},\mathbf{X}\big)}{\sum_{k=1}^{K} m\big(\mathbf{v}^{(k)}|\mathbf{y},\mathbf{X}\big)}\right\},$$

where $\mathbf{v}^{(k)}$, $k = 1,\dots,j-1,j+1,\dots,K$, are $K-1$ auxiliary values sampled from distribution $q_a\big(\mathbf{v}^{(k)}|\boldsymbol{\gamma}^{\star},\boldsymbol{\theta}\big)$, i.e. $\mathbf{v}^{(k)} \sim q_a\big(\mathbf{v}^{(k)}|\boldsymbol{\gamma}^{\star},\boldsymbol{\theta}\big)$, and $\mathbf{v}^{(j)} = \boldsymbol{\gamma}$.

One of the major novelties of the proposed MTM algorithm is the possibility of automatically tuning the vector of mixture probabilities $\boldsymbol{\theta}$ to adapt the proposal to the target online. As suggested by Haario et al. (2001); Andrieu and Moulines (2006); Andrieu and Thoms (2008), at iteration $t+1$ the update of the component $\theta_m^{(t+1)}$, $m = 1,\dots,M$, of the mixture proposal distribution $q_a\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}\big)$ defined in (2.11) can be formulated as

$$\theta_m^{(t+1)} = \theta_m^{(t)} + \eta_{t+1} H\big(\theta_m^{(t)}, \boldsymbol{\zeta}^{(t)}, \boldsymbol{\gamma}^{(t)}\big), \qquad \sum_{m=1}^{M} \theta_m^{(t+1)} = 1, \tag{2.14}$$

where $\eta_t = 1/(ct^a)$, with $c > 0$ and $a \in \big(0.5, 1\big]$, is a non-increasing sequence of positive step-sizes that satisfies the conditions $\sum_{t=1}^{\infty} \eta_t = \infty$ and $\sum_{t=1}^{\infty} \eta_t^{1+\delta} < \infty$, for some $\delta > 0$ (see, e.g. Haario et al., 2001). The function $H\big(\theta_m^{(t)}, \boldsymbol{\zeta}^{(t)}, \boldsymbol{\gamma}^{(t)}\big)$ in (2.14) should be carefully selected in order to guarantee that the mixture weights adaptation scheme forces the parameters to drive the MTM proposal closer to the target. One possibility is to select a valid divergence metric and to adapt $\boldsymbol{\theta}$ in order to minimize that divergence. To this aim, we propose to minimize the the Kullback-Leibler (KL, hereafter) divergence, as in Haario et al. (2001). A natural adaptation strategy would be to select the vector parameter $\boldsymbol{\theta}$ based on the minimization of the KL divergence between $\pi(\boldsymbol{\gamma})$ and the auxiliary distribution of latent vector $\boldsymbol{\zeta}$, $\pi\big(\boldsymbol{\zeta}|\boldsymbol{\theta}\big)$ defined in (2.10), specifically

$$\begin{aligned} \mathcal{KL}\Big[\pi(\boldsymbol{\gamma}) \| \pi(\boldsymbol{\zeta}|\boldsymbol{\theta})\Big] &= \sum_{\boldsymbol{\gamma} \in \{0,1\}^p} \pi(\boldsymbol{\gamma}) \log\Big(\frac{\pi\big(\boldsymbol{\gamma}\big)}{\pi\big(\boldsymbol{\zeta}|\boldsymbol{\theta}\big)}\Big) \\ &\propto - \sum_{\boldsymbol{\gamma} \in \{0,1\}^p} \pi(\boldsymbol{\gamma}) \log \pi\big(\boldsymbol{\zeta}|\boldsymbol{\theta}\big). \end{aligned} \tag{2.15}$$

However, since the proposal distribution $q_a(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma})$ in (2.11) is not tailored to the target density $\pi\big(\boldsymbol{\gamma}\big) = m\big(\boldsymbol{\gamma}|\mathbf{y}, \mathbf{X}\big)$ defined in (2.6), the maximization of the negative of the Shannon entropy in (2.15) does not lead to a valid adaptation strategy. Therefore we propose to adapt the mixing probabilities vector $\boldsymbol{\theta}$ of the mixture proposal distribution defined in (2.11) by minimizing the KL divergence between $\pi(\boldsymbol{\gamma})$ and the following distribution:

$$\pi\big(\boldsymbol{\zeta}|\boldsymbol{\theta}, \boldsymbol{\gamma}\big) \propto \prod_{m=1}^{M} \theta_m^{\sum_{k=1}^{K} \zeta_m^{(k)} \bar{w}_k\left(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}, \boldsymbol{\theta}\right)}, \tag{2.16}$$

with $\sum_{k=1}^{K} \sum_{m=1}^{M} \zeta_m^{(k)} = K$, $\bar{w}_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}\big) \in (0,1)$ being the normalized importance weight of the k-th proposal defined in (2.13), $\sum_{k=1}^{K} \bar{w}_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}, \boldsymbol{\theta}\big) = 1$ and $\zeta_m^{(k)}$ is defined in (2.10). $\pi\big(\boldsymbol{\zeta}|\boldsymbol{\theta}, \boldsymbol{\gamma}\big)$ in (2.16) is a proper distribution function and it corresponds to a weighted likelihood function as introduced by Hu and Zidek (2002) (see Appendix 2.A).

Proposition 2.3.1. *The mixing probabilities of $q_a\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}\big)$, $k = 1, \ldots, K$ are updated as the solution of the following convex constrained maximization problem:*

$$\begin{aligned} &\arg\max_{\theta_m} \sum_{\boldsymbol{\gamma} \in \{0,1\}^p} \pi(\boldsymbol{\gamma}) \log \pi\big(\boldsymbol{\zeta}|\boldsymbol{\theta}, \boldsymbol{\gamma}\big) \\ &\textit{s.t.} \quad \sum_{m=1}^{M} \theta_m = 1, \end{aligned} \tag{2.17}$$

i.e.

$$\theta_m^{(t+1)} = \theta_m^{(t)} + \eta_{t+1}\Big(h\big(\theta_m^{(t)}, \boldsymbol{\zeta}^{(t)}, \boldsymbol{\gamma}^{(t)}\big) - \bar{h}\big(\theta_m^{(t)}, \boldsymbol{\zeta}^{(t)}, \boldsymbol{\gamma}^{(t)}\big)\Big), \tag{2.18}$$

with

$$\begin{aligned} h\big(\theta_m^{(t)}, \boldsymbol{\zeta}^{(t)}, \boldsymbol{\gamma}^{(t)}\big) &= \frac{1}{\theta_m^{(t)}} \sum_{k=1}^{K} \zeta_m^{(k),(t)} \bar{w}_k\big(\boldsymbol{\gamma}^{(k)} | \boldsymbol{\gamma}^{(t)}\big), \\ \bar{h}\big(\theta_m^{(t)}, \boldsymbol{\zeta}^{(t)}, \boldsymbol{\gamma}^{(t)}\big) &= \frac{1}{M} \sum_{m=1}^{M} h\big(\theta_m^{(t)}, \boldsymbol{\zeta}^{(t)}, \boldsymbol{\gamma}^{(t)}\big) \\ &= \frac{1}{M} \sum_{m=1}^{M} \frac{1}{\theta_m^{(t)}} \sum_{k=1}^{K} \zeta_m^{(k),(t)} \bar{w}_k\big(\boldsymbol{\gamma}^{(k)} | \boldsymbol{\gamma}^{(t)}\big), \end{aligned}$$

where $\eta_t = 1/(ct^a)$ *is the step-size of the adaptation, with* $a \in \big(0.5, 1\big]$.

Proof. See Appendix 2.A. □

Remark 2.3.1. *Note that the update in* (2.18) *ensures* $\sum_{m=1}^{M} \theta_m^{(t+1)} = 1$*, but not that* $\theta_m^{(t+1)} > 0$*. Following Ji and Schmidler (2013), rather than adding slack variables to satisfy the Karush-Kuhn-Tucker conditions, we project negative weights in the interval* $(0, 1)$ *with the rule* $\theta_m^{(t+1)} = |\theta_m^{(t+1)}| / \sum_{m=1}^{M} |\theta_m^{(t+1)}|$.

Proposition 2.3.2. *The solution of the convex constrained optimization problem in* (2.17) *is unique.*

Proof. It follows immediately from the convexity of equation (2.17). See the proof of Proposition 2.3.1 in Appendix 2.A. □

Proposition 2.3.3. *Let* $q_a\big(\boldsymbol{\gamma}^{(k)} | \boldsymbol{\gamma}, \boldsymbol{\theta}\big)$ *be the proposal distribution defined in* (2.11)*, then the Markov chain generated by adaptive MTM algorithm satisfies the detailed balance condition and converges to its stationary distribution.*

Proof. See Appendix 2.A. □

2.4 Fast evaluation of the marginal density

Let $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}^\star$ be the current and new model indicators, respectively. In this section we present how to efficiently sample regression parameters vector $\boldsymbol{\beta}_{\gamma^\star}$ and evaluate marginal posterior $m\big(\boldsymbol{\gamma}^\star | \mathbf{y}, \mathbf{X}\big)$ defined in (2.6), following the addition or deletion of one or more predictors to/from current model $\boldsymbol{\gamma}$. Hereafter, let $\mathbf{R}_{1\gamma} \in \mathbb{R}^{p_\gamma \times p_\gamma}$ be the triangular matrix related to the thinQR decomposition (see Appendix 2.D.1) of

Algorithm 3: Adaptive MTM algorithm

Input: $B \in \mathbb{N}$, $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{X} \in \mathbb{R}^{n \times p}$, $k \in \mathbb{N}$, $\mathcal{D}_M = \{d_1, \ldots, d_M\}$, $(\nu, \lambda) \in \mathbb{R}^2_+$, $(\xi, \varphi) \in \mathbb{R}^2_+$, $v_1 \in \mathbb{R}^+$, $c > 1$ and $a \in (0, 1]$;

Initialization: sample $\phi^{(0)}$ and $\boldsymbol{\gamma}^{(0)}$ from their prior distributions;

for $(b = 1, \ldots, B)$ **do**

  for $(k = 1, \ldots, K)$ **do**

    Sample $\boldsymbol{\zeta}^{(k)} \sim \mathsf{Multinom}(1, \boldsymbol{\theta})$ and set $z^{(k)} = \sum_{m=1}^{M} m\zeta_m^{(k)}$;

    Let $\boldsymbol{\iota}_k = [\iota_1, \ldots, \iota_{d_{z^{(k)}}}]^\intercal$, sample $\iota_{k,1} \sim \mathsf{U}(\{1, \ldots, p\})$ and $\iota_{k,h} \sim \mathsf{U}(\{1, \ldots, p\} \setminus \{\iota_{k,1}, \ldots, \iota_{k,h-1}\})$ for $h > 1$;

    Set $\left\{\gamma_h^{(k)}\right\}_{h \in \boldsymbol{\iota}_k} = 1 - \left\{\gamma_h^{(b-1)}\right\}_{h \in \boldsymbol{\iota}_k}$ and compute the weights $w_k(\boldsymbol{\gamma}^{(k)} | \boldsymbol{\gamma}^{(b-1)})$;

  end

  Select $\boldsymbol{\gamma}^\star = \boldsymbol{\gamma}^{(j)}$ according to the probability density

$$\bar{w}_j = \frac{w_j(\boldsymbol{\gamma}^{(j)} | \boldsymbol{\gamma}^{(b-1)})}{\sum_{k=1}^{K} w_k(\boldsymbol{\gamma}^{(k)} | \boldsymbol{\gamma}^{(b-1)})}, \quad \text{for} \quad j = 1, \ldots, K;$$

  Set $\mathbf{v}^{(j)} = \boldsymbol{\gamma}^{(b-1)}$ and sample $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(j-1)}, \mathbf{v}^{(j+1)}, \ldots, \mathbf{v}^{(K)}$ auxiliary values:

  for $(k = 1, \ldots, j-1, j+1, \ldots, K)$ **do**

    Sample $\boldsymbol{\zeta}^{(k)} \sim \mathsf{Multinom}(1, \boldsymbol{\theta})$ and set $z^{(k)} = \sum_{m=1}^{M} m\zeta_m^{(k)}$;

    Let $\boldsymbol{\iota}_k = [\iota_1, \ldots, \iota_{d_{z^{(k)}}}]^\intercal$, sample $\iota_{k,1} \sim \mathsf{U}(\{1, \ldots, p\})$ and $\iota_{k,h} \sim \mathsf{U}(\{1, \ldots, p\} \setminus \{\iota_{k,1}, \ldots, \iota_{k,h-1}\})$ for $h > 1$;

    Set $\left\{\mathbf{v}_h^{(k)}\right\}_{h \in \boldsymbol{\iota}_k} = 1 - \left\{\boldsymbol{\gamma}_h^{(j)}\right\}_{h \in \boldsymbol{\iota}_k}$ and compute the weights $w_k(\mathbf{v}^{(k)} | \boldsymbol{\gamma}^{(j)})$;

  end

  Compute the MH acceptance probability:

$$\alpha(\boldsymbol{\gamma}^{(b-1)}, \boldsymbol{\gamma}^{(j)}) = \min\left\{1, \frac{\sum_{k=1}^{K} w_k(\boldsymbol{\gamma}^{(k)} | \boldsymbol{\gamma}^{(b-1)})}{\sum_{k=1}^{K} w_k(\mathbf{v}^{(k)} | \boldsymbol{\gamma}^{(j)})}\right\};$$

  Update vector $\boldsymbol{\theta}^{(b)}$ following the adaptation rule in (2.18);

  With probability $\alpha(\boldsymbol{\gamma}^{(b-1)}, \boldsymbol{\gamma}^{(j)})$ set $\boldsymbol{\gamma}^{(b)} = \boldsymbol{\gamma}^{(j)}$, otherwise set $\boldsymbol{\gamma}^{(b)} = \boldsymbol{\gamma}^{(b-1)}$;

  Sample $\phi^{(b)} \sim \mathsf{Beta}(\xi + p_\gamma, \varphi + p - p_\gamma)$;

  Optional: Sample $\boldsymbol{\beta}^{(b)}$ and $(\sigma^2)^{(b)}$ from the corresponding full-conditional distributions defined in (2.5).

end

matrix $\widetilde{\mathbf{X}}_\gamma \in \mathbb{R}^{(n+p_\gamma)\times p_\gamma}$ defined in (2.4) such that the current posterior variance-covariance matrix is $\boldsymbol{\Sigma}^\star_{\beta_\gamma} = \left(\widetilde{\mathbf{X}}_\gamma^\intercal \widetilde{\mathbf{X}}_\gamma\right)^{-1} = \left(\mathbf{R}_{1\gamma}^\intercal \mathbf{R}_{1\gamma}\right)^{-1}$. The bottleneck of the algorithms discussed in Section 2.3 is the update of matrix $\mathbf{R}_{1\gamma^\star} \in \mathbb{R}^{p^\star_\gamma \times p^\star_\gamma}$ and value $S^2_{\gamma^\star}$ after the addition or deletion of one or more columns to/from the current design matrix. Given $\mathbf{R}_{1\gamma}$, matrix $\mathbf{R}_{1\gamma^\star}$ can be efficiently computed following the novel thinQR updating methods presented in Appendix 2.D.2, 2.D.2, 2.D.3, 2.D.3 and 2.D.3. Note that $m\left(\boldsymbol{\gamma}^\star|\mathbf{y},\mathbf{X}\right)$ is invariant with respect to the ordering of the variables. Therefore, it is assumed that new predictors are added at the end of the design matrix, greatly improving the computational performances of the model.

The rest of this section assesses the problem of evaluating $S^2_{\gamma^\star}$ given current quantities $\mathbf{R}_{1\gamma}$, $\mathbf{b}_\gamma = \mathbf{X}_\gamma^\intercal \mathbf{y} \in \mathbb{R}^{p_\gamma}$ and $\mathbf{d}_\gamma = \mathbf{R}_{1\gamma}^{-\intercal}\mathbf{b}_\gamma \in \mathbb{R}^{p_\gamma}$, where $\mathbf{R}_{1\gamma}^{-\intercal}$ denotes the transpose of the inverse of $\mathbf{R}_{1\gamma}$, i.e. $\mathbf{R}_{1\gamma}^{-\intercal} = \left(\mathbf{R}_{1\gamma}^{-1}\right)^\intercal$. Let $\mathbf{X}_\gamma \in \mathbb{R}^{n\times p_\gamma}$ and $\mathbf{X}_{\gamma^\star} \in \mathbb{R}^{n\times p_{\gamma^\star}}$ be the current and new design matrices, respectively, the goal is the efficient computation of

$$\begin{aligned}
\mathbf{S}^2_{\gamma^\star} &= \mathbf{y}^\intercal\mathbf{y} - \mathbf{y}^\intercal\mathbf{X}_{\gamma^\star}\left(\mathbf{X}_{\gamma^\star}^\intercal\mathbf{X}_{\gamma^\star} + \boldsymbol{\Sigma}^{-1}_{\beta_{\gamma^\star}}\right)^{-1}\mathbf{X}_{\gamma^\star}^\intercal\mathbf{y} \\
&= \mathbf{y}^\intercal\mathbf{y} - \mathbf{y}^\intercal\mathbf{X}_{\gamma^\star}\left(\mathbf{R}_{1\gamma^\star}^\intercal\mathbf{R}_{1\gamma^\star}\right)^{-1}\mathbf{X}_{\gamma^\star}^\intercal\mathbf{y} \\
&= \mathbf{y}^\intercal\mathbf{y} - \mathbf{b}_{\gamma^\star}^\intercal\mathbf{R}^{-1}_{1\gamma^\star}\mathbf{R}^{-\intercal}_{1\gamma^\star}\mathbf{b}_{\gamma^\star} \\
&= \mathbf{y}^\intercal\mathbf{y} - \mathbf{d}_{\gamma^\star}^\intercal\mathbf{d}_{\gamma^\star},
\end{aligned}$$

where $\mathbf{b}_{\gamma^\star} = \mathbf{X}_{\gamma^\star}^\intercal\mathbf{y} \in \mathbb{R}^{p_{\gamma^\star}}$ and $\mathbf{d}_{\gamma^\star} = \mathbf{R}_{1\gamma^\star}^{-\intercal}\mathbf{b}_{\gamma^\star} \in \mathbb{R}^{p_{\gamma^\star}}$. Moreover, target density $m\left(\boldsymbol{\gamma}^\star|\mathbf{y},\mathbf{X}\right)$ relies on the evaluation of the determinant $|\boldsymbol{\Sigma}_{\beta_{\gamma^\star}}|$, which can be efficiently computed with $\mathcal{O}\left(p_{\gamma^\star}\right)$ operations given $\mathbf{R}_{1\gamma^\star}$ as follows:

$$|\boldsymbol{\Sigma}_{\beta_{\gamma^\star}}| = \prod_{j=1}^{p_{\gamma^\star}} \mathbf{R}_{1\gamma^\star}\left[j,\ j\right]^{-2}.$$

Eventually, the regression parameter vector $\boldsymbol{\beta}_{\gamma^\star}$ is computed as

$$\boldsymbol{\beta}_{\gamma^\star} = \mathbf{R}^{-1}_{1\gamma^\star}\left(\sigma\mathbf{z} + \mathbf{d}_{\gamma^\star}\right), \tag{2.19}$$

where $\mathbf{z}$ is a $p_{\gamma^\star}$-dimensional vector with entries $z_j \sim \mathsf{N}(0,1)$, $j = 1,\ldots,p_{\gamma^\star}$. The inversion of triangular matrix $\mathbf{R}_{1\gamma^\star}$ in (2.19) can be achieved by solving the linear equation $\mathbf{R}_{1\gamma^\star}\boldsymbol{\beta}_{\gamma^\star} = \sigma\mathbf{z} + \mathbf{d}_{\gamma^\star}$ by means of forward substitutions algorithm with $\mathcal{O}\left(p^2_{\gamma^\star}\right)$ operations.

2.4.1 Add and remove variables

Add variables

Given current matrix $\widetilde{\mathbf{X}}_\gamma \in \mathbb{R}^{(n+p_\gamma)\times p_\gamma}$, after the addition of a block of $m \geq 1$

columns $\mathbf{X}_\star \in \mathbb{R}^{n\times m}$ at the end the updated form is

$$\widetilde{\mathbf{X}}_{\gamma^\star} = \begin{bmatrix} \begin{pmatrix} \widetilde{\mathbf{X}}_\gamma \\ \mathbf{0}_{m\times p_\gamma} \end{pmatrix} & \widetilde{\mathbf{X}}_\star \end{bmatrix} = \begin{bmatrix} \mathbf{X}_\gamma & \mathbf{X}_\star \\ \boldsymbol{\Sigma}_{\beta_\gamma}^{-1/2} & \mathbf{0}_{p_\gamma\times m} \\ \mathbf{0}_{m\times p_\gamma} & \boldsymbol{\Sigma}_\star^{-1/2} \end{bmatrix} \in \mathbb{R}^{(n+p_\gamma^\star)\times p_{\gamma^\star}},$$

with $\widetilde{\mathbf{X}}_\star \in \mathbb{R}^{(n+p_{\gamma^\star})\times m}$, $\boldsymbol{\Sigma}_\star = \mathrm{diag}\{\upsilon_1^2,\ldots,\upsilon_1^2\} \in \mathbb{R}^{m\times m}$ and $p_{\gamma^\star} = p_\gamma + m$. See Appendix 2.D.3 for the computation of updated triangular matrix $\mathbf{R}_{1\gamma^\star} \in \mathbb{R}^{p_{\gamma^\star}\times p_{\gamma^\star}}$ following the addition of a block of columns at position $p_\gamma + 1$ (0 entries in $\widetilde{\mathbf{X}}_\star$ can be exploited to further reduce the computational costs). Exploiting the block-form of new matrix $\mathbf{R}_{1\gamma^\star}$, vector $\mathbf{d}_{\gamma^\star} \in \mathbb{R}^{p_{\gamma^\star}}$ is

$$\mathbf{d}_{\gamma^\star} = \begin{bmatrix} \mathbf{R}_{1\gamma}^\intercal & \mathbf{0}_{p_\gamma\times 1} \\ \mathbf{R}_{12\gamma^\star}^\intercal & \mathbf{R}_{22\gamma^\star}^\intercal \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{b}_\gamma \\ \mathbf{b}_\star \end{bmatrix}, \tag{2.20}$$

where $\mathbf{b}_\star = \mathbf{X}_\star^\intercal\mathbf{y} \in \mathbb{R}^m$, $\mathbf{R}_{12\gamma^\star} = \mathbf{R}_{1\gamma^\star}[1:p_\gamma,\ (p_\gamma+1):p_{\gamma^\star}] \in \mathbb{R}^{p_\gamma\times m}$ and $\mathbf{R}_{22\gamma^\star} = \mathbf{R}_{1\gamma^\star}[(p_\gamma+1):p_{\gamma^\star},\ (p_\gamma+1):p_{\gamma^\star}] \in \mathbb{R}^{m\times m}$. Inversion of the block triangular matrix in (2.20) yields

$$\begin{aligned} \mathbf{d}_{\gamma^\star} &= \begin{bmatrix} \mathbf{R}_{1\gamma}^{-\intercal} & \mathbf{0}_{p_\gamma\times 1} \\ -\mathbf{R}_{22\gamma^\star}^{-\intercal}\mathbf{R}_{12\gamma^\star}^\intercal\mathbf{R}_{1\gamma}^{-\intercal} & \mathbf{R}_{22\gamma^\star}^{-\intercal} \end{bmatrix} \begin{bmatrix} \mathbf{b}_\gamma \\ \mathbf{b}_\star \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_{1\gamma}^{-\intercal}\mathbf{b}_\gamma \\ -\mathbf{R}_{22\gamma^\star}^{-\intercal}\mathbf{R}_{12\gamma^\star}^\intercal\mathbf{R}_{1\gamma}^{-\intercal}\mathbf{b}_\gamma + \mathbf{R}_{22\gamma^\star}^{-\intercal}\mathbf{b}_\star \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{d}_\gamma \\ \mathbf{R}_{22\gamma^\star}^{-\intercal}\big(\mathbf{b}_\star - \mathbf{R}_{12\gamma^\star}^\intercal\mathbf{d}_\gamma\big) \end{bmatrix}. \end{aligned} \tag{2.21}$$

Following (2.21), $S_{\gamma^\star}^2$ is efficiently evaluated as

$$\begin{aligned} S_{\gamma^\star}^2 &= \mathbf{y}^\intercal\mathbf{y} - \mathbf{d}_\gamma^\intercal\mathbf{d}_\gamma - \|\big(\mathbf{b}_\star^\intercal - \mathbf{d}_\gamma^\intercal\mathbf{R}_{12\gamma^\star}\big)\mathbf{R}_{22\gamma^\star}^{-1}\|_2^2 \\ &= S_\gamma^2 - \|\big(\mathbf{b}_\star^\intercal - \mathbf{d}_\gamma^\intercal\mathbf{R}_{12\gamma^\star}\big)\mathbf{R}_{22\gamma^\star}^{-1}\|_2^2. \end{aligned} \tag{2.22}$$

Therefore, new $\mathbf{d}_{\gamma^\star}$ and $S_{\gamma^\star}^2$ are updated from current values of $\mathbf{d}_\gamma$, S_γ^2, $\mathbf{R}_{12\gamma^\star}$ and $\mathbf{R}_{22\gamma^\star}$. Inversion of triangular matrix $\mathbf{R}_{22\gamma^\star}$ in (2.21) is efficiently computed by means of forward substitutions algorithm with $\mathcal{O}(m^2)$ operations. Note that when $m = 1$, this algorithm only involves the computation of vectors and scalars. Quantities $\mathbf{d}_{\gamma^\star}$ and $S_{\gamma^\star}^2$ reduce to

$$\mathbf{d}_{\gamma^\star} = \begin{bmatrix} \mathbf{d}_\gamma \\ \frac{1}{r_{22\gamma^\star}}\big(b_\star - \mathbf{r}_{12\gamma}^\intercal\mathbf{d}_\gamma\big) \end{bmatrix}$$

and

$$S^2_{\gamma^\star} = S^2_\gamma - \frac{1}{r^2_{22\gamma_\star}}\big(b_\star - \mathbf{r}^\intercal_{12\gamma^\star}\mathbf{d}_\gamma\big)^2,$$

where $b_\star = \mathbf{x}^\intercal_\star\mathbf{y} \in \mathbb{R}$ and quantities $\mathbf{r}_{12\gamma^\star} = \mathbf{R}_{1\gamma^\star}[1 : p_\gamma,\ p_\gamma + 1] \in \mathbb{R}^{p_\gamma}$ and $r_{22\gamma^\star} = \mathbf{R}_{1\gamma^\star}[p_\gamma + 1,\ p_\gamma + 1] \in \mathbb{R}$ are computed as in Appendix 2.D.2 following the addition of one column.

Remove last variables

Equations (2.21) and (2.22) provide a way to easily calculate the value of the marginal likelihood when the last $m \geq 1$ variables, i.e. columns of the matrix $\widetilde{\mathbf{X}}^\gamma \in \mathbb{R}^{(n+p_\gamma)\times p_\gamma}$, are removed. Let $p_{\gamma^\star} = p_\gamma - m$, vector $\mathbf{d}_{\gamma^\star}$ and matrix $\mathbf{R}_{1\gamma^\star}$ are immediately computed by taking the $p_{\gamma^\star}$-dimensional sub-vector $\mathbf{d}_{\gamma^\star} = \mathbf{d}_\gamma\,[1 : p_{\gamma^\star}]^\intercal$ and the $(p_{\gamma^\star} \times p_{\gamma^\star})$-dimensional sub-matrix $\mathbf{R}_{1\gamma^\star} = \mathbf{R}_{1\gamma}\,[1 : p_{\gamma^\star},\ 1 : p_{\gamma^\star}]$ (see Appendix 2.D.2 and 2.D.3), whereas quantity $S^2_{\gamma^\star}$ is efficiently evaluated as

$$S^2_{\gamma^\star} = \mathbf{y}^\intercal\mathbf{y} - \mathbf{d}^\intercal_{\gamma^\star}\mathbf{d}_{\gamma^\star}.$$

Remove variables

Assume that $m = 1$ variable needs to be removed from current model γ. This is equal to delete column $\widetilde{\mathbf{x}}_k = \begin{bmatrix}\mathbf{x}^\intercal & \mathbf{0}^\intercal_{k-1} & 1/\upsilon_1 & \mathbf{0}^\intercal_{p-k-1}\end{bmatrix}^\intercal \in \mathbb{R}^{(n+p_\gamma)}$ at position $1 \leq k < p_\gamma$ from $\widetilde{\mathbf{X}} \in \mathbb{R}^{(n+p_\gamma)\times p_\gamma}$ which is achieved by applying a set of $p_\gamma - k$ Givens rotations

$$\mathbf{G} = \mathbf{G}_{p_\gamma}(p_\gamma - 1, p_\gamma)^\intercal \times \cdots \times \mathbf{G}_{k+1}(k, k+1)^\intercal$$

to triangular matrix $\mathbf{R}_{1\gamma} \in \mathbb{R}^{p_\gamma \times p_\gamma}$ (see Appendix 2.D.2). The sequence of Given rotations can be exploited to efficiently compute new vector $\mathbf{d}_{\gamma^\star} = \mathbf{R}^{-\intercal}_{1\gamma^\star}\mathbf{b}_{\gamma^\star} \in \mathbb{R}^{p^\star_\gamma}$, with $p_{\gamma^\star} = p_\gamma - 1$ and $\mathbf{b}_{\gamma^\star} = \mathbf{b}_{\gamma(k)}$, where subscript (k) denotes the deletion of the k-th entry from $\mathbf{b}_\gamma$. New vector $\mathbf{d}_{\gamma^\star}$ is

$$\begin{aligned}
\mathbf{d}_{\gamma^\star} &= \mathbf{R}^{-\intercal}_{1\gamma^\star}\mathbf{b}_{\gamma^\star} \\
&= \big(\,[\mathbf{G}^\intercal\mathbf{R}_{1\gamma}]_{(p_\gamma,k)}\,\big)^{-\intercal}\mathbf{b}_{\gamma(k)} \\
&= \begin{bmatrix} \mathbf{R}^{11}_{1\gamma} & \big[\mathbf{R}^{12}_{1\gamma}\big]_{(\cdot,1)} \\ \mathbf{0}_{(p_\gamma-k)\times(k-1)} & \big[\mathbf{G}^\intercal_k\mathbf{R}^{22}_{1\gamma}\big]_{(p_\gamma-k+1,1)} \end{bmatrix}^{-\intercal}\mathbf{b}_{\gamma(k)},
\end{aligned} \tag{2.23}$$

where $\mathbf{G}_k = \mathbf{G}\,[k : p_\gamma,\ k : p_\gamma]$. Subscript (l, h) denote a matrix without row l and column h, whereas $\cdot$ indicates that no row (or column) is removed. The block form in equation (2.23) yields

$$\mathbf{d}_{\gamma^\star} = \begin{bmatrix} \big(\mathbf{R}^{11}_{1\gamma}\big)^{-\intercal} & \mathbf{0}_{(p_\gamma-k+1)\times(k-1)} \\ -\big(\,\big[\mathbf{G}^\intercal_k\mathbf{R}^{22}_{1\gamma}\big]_{(p_\gamma-k+1,1)}\,\big)^{-\intercal}\big(\,\big[\mathbf{R}^{12}_{1\gamma}\big]_{(\cdot,1)}\,\big)^\intercal\big(\mathbf{R}^{11}_{1\gamma}\big)^{-\intercal} & \big(\,\big[\mathbf{G}^\intercal_k\mathbf{R}^{22}_{1\gamma}\big]_{(p_\gamma-k+1,1)}\,\big)^{-\intercal} \end{bmatrix}\mathbf{b}_{\gamma(k)}, \tag{2.24}$$

where $([\mathbf{G}_k^\intercal \mathbf{R}_{1\gamma}^{22}]_{(p_\gamma-k+1,1)})^{-\intercal} = [\mathbf{G}_k^\intercal]_{(p_\gamma-k+1,\cdot)}[(\mathbf{R}_{1\gamma}^{22})^{-\intercal}]_{(\cdot,1)}$. Exploiting the properties of the Givens rotations, inverse of sub-matrix $\mathbf{R}_{1\gamma^\star}^{22} = [\mathbf{G}_k^\intercal \mathbf{R}_{1\gamma}^{22}]_{(p_\gamma-k+1,1)}$ is

$$[\mathbf{G}_k^\intercal]_{(p_\gamma-k+1,\cdot)} \left(\mathbf{R}_{1\gamma}^{22}\right)^{-\intercal} = \begin{pmatrix} \mathbf{0}_{(p_\gamma-k)\times 1} & [\mathbf{G}_k^\intercal]_{(p_\gamma-k+1,\cdot)} \left[\left(\mathbf{R}_{1\gamma}^{22}\right)^{-\intercal}\right]_{(\cdot,1)} \end{pmatrix},$$

which implies the following equalities:

$$\begin{aligned} -\left(\left[\mathbf{G}_k^\intercal \mathbf{R}_{1\gamma}^{22}\right]_{(p_\gamma-k+1,1)}\right)^{-\intercal} \left(\left[\mathbf{R}_{1\gamma}^{12}\right]_{(\cdot,1)}\right)^\intercal \left(\mathbf{R}_{1\gamma}^{11}\right)^{-\intercal} \mathbf{b}_\gamma\left[1:(k-1)\right] = \\ -[\mathbf{G}_k^\intercal]_{(p_\gamma-k+1,\cdot)} \left(\mathbf{R}_{1\gamma}^{22}\right)^{-\intercal} \left(\mathbf{R}_{1\gamma}^{12}\right)^\intercal \mathbf{d}_\gamma\left[1:(k-1)\right] \end{aligned}$$

and

$$\left(\left[\mathbf{G}_k^\intercal \mathbf{R}_{1\gamma}^{22}\right]_{(p_\gamma-k+1,1)}\right)^{-\intercal} \mathbf{b}_\gamma\left[(k+1):p_\gamma\right] = [\mathbf{G}_k^\intercal]_{(p_\gamma-k+1,\cdot)} \left(\mathbf{R}_{1\gamma}^{22}\right)^{-\intercal} \mathbf{b}_\gamma\left[k:p_\gamma\right].$$

Finally, new vector $\mathbf{d}_{\gamma^\star}$ is computed by plugging these results in 2.24, which yields

$$\begin{aligned} \mathbf{d}_{\gamma^\star} &= \begin{bmatrix} \left(\mathbf{R}_{1\gamma}^{11}\right)^{-\intercal} \mathbf{b}_\gamma\left[1:(k-1)\right] \\ [\mathbf{G}_k^\intercal]_{(p_\gamma-k+1,\cdot)} \left(\mathbf{R}_{1\gamma}^{22}\right)^{-\intercal} \left(\mathbf{b}_\gamma\left[k:p_\gamma\right] - \left(\mathbf{R}_{1\gamma}^{12}\right)^\intercal \mathbf{d}_\gamma\left[1:(k-1)\right]\right) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{d}_\gamma\left[1:(k-1)\right] \\ [\mathbf{G}_k^\intercal]_{(p_\gamma-k+1,\cdot)} \mathbf{d}_\gamma\left[k:p_\gamma\right] \end{bmatrix}. \end{aligned}$$

Therefore, $\mathbf{d}_{\gamma^\star}$ is efficiently updated alongside the computation of matrix $\mathbf{R}_{1\gamma^\star}$ when a variable is removed from the current design matrix. This is achieved by pre-multiplying previous $\mathbf{d}_\gamma$ by the sequence of Givens rotations required to update matrix $\mathbf{R}_{1\gamma}$. Updating $\mathbf{d}_\gamma$ prevents the calculation of inverse $\mathbf{R}_{1\gamma^\star}^{-1}$ needed to evaluate $S_{\gamma^\star}^2$, which becomes computationally infeasible as $p_{\gamma^\star}$ increases. Given matrix $\mathbf{G}_k$, this update is linear in p_γ, i.e. the evaluation of $\mathbf{d}_{\gamma^\star}$ is achieved with $\mathcal{O}\big(p_\gamma - k\big)$ operations.

Extension to the case $m > 1$ is done in the same fashion by replacing the Givens rotations with the Householder reflections, which share similar properties. Following Appendix 2.D.3, a block of columns $\widetilde{\mathbf{X}}_\star \in \mathbb{R}^{(n+p_\gamma)\times m}$ is deleted from matrix $\widetilde{\mathbf{X}}_\gamma \in \mathbb{R}^{(n+p_\gamma)\times p_\gamma}$ at position $k = 1, \ldots, p_\gamma - m$ by applying a sequence of $p_\gamma - k - m + 1$ Householder reflections

$$\mathbf{H} = \mathbf{H}_{p_\gamma}(p_\gamma - m + 1, p_\gamma) \times \cdots \times \mathbf{H}_{k+m}(k+1, k+m)$$

to current triangular matrix $\mathbf{R}_{1\gamma} \in \mathbb{R}^{p_\gamma \times p_\gamma}$. Relying on the same strategy for the case $m = 1$, new vector $\mathbf{d}_{\gamma^\star} \in \mathbb{R}^{p_{\gamma^\star}}$, with $p_{\gamma^\star} = p_\gamma - m$, is updated as

$$\mathbf{d}_{\gamma^\star} = \begin{bmatrix} \mathbf{d}_\gamma\left[1:(k-1)\right] \\ [\mathbf{H}_k]_{(p_\gamma-k-m+1,\cdot)} \mathbf{d}_\gamma\left[k:p_\gamma\right] \end{bmatrix}.$$

Also in this case, given matrix $\mathbf{H}_k$ computed alongside the update of $\mathbf{R}_{1\gamma}$ after the deletion of m columns, vector $\mathbf{d}_{\gamma^\star}$ is easily updated with $\mathcal{O}(p_\gamma - k - m)$ operations.

Appendix 2.D.3 assesses the problem of deleting a block of $m > 1$ non-adjacent columns. The solution is given by applying a combination of Givens rotations and Householder reflections. Therefore, the methods discussed in this section are applied to efficiently evaluate the marginal posterior distribution following the deletion of a set of non-adjacent variables given previous values of S^2_γ, $\mathbf{d}_\gamma$ and $\mathbf{R}_{1\gamma}$.

2.5 Simulation studies

In this Section, we assess the sampling properties of the algorithms discussed in Section 2.3 with several simulation experiments. We test variable selection and computational efficiency of RJ, MTM and adaptive MTM against the most efficient stochastic search variable selection algorithm which is the scalable spike-and-slab of Biswas et al. (2022) (SSS, R package "ScaleSpikeSlab"). For variable selection, we consider the *median probability model* (MPM) of Barbieri and Berger (2004) and the *maximum a-posteriori* (MAP) for RJ, MTM and adaMT, whereas only MPM is evaluated for SSS. To assess the efficiency of the target distribution update based on the thinQR methods explained in Section 2.4, we include in the analysis also the RJ algorithm with ordinary QR updating methods (see, e.g. Chambers, 1971). Eventually, we compare algorithms RJ, MTM and adaMTM and show how the latter outperforms the other competitors in terms of acceptance probability and exploration of the target distribution.

We consider the simulation scheme of Johnson and Rossell (2012) with different settings of n, p, p_0, where p_0 denotes the real number of non-zero coefficients. The response vector is generated following the linear model defined in (2.1) with $\sigma^2 = 1$. The p-dimensional vector $\boldsymbol{\beta}$ is defined as $\boldsymbol{\beta} = (\boldsymbol{\beta}_0^\intercal, 0_{p-p_0}^\intercal)^\intercal$, where

$$\boldsymbol{\beta}_0 = (-1)^{\mathbf{u}} 5\Big(\frac{\log(n)}{\sqrt{n}} + |\mathbf{z}|\Big), \tag{2.25}$$

with $\mathbf{z} \sim \mathsf{N}_{p_0}(0, \mathbf{I}_{p_0})$ and $\mathbf{u} \sim \mathsf{Bin}(p_0, 0.4)$.

For each simulated dataset, we perform 50000 updates of parameter $\boldsymbol{\gamma}$ (we do not sample $\boldsymbol{\beta}$ and σ^2) and variable selection is evaluated over a post-burnin period of 25000 iterations. For each chain, in order to avoid the exploration of unreliable models with huge dimension, we set the maximum number of predictors to 250. We consider 20 replications of each simulation case. As concerns the choice of the divergence set for MTM and adaMTM models and prior hyper-parameters setting, previous empirical evidence suggests $\mathcal{D}_K = \{1, 3, 5, 10\}$ and $\sigma^2 \sim \mathsf{IG}(0.5, 5)$, for all the synthetic and standardized data examples. Higher values for λ, i.e. $\sigma^2 \sim (0.5, 10)$, are set when dealing with small signal to noise ratio in order to avoid

overfitting, for example when $p > 5000$. We specify hyper-parameters ξ, φ and v_1^2 following Narisetty and He (2014). In particular, we compute ξ and φ by fixing the mean and standard deviation of the Beta hyper-prior in advance. In order to force a sparse solution, we fix the prior standard deviation of ϕ to 0.01, whereas the mean is computed such that $\mathbb{P}\big(\sum_{j=1}^{p}\gamma_j = 1 > Q|\phi\big) = 0.1$, for a default value $Q = \max\{40, \log(n)\}$. Finally, we set $v_1^2 = \max\{p^{2.1}/(100n), \log(n)\}$.

Simulation study 1. In the first scenario, we consider independent predictors and the design matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$ is generated as $x_{ij} \sim \mathsf{N}(0,1)$, $i = 1,\dots,n$, $j = 1,\dots,p$. We study the cases $n = \{100, 200, 400\}$, $p = \{500, 1000, 3000, 5000, 10000\}$ and $p_0 = \{10, 20, 30\}$. The results of variable selection are shown in Figures 2.1 and 2.2, which represent the AUC and F1 scores for each considered value of n, p and p_0. Overall, adaMTM and SSS provide the best AUC scores, with the latter performing better in the case of lowest information, i.e. when $n = 100$. Indeed, in this case adaMTM works well up to $p = 3000$ and $p_0 = 20$, with AUC score around 0.7, whereas the performance dramatically worsen as p increases. On the contrary, F1 scores are always better for adaMTM algorithm. This is due to the fact that MPM in not the optimal model for SSS, as it provides low marginal inclusion probabilities estimates (Figure 2.B.1 in Appendix 2.B). Therefore, the optimal threshold is never 0.5 and other methods (such as BIC criterion) should be implemented for its choice. When $n = 200$, RJ and MTM still provide low scores of AUC and F1 when $p = 10000$, and they reach good results only when $n = 400$. The difference between MPM and MAP models is small when $n = 200$ and $n = 400$, with the former achieving slightly higher scores for both AUC and F1 indexes. On the other hand, MPM regularly outperforms MAP model when $n = 100$.

Figure 2.B.2 in Appendix 2.B shows the comparison between RJ, MTM and adaMTM in terms of exploration of the target density, e.g. the Hamming distance between the visited and true models and the acceptance rate of the MCMC algorithms. Algorithm adaMTM provides the fastest convergence to the true model (Figures 2.B.2a and 2.B.2b) and the highest acceptance rate (Figure 2.B.2c). Finally, we analysed the efficiency of the considered competitors: thinQR updating methods yield the most efficient approach, with a 10-fold decrease in computational time when comparing adaMTM and SSS algorithms. Therefore, to sum up, the former is able to approach the variable selection performance of the latter with a great improvement in terms of efficiency.

Simulation study 2. In the second simulation study, we sample the design matrix with correlated predictors as $\mathbf{x}_i \sim \mathsf{N}_p(\mathbf{0}_p, \mathbf{\Sigma}_X)$, $i = 1,\dots,n$, where $\mathbf{\Sigma}_X$ is such that $(\mathbf{\Sigma}_X)_{lj} = \rho^{|l-j|}$ and we fix $\rho = 0.5$. The considered cases are $n = \{200, 400\}$,

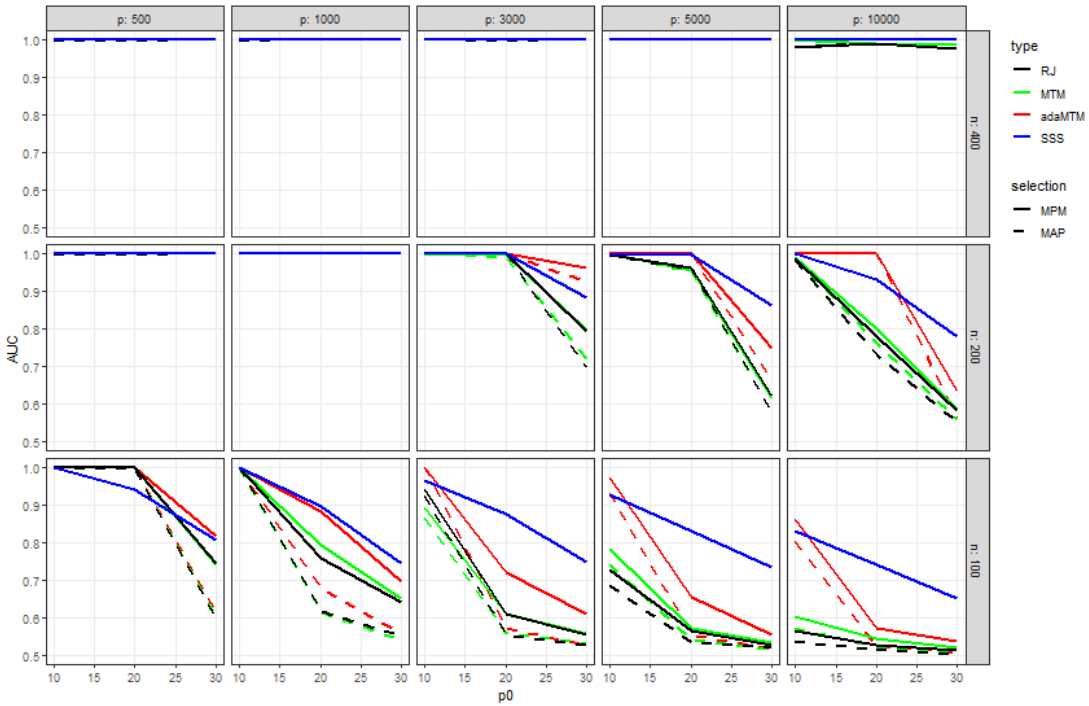


Figure 2.1: Average AUC score for variable selection (over 20 replications) using different algorithms in the simulation study 1 for different values of n, p and p_0. Each replication consists of 25000 post-burnin draws from the posterior distribution of $\boldsymbol{\gamma}$. The compared algorithms are: RJ, MTM, adaMTM and the Scalable Spike-and-Slab algorithm of Biswas et al. (2022) ("SSS"). The data are simulated as suggested by Johnson and Rossell (2012).

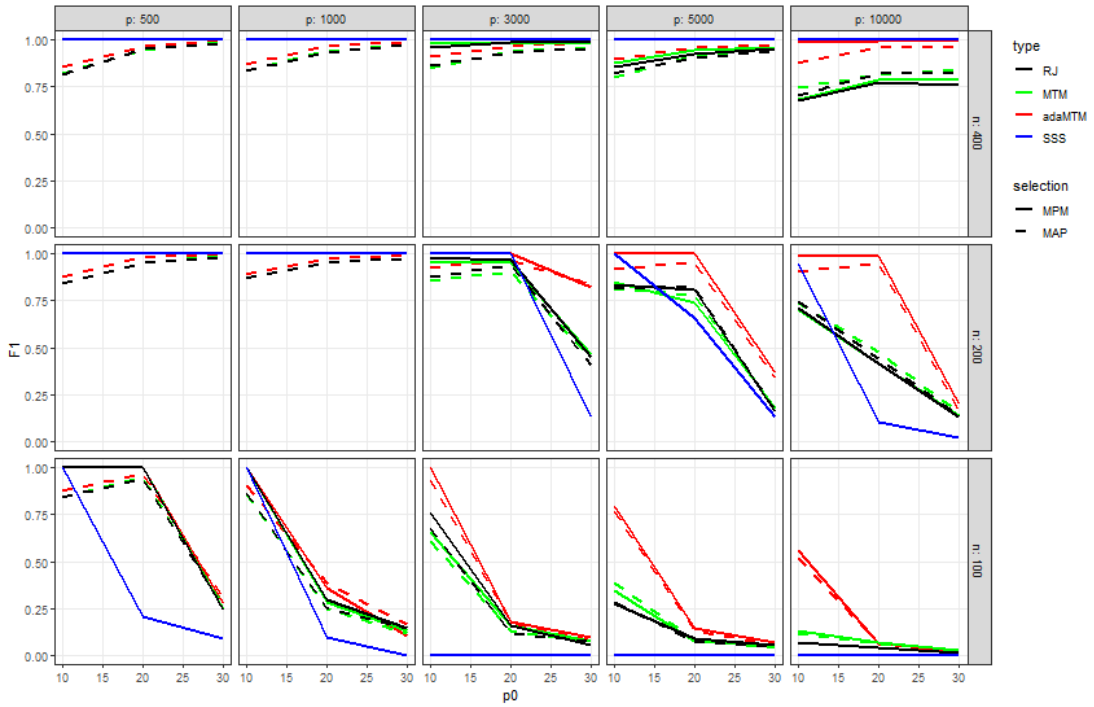


Figure 2.2: Average F1 score for variable selection (over 20 replications) using different algorithms in the simulation study 1 for different values of n, p and p_0. Each replication consists of 25000 post-burnin draws from the posterior distribution of $\boldsymbol{\gamma}$. The compared algorithms are: RJ, MTM, adaMTM and the Scalable Spike-and-Slab algorithm of Biswas et al. (2022) ("SSS"). The data are simulated as suggested by Johnson and Rossell (2012).

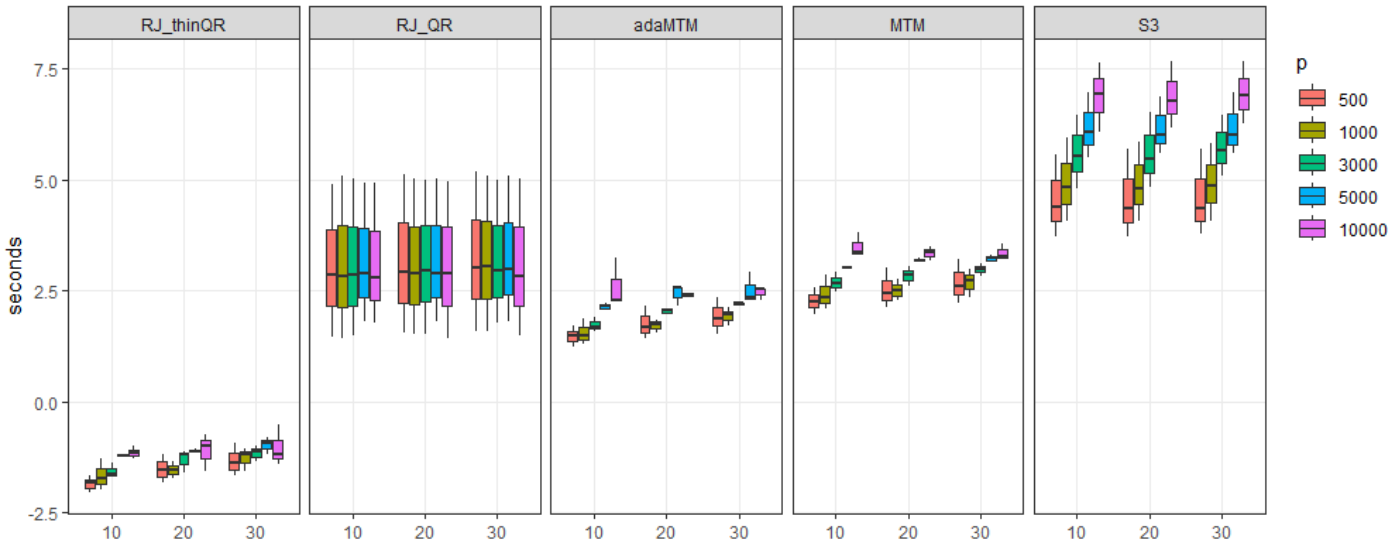


Figure 2.3: Average computation time (over 20 replications) in log-seconds using different algorithms in the simulation study 1 for different values of p and p_0. Data have been aggregated for values of $n = \{100, 200, 400\}$. Each replication consists of 50000 draws from the posterior distribution of $\boldsymbol{\gamma}$. The compared algorithms are: RJ featuring the thinQR decomposition "thinQR", RJ featuring the QR decomposition "QR", MTM, adaMTM and the Scalable Spike-and-Slab algorithm of Biswas et al. (2022) ("SSS"). The data are simulated as suggested by Johnson and Rossell (2012).

$p = \{1000, 3000, 5000, 10000\}$ and $p_0 = \{10, 20, 30\}$. AUC and F1 scores are shown in Figure 2.B.3 and 2.B.4 in Appendix 2.B. Similar considerations to those of simulation study 1 can be done, however the difference between adaMTM and SSS in terms of AUC is smaller, whereas F1 scores are regularly better for the former method, especially in the most high dimensional case. Even with correlated predictors, MPM model outperforms MAP in terms of variable selection.

2.6 Real data applications

In this Section, we present the application of models RJ, MTM and adaMTM defined in Section 2.3 to two real datasets. In the first application we consider a low-dimensional case in order to assess the quality of the $\boldsymbol{\beta}$ estimates and compare the marginal posterior inclusion probabilities against the scalable spike-and-slab of Biswas et al. (2022) (SSS). We then apply the methods to a high-dimensional microarray dataset concerning gene expression from eye tissue in laboratory rats.

2.6.1 Inflation data

The first dataset, "*Inflation*", is taken from Bernardi et al. (2016) and considers predicting US inflation, measured as the changes in the US consumer price index, using quarterly data from several macroeconomic indicators. In this example, we consider all the observations between 1978-Q2 and 2021-Q3, for a total of $n = 147$ observations and $p = 14$ variables. Further details on the variables and their sources can be found in Appendix 2.C, where Table 2.C.1 provides a complete description of the variables used as covariates in the linear regression model. We perform 5000 iterations, with a post-burnin of period 2500. The hyperparameters for RJ, MTM and adaMTM are set to $\sigma^2 \sim \mathsf{IG}(0.5, 5)$, $\phi \sim \mathsf{Beta}(1, 1)$ and $v_1^2 = \max\{p^{2.1}/(100n), \log(n)\}$.

Figure 2.4 shows the boxplot of the marginal posterior distribution of the components $\boldsymbol{\beta}$: these are similar for all the considered competitors. The main difference concerns the estimate of the smallest effects, as SSS provides a distribution centered around 0, while RJ, MTM and adaMTM set those coefficients exactly to 0. The estimated marginal posterior inclusion probabilities (Figure 2.C.1 in Appendix 2.C) shows a higher degree of shrinkage on the coefficients for algorithm RJ, MTM and adaMTM, which regularly provide lower probabilities for the zero coefficients than SSS.

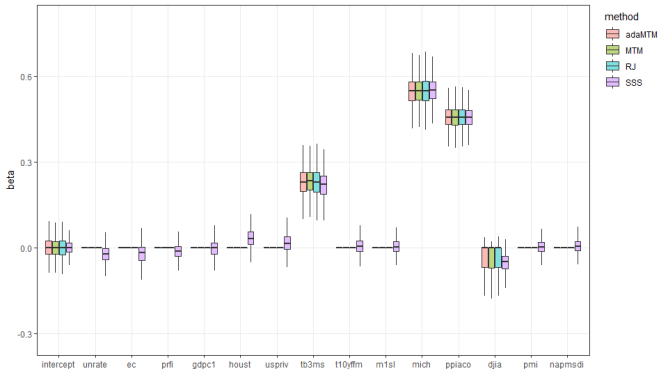


Figure 2.4: Boxplot of the posterior distribution of parameters $\boldsymbol{\beta}$ for dataset Inflation. Each algorithm has performed 2500 post-burnin iterations. The compared algorithms are: RJ, MTM, adaMTM and the Scalable Spike-and-Slab algorithm of Biswas et al. (2022) ("SSS").

2.6.2 Microarray data

The second dataset, "*Bardet-Biedl*", is a microarray dataset consisting of gene expression measurements from the eye tissue of 120 laboratory rats. The data was originally studied by Scheetz et al. (2006) to investigate mammalian eye disease, and later analyzed by Breheny and Huang (2015); Bai et al. (2022) to demonstrate the performance of their group variable selection algorithms. The goal of this analysis is to identify genes which are associated with the gene TRIM32. TRIM32 has previously been shown to cause Bardet-Biedl syndrome (Chiang et al., 2006), a disease affecting multiple organs including the retina. Following the approach in Scheetz et al. (2006), 18976 of the 31042 probe sets on the array "exhibited sufficient signal for reliable analysis and at least 2-fold variation in expression". These probe sets include TRIM32 and 18975 other genes that potentially influence its expression. Among these, we consider a subset of most correlated predictors with the response variable, for a total of $p = 4703$ selected probes.

We estimated the models from 10 different starting points, where each replication consists of a total of 50000 draws from the posterior distribution, with a post-burnin period of 25000. As concerns the hyperparameters setting, after controlling for the degree of sparsity of the estimated models, we set $\sigma^2 \sim \mathsf{IG}(0.5, 3.5)$, whereas υ_1^2, ξ and φ are estimated following Narisetty and He (2014), as in Section 2.5. The set of divergence for MTM and adaMTM is $\mathcal{D}_K = \{1, 3, 5, 10\}$, whereas parameters of the adaptive step-size are set to $c = 10$ and $a = 0.55$.

The average marginal posterior inclusion probabilities of the predictors estimated by RJ, MTM, adaMTM and SSS across the 10 replications are shown in

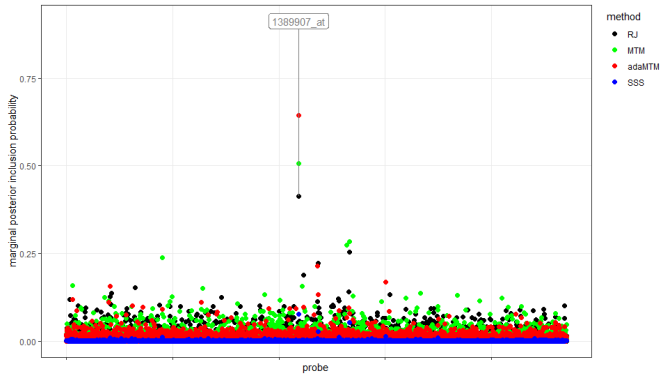


Figure 2.5: Average marginal posterior inclusion probabilities of predictors for dataset Bardet-Biedl across 10 replications of the models (25000 post-burnin iterations for each replication). The compared algorithms are: RJ, MTM, adaMTM and the Scalable Spike-and-Slab algorithm of Biswas et al. (2022) ("SSS").

Figure 2.5. SSS does not include any gene in the model, with really low inclusion probabilities. The probe with the highest inclusion probability is 1389907, which is included in most of the MAP models (see Table 2.C.3). RJ tends to include a larger number of predictors, whereas MTM and adaMTM assign higher probabilities to a smaller number of probes. The main discrepancies between these last two methods concern probes 1390168, 1378316 and 1391096. In particular, the latter is never included by algorithm MTM.

The trace of the log-target density for RJ, MTM and adaMTM is shown in Figure 2.C.2 in Appendix 2.C. All the three methods show similar behaviour across the replications, even when starting from a low-density zone (chain 8). In this latter case, adaMTM converges faster to local maximum. Finally, we study the convergence of the MCMC chains by estimating the potential scale reduction factor (Gelman and Rubin, 1992) marginally for each β_j, $j = 1, \ldots, 4703$ (see Table 2.C.2). These values give insights on the convergence, but they should not be completely trusted, as the β are not always sampled at each iteration. Algorithm adaMTM provides the best estimated values, i.e. closest to the interval $(1.0, 1.2)$.

2.7 Conclusion and discussion

With this paper we develop multiple trans-dimensional MCMC sampling methods for model selection in high-dimensional linear regression with Gaussian errors. The introduced methods rely on a Delta spike-and-slab prior (George and McCulloch, 1993, 1997), with prior inclusion probability of the predictors guided by a Beta

hyper-prior. In particular, we implement three different algorithms (RJ, MTM and adaMTM described in Section 2.3) and assess their sampling properties with intensive simulations and the application to two real datasets. The most promising approach is represented by the adaptive method adaMTM, which provides better results in terms of variable selection, exploration of the target density and rate of convergence. Moreover, by relying on the thinQR updating methods discussed in Section 2.4, these results are achieved with a much improved computational efficiency when compared to SSS model of Biswas et al. (2022).

In Section 2.5 and 2.6 we analysed different settings for the hyperparameters and we find that our approaches are sensitive to their choice. We rely on the considerations of Narisetty and He (2014) in order to provide sensible values of υ_1^2, ξ and φ. The most delicate issue concerns the specification of ν and λ related to the prior distribution of residual variance σ^2: different choices lead to different degrees of sparsity in the estimated models and influence the convergence of the algorithms. Therefore, an optimal calibration, based also on prior evidence, is fundamental for obtaining accurate results.

Algorithm adaMTM is justified by some theoretical properties of the adaptive scheme in Appendix 2.A. However, additional proofs on the ergodicity of the MCMC are needed. To this aim, the works of Ji and Schmidler (2013) and Fontaine and Bédard (2022) provide promising results for the analysis of the convergence of our approach.

The main drawback of adaMTM is the important loss of accuracy when low information is available, i.e. when the number of observation is particularly small. In this case, further tuning of the mixture proposal distribution is required, where a possible extension is provided by the informed trans-dimensional transitions (Gagnon, 2021).

Future work will be to account for the considerations made in Martino and Louzada (2017) and to calibrate the optimal number of trials. An interesting solution could be to assume a random maximum divergence $K \geq 1$, with the goal of adapting the MCMC jumps size as the chain proceeds. Such generalization must come with a theoretical justification of the method, as it is not clear whether it provides an ergodic MCMC algorithm.

Finally, the described methods can be extended to the case of binary outcome via probit data-augmentation scheme (Albert and Chib, 1993). However, the updating methods implemented for linear regression with Gaussian errors can not be directly applied to this case, as the introduction of the probit latent variable does not allow the update of the fixed vector $\mathbf{d}_\gamma = (\boldsymbol{\Sigma}^\star_{\beta_\gamma})^{-1/2}\mathbf{X}_\gamma^\intercal\mathbf{y}$ (defined in Section 2.4) after the addition or deletion of a set of variables and, therefore, further considerations are required.

Appendix

Appendix 2.A Additional theoretical results and proofs

Proposition 2.A.1. *(Proposal distribution for the MTM algorithm) Let $\boldsymbol{\gamma}$ be the current model, the number of models that differ from the current model by adding or deleting d_k predictors in the adaptive MTM algorithm defined in Subsection 2.3.2 is*

$$\sum_{j=0}^{d_k} \binom{p-p_\gamma}{j} \binom{p_\gamma}{d_k-j} = \binom{p}{d_k}, \tag{2.26}$$

where $\binom{n}{k}$ is the Binomial coefficient.

Proof. The Quantity $\binom{p-p_\gamma}{j}\binom{p_\gamma}{d_k-j}$ in (2.26) denotes the total number of models that differ from the current model $\boldsymbol{\gamma}$ by adding j and deleting $d_k - j$ different predictors. The right hand side of (2.26) is obtained by applying the Vandermonde's convolution formula Graham et al. (1994) as it is evident since there are $\binom{p}{d_k}$ ways to choose an (unordered) subset of d_k regressors from the set of p covariates. □

Proof. (of weighted likelihood being a valid distribution) After some simple algebraic operations, at the t-th iteration, the updating probability distribution is

$$\begin{aligned}\pi\big(\boldsymbol{\zeta}|\boldsymbol{\theta},\boldsymbol{\gamma}\big) &\propto \prod_{k=1}^{K}\left(\pi\big(\boldsymbol{\zeta}^{(k)}|\boldsymbol{\theta}\big)\right)^{\bar{w}_k\left(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}^{(t)}\right)} \\ &\propto \prod_{k=1}^{K}\left(\prod_{m=1}^{M}\theta_m^{\zeta_m^{(k)}}\right)^{\bar{w}_k\left(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}^{(t)}\right)} \\ &\propto \prod_{k=1}^{K}\prod_{m=1}^{M}\theta_m^{\zeta_m^{(k)}\bar{w}_k\left(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}^{(t)}\right)} \\ &\propto \prod_{m=1}^{M}\left(\prod_{k=1}^{K}\theta_m^{\zeta_m^{(k)}\bar{w}_k\left(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}^{(t)}\right)}\right) \\ &\propto \prod_{m=1}^{M}\theta_m^{\sum_{k=1}^{K}\zeta_m^{(k)}\bar{w}_k\left(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}^{(t)}\right)},\end{aligned} \tag{2.27}$$

with $\sum_{k=1}^{K}\sum_{m=1}^{M}\zeta_m^{(k)} = K$, where $\pi(\boldsymbol{\zeta}^{(k)}|\boldsymbol{\theta})$ is the likelihood function of latent vector parameter $\boldsymbol{\zeta}^{(k)}$ defined in (2.10) and $\bar{w}_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}^{(t)}\big) \in (0,1)$ is the normalized importance weight of the k-th proposal defined in (2.13), with $\sum_{k=1}^{K}\bar{w}_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}^{(t)}\big) = 1$. Therefore, from (2.27), $\pi\big(\boldsymbol{\zeta}|\boldsymbol{\theta},\boldsymbol{\gamma}\big)$ is the probability density function of a Multinomial random variable $\pi\big(\boldsymbol{\zeta}^{(k)}|\boldsymbol{\theta}\big)$ weighted by $\bar{w}_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}^{(t)}\big) > 0$. The normalizing constant of (2.27) is

$$\prod_{k=1}^{K}\left(\frac{M!}{\prod_{m=1}^{M}\big(\zeta_m^{(k)}\big)!}\right)^{\bar{w}_k\left(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}^{(t)}\right)},$$

which completes the proof. □

Proof. (of Proposition 2.3.1, theoretical justification for adaptive MTM updating mechanism) As suggested by Haario et al. (2001); Andrieu and Moulines (2006); Andrieu and Thoms (2008), the update of the component $\theta_m^{(t+1)}$, $m = 1,\ldots,M$, of algorithm adaMTM proposal distribution $q_a(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}) = \sum_{m=1}^{M}\theta_m q_m(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma})$ at iteration $t+1$ can be formulated as

$$\theta_m^{(t+1)} = \theta_m^{(t)} + \eta_{t+1}H\big(\theta_m^{(t)},\boldsymbol{\zeta}^{(t)},\boldsymbol{\gamma}^{(t)}\big), \quad \sum_{m=1}^{M}\theta_m^{(t+1)} = 1, \tag{2.28}$$

where $\eta_t = 1/(ct^a)$, with $c > 0$ and $a \in \big(0.5, 1\big]$, is a non-increasing sequence of positive step-sizes that satisfies the conditions $\sum_{t=1}^{\infty}\eta_t = \infty$ and $\sum_{t=1}^{\infty}\eta_t^{1+\delta} < \infty$,

for some $\delta > 0$ (see, e.g. Haario et al., 2001). The function $H\big(\theta_m^{(t)}, \boldsymbol{\zeta}^{(t)}, \boldsymbol{\gamma}^{(t)}\big)$ in (2.28) is selected in order to minimize the KL divergence between $\pi(\boldsymbol{\gamma})$ and $\pi\big(\boldsymbol{\zeta}|\boldsymbol{\theta}, \boldsymbol{\gamma}\big)$ as defined in equation (2.16), i.e.

$$\begin{aligned} &\arg\max_{\theta_m} \sum_{\boldsymbol{\gamma}\in\{0,1\}^p} \pi(\boldsymbol{\gamma}) \log \pi\big(\boldsymbol{\zeta}|\boldsymbol{\theta}, \boldsymbol{\gamma}\big) \\ &\text{s.t.} \quad \sum_{m=1}^{M} \theta_m = 1. \end{aligned}$$

Under the constraint $\sum_{m=1}^{M} \theta_m = 1$, the minimization problem above can be solved by means of Lagrange multipliers, (see, e.g. Nocedal and Wright, 2006). Specifically

$$\begin{aligned} H\big(\theta_m, \boldsymbol{\zeta}, \boldsymbol{\gamma}\big) &= \arg\min_{\theta_m} \left[- \sum_{\boldsymbol{\gamma}\in\{0,1\}^p} \pi(\boldsymbol{\gamma}) \log \pi\big(\boldsymbol{\zeta}|\boldsymbol{\theta}, \boldsymbol{\gamma}\big) - \lambda\Big(\sum_{m=1}^{M} \theta_m - 1\Big) \right] \\ &= \arg\max_{\theta_m} \left[\sum_{\boldsymbol{\gamma}\in\{0,1\}^p} \pi(\boldsymbol{\gamma}) \log \pi\big(\boldsymbol{\zeta}|\boldsymbol{\theta}, \boldsymbol{\gamma}\big) + \lambda\Big(\sum_{m=1}^{M} \theta_m - 1\Big) \right] \\ &= \frac{\partial}{\partial \theta_m} \left[\sum_{\boldsymbol{\gamma}\in\{0,1\}^p} \pi\big(\boldsymbol{\gamma}\big) \log \pi\big(\boldsymbol{\zeta}|\boldsymbol{\theta}, \boldsymbol{\gamma}\big) + \lambda\Big(\sum_{m=1}^{M} \theta_m - 1\Big) \right] \\ &= \sum_{\boldsymbol{\gamma}\in\{0,1\}^p} \pi\big(\boldsymbol{\gamma}\big) \frac{\sum_{k=1}^{K} \zeta_m^{(k)} \bar{w}_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}, \boldsymbol{\theta}^{(t)}\big)}{\theta_m} + \lambda, \end{aligned}$$

where $\lambda \geq 0$ is the Lagrange multiplier. Since $H\big(\theta_m, \boldsymbol{\zeta}, \boldsymbol{\gamma}\big)$ involve the intractable summation over the space of competing models we rely on the Monte Carlo approximation

$$\widehat{H}\big(\theta_m, \boldsymbol{\zeta}, \boldsymbol{\gamma}\big) = \frac{1}{B} \sum_{b=1}^{B} \Big(\frac{\sum_{k=1}^{K} \zeta_m^{(k)} \bar{w}_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}^{(b)}, \boldsymbol{\theta}^{(t)}\big)}{\theta_m} \Big) + \lambda,$$

where $\boldsymbol{\gamma}^{(b)} \sim \pi(\boldsymbol{\gamma})$, for $b = 1, \ldots, B$, and $B > 1$ is the number of MC samples. Taking $B = 1$, i.e. only the quantity evaluated at the last iteration, and imposing the property $\sum_{m=1}^{M} \widehat{H}\big(\theta_m, \boldsymbol{\zeta}, \boldsymbol{\gamma}\big) = 0$ in order to ensure that vector of probabilities $\boldsymbol{\theta}$ sum up to 1, it yields

$$\lambda = -\frac{1}{M} \sum_{m=1}^{M} \frac{1}{\theta_m} \sum_{k=1}^{K} \zeta_m^{(k)} \bar{w}_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma}, \boldsymbol{\theta}^{(t)}\big),$$

which completes the proof. □

Proof. (of Proposition 2.3.3, detailed balance condition for the adaptive MTM algorithm) Let $\pi(\boldsymbol{\gamma}) = m(\boldsymbol{\gamma}|\mathbf{y}, \mathbf{X})$ denote the target density. To guarantee that the Markov chain generated by the adaptive MTM algorithm converges to its stationary distribution, we prove that the transition kernel $\mathbb{T}(\cdot, \cdot)$ generated by algorithm 3 fulfills the property $\pi(\boldsymbol{\gamma})A(\boldsymbol{\gamma}, \boldsymbol{\gamma}^\star) = \pi(\boldsymbol{\gamma}^\star)A(\boldsymbol{\gamma}^\star, \boldsymbol{\gamma})$, where $\boldsymbol{\gamma}^\star = \boldsymbol{\gamma}^{(j)}$ is the selected j-th proposal among the K trials and $A(\boldsymbol{\gamma}, \boldsymbol{\gamma}^\star)$ is the transition probability for the jump from $\boldsymbol{\gamma}$ to $\boldsymbol{\gamma}^\star$. Specifically, let

$$\mathbb{T}_{\boldsymbol{\zeta}}\big(\boldsymbol{\gamma}^\star|\boldsymbol{\gamma}\big) \equiv \mathbb{T}\big(\boldsymbol{\gamma}^\star|\boldsymbol{\gamma}, \boldsymbol{\zeta}\big) = \sum_{m=1}^{M} \zeta_m^{(j)} q_m\big(\boldsymbol{\gamma}^\star|\boldsymbol{\gamma}\big),$$

be the transition kernel conditional to the component indicator $z^{(j)} = \sum_{m=1}^{M} m\zeta_m^{(j)}$ and let

$$\begin{aligned}\mathbb{T}\big(\boldsymbol{\gamma}^\star|\boldsymbol{\gamma}, \boldsymbol{\theta}\big) &= \sum_{m=1}^{M} \mathbb{P}\big(z^{(j)} = m|\boldsymbol{\theta}\big) q_m\big(\boldsymbol{\gamma}^\star|\boldsymbol{\gamma}\big) \\ &= \sum_{m=1}^{M} \theta_m q_m\big(\boldsymbol{\gamma}^\star|\boldsymbol{\gamma}\big),\end{aligned}$$

be the corresponding unconditional kernel to transit from $\boldsymbol{\gamma}$ to $\boldsymbol{\gamma}^\star$. Following Liu et al. (2000), we define

$$w_j\big(\boldsymbol{\gamma}^\star|\boldsymbol{\gamma}, \boldsymbol{\theta}\big) = \pi(\boldsymbol{\gamma}^\star)\mathbb{T}\big(\boldsymbol{\gamma}|\boldsymbol{\gamma}^\star, \boldsymbol{\theta}\big)\lambda\big(\boldsymbol{\gamma}^\star, \boldsymbol{\gamma}\big)$$

where $\lambda\big(\boldsymbol{\gamma}^\star, \boldsymbol{\gamma}\big)$ is a symmetric function, i.e. $\lambda\big(\boldsymbol{\gamma}^\star, \boldsymbol{\gamma}\big) = \lambda\big(\boldsymbol{\gamma}, \boldsymbol{\gamma}^\star\big)$. The importance weights for the adaptive MTM algorithm discussed in Section 2.3.3 imply the following choice of the function $\lambda\big(\boldsymbol{\gamma}^\star, \boldsymbol{\gamma}\big)$:

$$\lambda\big(\boldsymbol{\gamma}^\star, \boldsymbol{\gamma}\big) = \mathbb{T}\big(\boldsymbol{\gamma}^\star|\boldsymbol{\gamma}, \boldsymbol{\theta}\big)^{-1}\mathbb{T}\big(\boldsymbol{\gamma}|\boldsymbol{\gamma}^\star, \boldsymbol{\theta}\big)^{-1}.$$

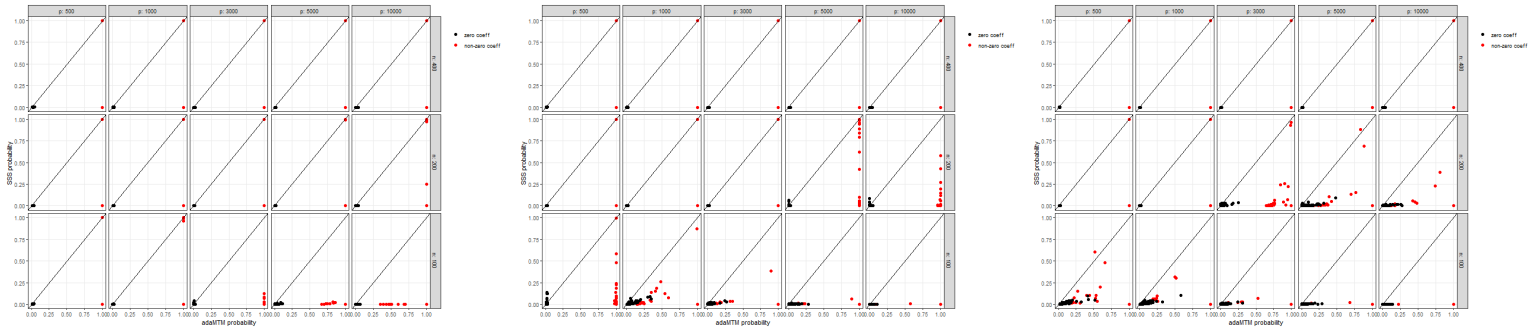
Without loss of generality, assume $\boldsymbol{\gamma} \neq \boldsymbol{\gamma}^\star$ and that the j-th component is sampled, the detailed balance condition states

$$
\begin{aligned}
\pi(\boldsymbol{\gamma})A(\boldsymbol{\gamma},\boldsymbol{\gamma}^\star) = K\pi(\boldsymbol{\gamma}) &\sum_{\boldsymbol{\gamma}^{(1)}}\cdots\sum_{\boldsymbol{\gamma}^{(j-1)}}\sum_{\boldsymbol{\gamma}^{(j+1)}}\cdots\sum_{\boldsymbol{\gamma}^{(K)}}\sum_{\mathbf{v}^{(1)}}\cdots\sum_{\mathbf{v}^{(j-1)}}\sum_{\mathbf{v}^{(j+1)}}\cdots\sum_{\mathbf{v}^{(K)}} \\
&\mathbb{T}\big(\boldsymbol{\gamma}^\star|\boldsymbol{\gamma},\boldsymbol{\theta}\big)\mathbb{T}\big(\boldsymbol{\gamma}^{(1)}|\boldsymbol{\gamma},\boldsymbol{\theta}\big)\times\cdots\times\mathbb{T}\big(\boldsymbol{\gamma}^{(j-1)}|\boldsymbol{\gamma},\boldsymbol{\theta}\big)\mathbb{T}\big(\boldsymbol{\gamma}^{(j+1)}|\boldsymbol{\gamma},\boldsymbol{\theta}\big)\times\cdots \\
&\times\mathbb{T}\big(\boldsymbol{\gamma}^{(K)}|\boldsymbol{\gamma},\boldsymbol{\theta}\big)\frac{w_j\big(\boldsymbol{\gamma}^\star|\boldsymbol{\gamma},\boldsymbol{\theta}\big)}{w_j\big(\boldsymbol{\gamma}^\star|\boldsymbol{\gamma},\boldsymbol{\theta}\big)+\sum_{k=1,k\neq j}^K w_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}\big)} \\
&\times\min\left\{1,\frac{w_j\big(\boldsymbol{\gamma}^\star|\boldsymbol{\gamma},\boldsymbol{\theta}\big)+\sum_{k=1,k\neq j}^K w_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}\big)}{w_j\big(\boldsymbol{\gamma}|\boldsymbol{\gamma}^\star,\boldsymbol{\theta}\big)+\sum_{k=1,k\neq j}^K w_k\big(\mathbf{v}^{(k)}|\boldsymbol{\gamma}^\star,\boldsymbol{\theta}\big)}\right\} \\
&\times\mathbb{T}\big(\mathbf{v}^{(1)}|\boldsymbol{\gamma}^\star,\boldsymbol{\theta}\big)\times\cdots \\
&\times\mathbb{T}\big(\mathbf{v}^{(j-1)}|\boldsymbol{\gamma}^\star,\boldsymbol{\theta}\big)\mathbb{T}\big(\mathbf{v}^{(j+1)}|\boldsymbol{\gamma}^\star,\boldsymbol{\theta}\big)\times\cdots\times\mathbb{T}\big(\mathbf{v}^{(K)}|\boldsymbol{\gamma}^\star,\boldsymbol{\theta}\big) \\
= K\frac{w_j\big(\boldsymbol{\gamma}|\boldsymbol{\gamma}^\star,\boldsymbol{\theta}\big)w_j\big(\boldsymbol{\gamma}^\star|\boldsymbol{\gamma},\boldsymbol{\theta}\big)}{\lambda\big(\boldsymbol{\gamma}^\star,\boldsymbol{\gamma}\big)} &\\
\times\sum_{\boldsymbol{\gamma}^{(1)}}\cdots\sum_{\boldsymbol{\gamma}^{(j-1)}}\sum_{\boldsymbol{\gamma}^{(j+1)}}\cdots\sum_{\boldsymbol{\gamma}^{(K)}}\sum_{\mathbf{v}^{(1)}}\cdots\sum_{\mathbf{v}^{(j-1)}}\sum_{\mathbf{v}^{(j+1)}}\cdots\sum_{\mathbf{v}^{(K)}} &\\
\mathbb{T}\big(\boldsymbol{\gamma}^{(1)}|\boldsymbol{\gamma},\boldsymbol{\theta}\big)\times\cdots\times\mathbb{T}\big(\boldsymbol{\gamma}^{(j-1)}|\boldsymbol{\gamma},\boldsymbol{\theta}\big)\mathbb{T}\big(\boldsymbol{\gamma}^{(j+1)}|\boldsymbol{\gamma},\boldsymbol{\theta}\big)\times\cdots &\\
\times\mathbb{T}\big(\boldsymbol{\gamma}^{(K)}|\boldsymbol{\gamma},\boldsymbol{\theta}\big) &\\
\times\min\left\{\frac{1}{w_j\big(\boldsymbol{\gamma}^\star|\boldsymbol{\gamma},\boldsymbol{\theta}\big)+\sum_{k=1,k\neq j}^K w_k\big(\boldsymbol{\gamma}^{(k)}|\boldsymbol{\gamma},\boldsymbol{\theta}\big)},\right. &\\
\left.\frac{1}{w_j\big(\boldsymbol{\gamma}|\boldsymbol{\gamma}^\star,\boldsymbol{\theta}\big)+\sum_{k=1,k\neq j}^K w_k\big(\mathbf{v}^{(k)}|\boldsymbol{\gamma}^\star,\boldsymbol{\theta}\big)}\right\}\times &\\
\times\mathbb{T}\big(\mathbf{v}^{(1)}|\boldsymbol{\gamma}^\star,\boldsymbol{\theta}\big)\times\cdots\times\mathbb{T}\big(\mathbf{v}^{(j-1)}|\boldsymbol{\gamma}^\star,\boldsymbol{\theta}\big)\times &\\
\times\mathbb{T}\big(\mathbf{v}^{(j+1)}|\boldsymbol{\gamma}^\star,\boldsymbol{\theta}\big)\times\cdots\times\mathbb{T}\big(\mathbf{v}^{(K)}|\boldsymbol{\gamma}^\star,\boldsymbol{\theta}\big), &
\end{aligned}
$$

where $\mathbf{v}^{(k)}$, $k = 1,\ldots,j-1,j+1,\ldots,K$, are $K-1$ auxiliary values sampled from distribution $q_a(\mathbf{v}^{(k)}|\boldsymbol{\gamma}^\star,\boldsymbol{\theta})$, i.e. $\mathbf{v}^{(k)} \sim q_a(\mathbf{v}^{(k)}|\boldsymbol{\gamma}^\star,\boldsymbol{\theta})$, and $\mathbf{v}^{(j)} = \boldsymbol{\gamma}$. Because of the symmetry of $\lambda\big(\boldsymbol{\gamma}^\star,\boldsymbol{\gamma}\big)$, we conclude that $\pi(\boldsymbol{\gamma})A(\boldsymbol{\gamma},\boldsymbol{\gamma}^\star) = \pi(\boldsymbol{\gamma}^\star)A(\boldsymbol{\gamma}^\star,\boldsymbol{\gamma})$. □

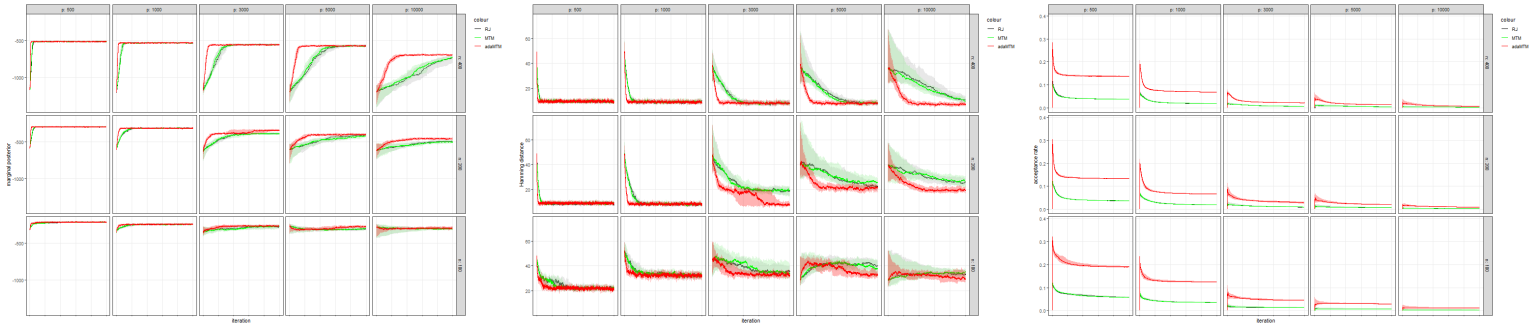
Appendix 2.B Additional results for the simulation study

In this Appendix we report some additional figures for the evaluation of variable selection and the exploration of the target density concerning the simulation studies in Section 2.5.



(a) $p_0 = 10$ (b) $p_0 = 20$ (c) $p_0 = 30$

Figure 2.B.1: Average marginal inclusion probability of the predictors (over 20 replications) using different algorithms in the simulation study 1 for different values of n, p and p_0. Each replication consists of 25000 post-burnin samples from the posterior distribution of $\boldsymbol{\gamma}$. The compared algorithms are: adaMTM and the Scalable Spike-and-Slab algorithm of Biswas et al. (2022) ("SSS"). The data are simulated as suggested by Johnson and Rossell (2012).



(a) log-posterior (b) Hamming distance (c) acceptance rate

Figure 2.B.2: Quality comparison of the performances of algorithms RJ, MTM and adaMT. The panels show the median and the first and third quartiles (over 20 replications) in the simulation study 1 of a) the value of the target density at each iteration, b) the Hamming distance between the model at each iteration and the true model and c) the average acceptance rate of the algorithms. The data have been aggregated for values of $p_0 = \{10, 20, 30\}$. The data are simulated as suggested by Johnson and Rossell (2012).

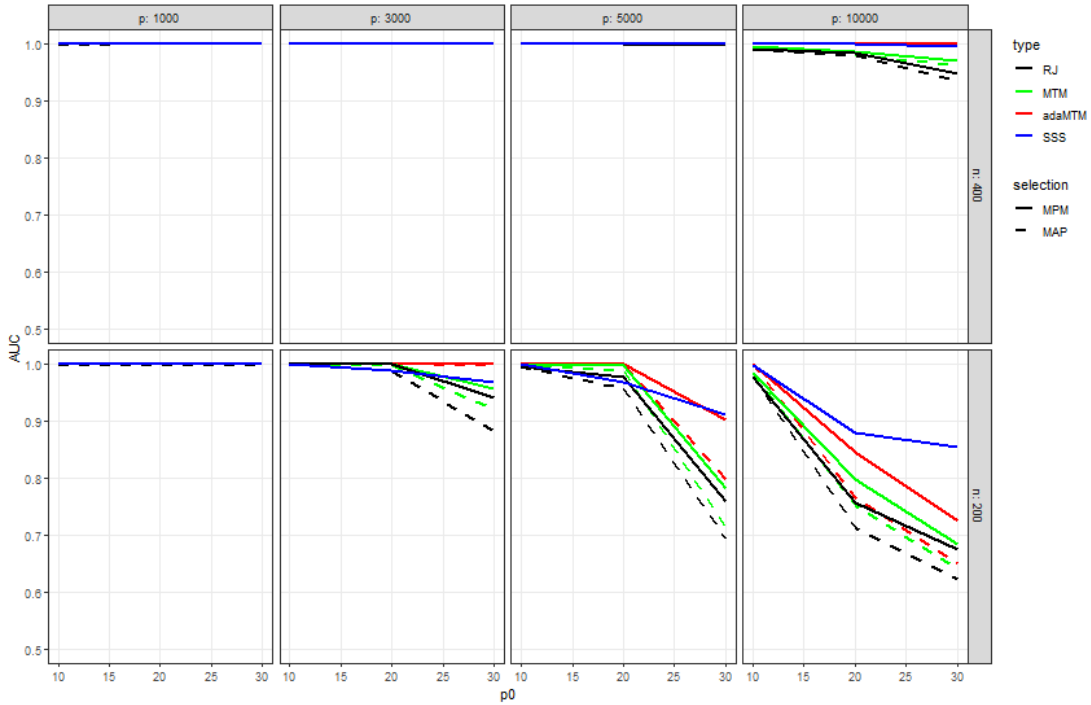


Figure 2.B.3: Average AUC score for variable selection (over 20 replications) using different algorithms in the simulation study 2 for different values of n, p and p_0. Each replication consists of 25000 post-burnin draws from the posterior distribution of $\boldsymbol{\gamma}$. The compared algorithms are: RJ, MTM, adaMTM and the Scalable Spike-and-Slab algorithm of Biswas et al. (2022) ("SSS"). The data are simulated as suggested by Johnson and Rossell (2012) with correlated predictors.

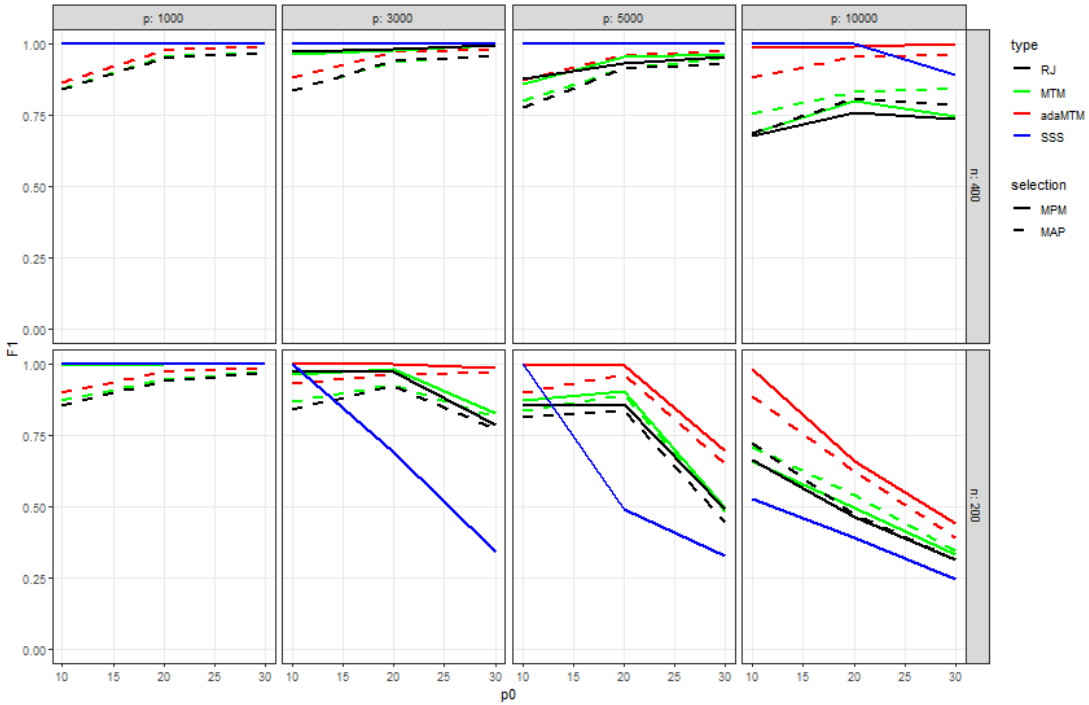


Figure 2.B.4: Average F1 score for variable selection (over 20 replications) using different algorithms in the simulation study 2 for different values of n, p and p_0. Each replication consists of 25000 post-burnin draws from the posterior distribution of $\boldsymbol{\gamma}$. The compared algorithms are: RJ, MTM, adaMTM and the Scalable Spike-and-Slab algorithm of Biswas et al. (2022) ("SSS"). The data are simulated as suggested by Johnson and Rossell (2012) with correlated predictors.

Appendix 2.C Additional material for real data applications

In this Appendix some additional materials on the datasets used in Section 2.6 are reported.

2.C.1 Inflation data

Inflation prediction, measured as the changes in the consumer price index (CPIAUCSL, CPILFESL), using quarterly data from several macroeconomic indicators, see Bernardi et al. (2016). Table 2.C.1 provides a complete description of the variables used as covariates in the linear regression model. In this example, we consider all the observations between 1978-Q2 and 2021-Q3. Further details on the variables used and their sources can be found in the Data Appendix of Bernardi et al. (2016).

Table 2.C.1: Inflation database, see Bernardi et al. (2016) for further details. All the variables are publicly available for download from the FRED database maintained by the Federal Reserve Bank of St. Louis, https://fred.stlouisfed.org.

#	Variable name	Variable type	Variable description
1	DATE	date	date
2	CPIAUCSL	numerical	Consumer Price Index for All Urban Consumers: All Items in U.S. City Average
3	CPILFESL	numerical	Consumer Price Index for All Urban Consumers: All Items Less Food and Energy in U.S. City Average
4	UNRATE	numerical	Unemployment Rate
5	PCEC	numerical	Real Personal Consumption Expenditures
6	PRF	numerical	Private Residential Fixed Investment
7	GDPC1	numerical	Real Gross Domestic Product
8	HOUST	numerical	New Privately-Owned Housing Units Started: Total Units
9	USPRIV	numerical	Employees, Total Private
10	TB3MS	numerical	3-Month Treasury Bill Secondary Market Rate
11	T10Y3MM	numerical	10-Year Treasury Constant Maturity Minus 3-Month Treasury Constant Maturity
12	M1SL	numerical	Money supply - M1
13	MICH	numerical	University of Michigan: Inflation Expectation
14	PPIACO	numerical	Producer Price Index by Commodity: All Commodities
15	DJIA	numerical	Dow Jones Industrial Average Index
16	PMI	numerical	Purchasing Manager's composite index (Institute of Supply Management)
17	VENDOR	numerical	NAPM vendor deliveries index

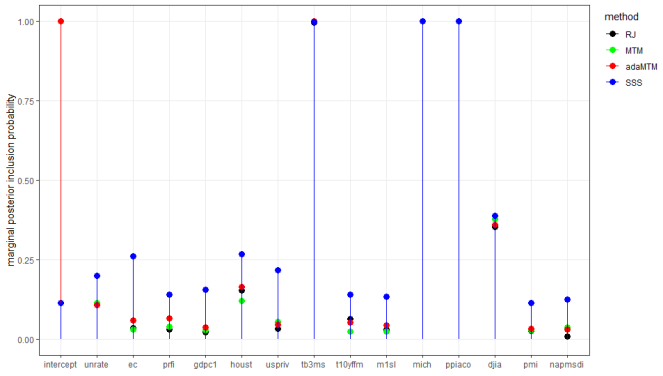


Figure 2.C.1: Estimated marginal posterior inclusion probability for each predictors of dataset Inflation. Each algorithm has performed 2500 post-burnin iterations. The compared algorithms are: RJ, MTM, adaMTM and the Scalable Spike-and-Slab algorithm of Biswas et al. (2022) ("SSS").

2.C.2 Microarray data

Here, we provide additional details on the application of algorithms RJ, MTM and adaMTM to the dataset Bardet-Biedl.

Table 2.C.2: Distribution of the estimated potential scale reduction factors computed over a post-burning period of 25000 updates for regression parameter $\boldsymbol{\beta}$ across 10 replications. NAs are related to those predictors with marginal posterior inclusion probability equal to 0. Optimal values of the index should lie in the interval $(1, 1.2)$.

Algorithm	Min	1st Qu.	Median	Mean	3rd Qu.	Max	NAs
RJ	1.158	1.304	1.348	1.479	1.457	5.352	3557
MTM	1.154	1.305	1.345	1.456	1.447	5.443	3559
adaMTM	1.101	1.291	1.299	1.306	1.327	2.906	2133

Table 2.C.3: Probes of dataset Bardet-Biedl selected more than once across 10 replications by the MAP models with RJ, MTM and adaMTM, along with the average β estimate and marginal posterior inclusion probability (mip).

probe	RJ		MTM		adaMTM		SSS	
	est.	mip	est.	mip	est.	mip	est.	mip
1371109_at	0.04	0.14	0.00	0.00	0.00	0.02	0.00	0.00
1372671_at	0.06	0.15	0.01	0.04	0.00	0.01	0.01	0.00
1378289_at	-0.02	0.14	0.00	0.00	-0.02	0.10	-0.01	0.00
1378316_at	-0.06	0.25	-0.07	0.28	-0.02	0.08	-0.02	0.05
1383783_at	0.02	0.09	0.00	0.00	0.00	0.01	0.00	0.00
1384110_at	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00
1389907_at	0.15	0.41	0.20	0.51	0.27	0.64	0.03	0.08
1390168_a_at	-0.07	0.19	0.00	0.00	-0.02	0.10	0.00	0.00
1391096_at	-0.04	0.11	-0.02	0.05	-0.09	0.21	-0.01	0.01
1391322_at	0.00	0.04	0.00	0.02	0.00	0.02	0.00	0.00
1391484_at	-0.04	0.22	-0.01	0.04	-0.03	0.13	-0.01	0.03
1394037_at	0.02	0.09	0.00	0.01	0.00	0.02	0.01	0.00
1368625_at	0.01	0.04	0.03	0.10	0.00	0.00	0.00	0.00
1372262_at	0.02	0.09	0.00	0.00	0.00	0.00	0.00	0.00
1373777_at	0.02	0.10	0.00	0.00	0.00	0.01	0.00	0.00
1378452_at	-0.01	0.05	0.00	0.00	0.00	0.01	0.00	0.00
1383638_at	0.00	0.09	0.00	0.00	0.00	0.01	0.00	0.00
1384903_at	-0.01	0.04	0.00	0.03	0.00	0.01	0.00	0.00
1390682_at	-0.01	0.04	0.00	0.01	-0.01	0.04	0.00	0.00
1393063_at	0.02	0.09	0.01	0.05	0.00	0.04	0.01	0.00
1393360_at	0.01	0.08	0.00	0.00	0.00	0.00	0.00	0.00
1395517_at	0.02	0.06	0.00	0.01	0.00	0.01	0.00	0.00
1395881_at	0.01	0.05	0.00	0.00	0.00	0.02	0.00	0.00
1398590_at	-0.01	0.10	0.00	0.00	0.00	0.04	0.00	0.00
1367874_at	0.01	0.12	0.00	0.00	0.00	0.02	0.00	0.00
1368484_at	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00
1368980_at	-0.02	0.08	0.00	0.01	0.00	0.00	0.00	0.00
1370201_at	0.00	0.04	0.00	0.00	0.00	0.00	0.01	0.00
1370411_at	0.01	0.08	0.00	0.00	0.00	0.01	0.00	0.00
1371524_at	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00
1371841_at	0.01	0.05	0.00	0.00	0.00	0.00	0.00	0.00
1372821_at	0.00	0.04	0.00	0.02	0.00	0.01	0.00	0.00
1378438_at	0.01	0.08	0.01	0.03	0.00	0.02	0.00	0.00
1378524_at	0.00	0.08	0.00	0.00	0.00	0.00	0.00	0.00
1383106_at	0.01	0.05	0.00	0.00	0.00	0.01	0.01	0.00
1387732_at	0.00	0.10	0.00	0.00	0.00	0.01	0.00	0.00
1389618_at	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00

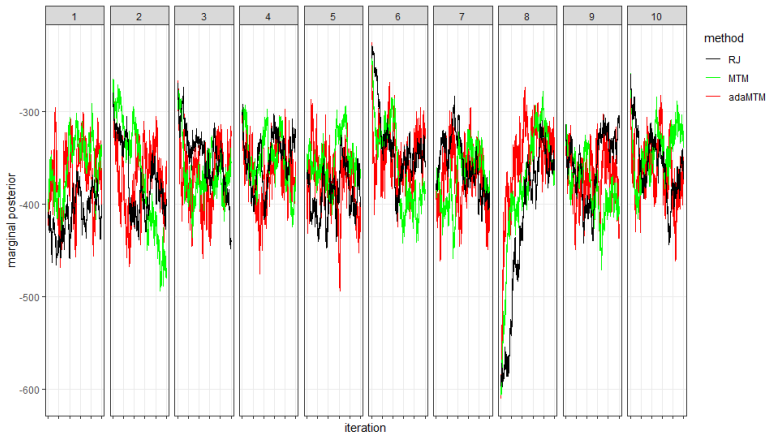


Figure 2.C.2: Exploration of the target density for dataset Bardet-Biedl across 10 replications of the models. The compared algorithms are: RJ, MTM and adaMTM.

Appendix 2.D ThinQR update

In Appendix 2.D.1 we present an overview of QR and thinQR decompositions, as well as the most common methods for their computation (see Golub and Van Loan (2013) for a detailed dissertation). In Appendix 2.D.2 and 2.D.3 we discuss novel updating algorithms for the efficient update of thinQR decomposition.

2.D.1 QR and thinQR decompositions

Many statistical applications require the inversion of matrix $\widetilde{\mathbf{X}}^\intercal\widetilde{\mathbf{X}} = \mathbf{X}^\intercal\mathbf{X} + \boldsymbol{\Sigma}_{\beta_\gamma}^{-1}$, where $\widetilde{\mathbf{X}} \in \mathbb{R}^{(n+p_\gamma)\times p_\gamma}$. A way of speeding up this inversion is by exploiting the QR decomposition. In what follows we will consider a generic $N \times m$ matrix $\mathbf{X}$ of full column rank.

The QR decomposition factorises matrix $\mathbf{X}$ into $\mathbf{QR}$, where $\mathbf{Q}$ is a $N \times N$ orthogonal matrix and $\mathbf{R}$ is a $N \times m$ upper trapezoidal matrix $\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0}_{N-m,m} \end{bmatrix}$ with $\mathbf{R}_1$ being a square upper triangular matrix and

$$\mathbf{Q}^\intercal\mathbf{X} = \mathbf{R}.$$

The two most common methods to obtain such factorisation are by exploiting either a sequence of Householder reflections or Givens rotations. For further references see Golub and Van Loan (2013) and Björck (2015).

Following the work of Golub and Van Loan (2013), it can be shown that, for $k \in \{1, \ldots, m\}$ it yields

$$\operatorname{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_k\} = \operatorname{span}\{\mathbf{q}_1, \ldots, \mathbf{q}_k\},$$

where $\mathbf{q}_1, \ldots, \mathbf{q}_k$ is the k-dimensional subspace formed by the columns of matrix $\mathbf{Q}$ and $\operatorname{span}\{S\}$ refers to the smallest linear subspace of S. This result allows a reduced QR decomposition such that

$$\begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}^\intercal \mathbf{X} = \begin{bmatrix}\mathbf{R}_1 \\ \mathbf{0}\end{bmatrix},$$

where $\mathbf{Q}_1 \in \mathbb{R}^{N\times m}$, $\mathbf{Q}_2 \in \mathbb{R}^{N\times\left(N-m\right)}$ and $\mathbf{R}_1 \in \mathbb{R}^{m\times m}$ represents the thinQR decomposition of $\mathbf{X}$. It is straightforward to show that $\mathbf{X} = \mathbf{Q}_1\mathbf{R}_1$. Hence, computational costs may be lowered by applying algorithms that update only the reduced matrix $\mathbf{R}_1$.

Householder reflections

The most common method of computing QR decomposition relies on multiple Householder reflections applied to $\mathbf{X}$. An Householder matrix with normal vector $\mathbf{v} \in \mathbb{R}^N$ is a $N \times N$ symmetric and orthogonal matrix defined as

$$\mathbf{H} = \mathbf{I}_N - \tau\mathbf{v}\mathbf{v}^\intercal, \quad \tau = \frac{2}{\|\mathbf{v}\|_2^2},$$

where $\|\mathbf{v}\|_2 = \sqrt{\mathbf{v}^\intercal\mathbf{v}}$ is the ℓ_2-norm of the vector $\mathbf{v}$. The Householder matrix $\mathbf{H}$ for the first column of $\mathbf{X}$ is built so that $\mathbf{H}\mathbf{x}_1 = \alpha\mathbf{e}_1$, where $\mathbf{e}_1 \in \mathbb{R}^N$ is the first column of $\mathbf{I}_N$. By setting

$$\begin{aligned}\widetilde{\mathbf{v}} &= \begin{bmatrix}\mathbf{x}_1[1] + \mathsf{sign}\big(\mathbf{x}_1[1]\big)\|\mathbf{x}_1\|_2 \\ \mathbf{x}_1[2:N]\end{bmatrix} \\ \mathbf{v} &= \frac{\widetilde{\mathbf{v}}}{\widetilde{\mathbf{v}}[1]},\end{aligned}$$

it can be shown that $\mathbf{H}\mathbf{x}_1 = -\mathsf{sign}\big(\mathbf{x}_1[1]\big)\|\mathbf{x}_1\|_2\mathbf{e}_1$. Thus, the QR decomposition of $\mathbf{X}$ can be computed as a sequence of m Householder reflections applied to $\mathbf{X}$. This way all elements under the diagonal of $\mathbf{X}$ are set to zero. The orthogonal matrix $\mathbf{Q}^\intercal$ is then defined as

$$\mathbf{H}_m\mathbf{H}_{m-1} \times \cdots \times \mathbf{H}_1 = \mathbf{Q}^\intercal,$$

where $\mathbf{H}_i$, $i = 1, 2, \ldots, m$, is the Householder matrix related to the i-th column of $\mathbf{X}$, with normal vectors $\mathbf{v}_i = \widetilde{\mathbf{v}}_i / \widetilde{\mathbf{v}}_i[i]$ with

$$\widetilde{\mathbf{v}}_i = \begin{bmatrix} \mathbf{0}_{i-1} \\ \mathbf{x}_i\,[i] + \mathsf{sign}\big(\mathbf{x}_i\,[i]\,\big) \|\mathbf{x}_i\,[i:N]\,\|_2 \\ \mathbf{x}_i\,[(i+1):N] \end{bmatrix}.$$

In general, it is possible to set to 0 the elements from $j > i$ to k of column i of $\mathbf{X}$ while modifying only element $\mathbf{x}_i[i]$ of column i. This can be done by multiplying $\mathbf{X}$ by the Householder matrix $\mathbf{H}_i(j,k)$, which has normal vector $\mathbf{v}_{i(j,k)} = \widetilde{\mathbf{v}}_{i(j,k)} / \widetilde{\mathbf{v}}_{i(j,k)}[i]$ with

$$\widetilde{\mathbf{v}}_{i(j,k)} = \begin{bmatrix} \mathbf{0}_{i-1,1} \\ \mathbf{x}_i[i] + \mathsf{sign}\big(\mathbf{x}_i\,[i]\,\big) \|\mathbf{x}_i[\star]\|_2 \\ \mathbf{0}_{j-i-1,1} \\ \mathbf{x}_i[j:k] \\ \mathbf{0}_{N-k,1} \end{bmatrix},$$

where $\mathbf{x}_i[\star] = [\mathbf{x}_i[i] \quad \mathbf{x}_i[j:k]]$ is the vector obtained by stacking entry i and entries from j to k of column $\mathbf{x}_i$.

Givens rotation

Another method to compute QR decomposition of a matrix $\mathbf{X} \in \mathbb{R}^{N \times m}$ relies on Givens rotations, which introduces one zero at a time under the diagonal. A Givens matrix is a $N \times N$ orthogonal matrix defined as

$$\mathbf{G}(i,j) = \begin{array}{c} \begin{matrix} \qquad & \qquad & i & \qquad & j & \qquad \end{matrix} \\ \begin{bmatrix} \mathbf{I}_{i-1} & \vdots & \mathbf{0}_{i-1,j-i-1} & \vdots & \mathbf{0}_{i-1,N-j} \\ \cdots & c & \cdots & s & \cdots \\ \mathbf{0}_{j-i-1,i-1} & \vdots & \mathbf{I}_{j-i-1} & \vdots & \mathbf{0}_{j-i-1,N-j} \\ \cdots & -s & \cdots & c & \cdots \\ \mathbf{0}_{N-j,i-1} & \vdots & \mathbf{0}_{N-j,j-i-1} & \vdots & \mathbf{I}_{N-j} \end{bmatrix} \end{array} \begin{matrix} \\ \\ i \\ \\ j \end{matrix}$$

where the dots stand for vectors of 0's of the appropriate dimension, $c = \cos\big(\theta\big)$ and $s = \sin\big(\theta\big)$, for some θ and $i < j$ with $i, j \in \mathbb{Z}^+$. Let $\mathbf{x}$ be a N-dimensional vector, then values c and s can be computed analytically as

$$c = \frac{x_i}{\sqrt{x_i^2 + x_j^2}}, \quad s = \frac{-x_j}{\sqrt{x_i^2 + x_j^2}},$$

and $\mathbf{G}(i,j)^\intercal \mathbf{x} = \widetilde{\mathbf{x}}$, where

$$\widetilde{x}_k = \begin{cases} cx_i - sx_j, & \text{if } k = i \\ 0, & \text{if } k = j, \\ x_k & \text{otherwise.} \end{cases}$$

The QR decomposition can be computed by applying a sequence of Givens rotations to sequentially set to zero all elements under the diagonal. In particular, orthogonal matrix $\mathbf{Q}$ can be computed as

$$\mathbf{G}_m(m, m+1)^\intercal \cdots \mathbf{G}_m(m, N)^\intercal \cdots \mathbf{G}_1(1,2)^\intercal \cdots \mathbf{G}_1(1, N)^\intercal = \mathbf{Q}^\intercal,$$

where subscript j denotes the Givens rotation applied to the j-th column of $\mathbf{X}$.

2.D.2 Adding and deleting one column

Here we consider the update of triangular matrix $\mathbf{R}_1$ following the addition of column $\mathbf{x}_\star \in \mathbb{R}^N$ at the end or the deletion of column $\mathbf{x}_k \in \mathbb{R}^N$ at position $k = \{1, \ldots, m\}$.

Adding one column

The thinQR update of the augmented matrix $\mathbf{X}^+ = \begin{bmatrix} \mathbf{X} & \mathbf{x}_\star \end{bmatrix} \in \mathbb{R}^{N \times \left(m+1\right)}$ after the addition of column $\mathbf{x}_\star \in \mathbb{R}^N$ at the end of matrix $\mathbf{X}$ (i.e. at position $m+1$) is

$$\begin{bmatrix} \mathbf{Q}_1^\intercal \\ \mathbf{Q}_2^\intercal \end{bmatrix} \mathbf{X}^+ = \begin{bmatrix} \mathbf{R}_1 & \mathbf{z}_{\star 1} \\ \mathbf{0} & \mathbf{z}_{\star 2} \end{bmatrix} = \widetilde{\mathbf{R}}_1^+,$$

where $\mathbf{z}_{\star 1} = \mathbf{Q}_1^\intercal \mathbf{x}_\star$ and $\mathbf{z}_{\star 2} = \mathbf{Q}_2^\intercal \mathbf{x}_\star$. Matrix $\mathbf{R}_1^+$ can be obtained by setting to zero the last $N - m - 1$ elements of the last column of $\widetilde{\mathbf{R}}_1^+$, as shown in Figure 2.D.1. This may be achieved by pre-multiplying $\widetilde{\mathbf{R}}_1^+$ by the appropriate set of Givens matrices. However, this procedure requires the evaluation of matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$. In order to avoid such computation, $\mathbf{z}_{\star 1}$ can be determined by solving the linear system $\mathbf{R}_1^\intercal \mathbf{z}_{\star 1} = \mathbf{X}^\intercal \mathbf{x}_\star$, while the element $\mathbf{R}_1^+[m+1, m+1]$ can be computed exploiting the relation $\mathbf{R}_1^{+\intercal}[, m+1]\mathbf{R}_1^+[, m+1] = \mathbf{x}_\star^\intercal \mathbf{x}_\star$, therefore

$$\mathbf{R}_1^+ [m+1, m+1] = \sqrt{\mathbf{x}_\star^\intercal \mathbf{x}_\star - \sum_{i=1}^{p} \mathbf{z}_{\star 1}^2[i]}.$$

This update takes $\mathcal{O}\big(Nm\big)$ operations. The algorithm is shown in Algorithm 6.

Note that the thinQR decomposition only allows the update of $\mathbf{R}_1$ following the addition of the column at the end. However, in our case this is not an issue, as marginal posterior distribution $m\big(\boldsymbol{\gamma}|\mathbf{y},\mathbf{X}\big)$ is invariant with respect to the ordering of the variables. Therefore, it is assumed that they are added at the end of the design matrix.

$$\widetilde{\mathbf{R}}_1 = \begin{bmatrix} + & + & + & + & \mathbf{z}_{\star 1}[1] \\ 0 & + & + & + & \mathbf{z}_{\star 1}[2] \\ 0 & 0 & + & + & \mathbf{z}_{\star 1}[3] \\ 0 & 0 & 0 & + & \mathbf{z}_{\star 1}[4] \\ 0 & 0 & 0 & 0 & \mathbf{z}_{\star 2}[1] \\ 0 & 0 & 0 & 0 & \mathbf{z}_{\star 2}[2] \\ 0 & 0 & 0 & 0 & \mathbf{z}_{\star 2}[3] \end{bmatrix} \rightarrow \begin{bmatrix} + & + & + & + & \mathbf{z}_{\star 1}[1] \\ 0 & + & + & + & \mathbf{z}_{\star 1}[2] \\ 0 & 0 & + & + & \mathbf{z}_{\star 1}[3] \\ 0 & 0 & 0 & + & \mathbf{z}_{\star 1}[4] \\ 0 & 0 & 0 & 0 & \widetilde{z}_{\star 2}[1] \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \mathbf{R}_1^+$$

Figure 2.D.1: Add one column at the end with $N = 7$ and $m = 4$; vector $\mathbf{z}_{\star 1}$ is computed by solving the linear system $\mathbf{R}_1^\intercal \mathbf{z}_{\star 1} = \mathbf{X}^\intercal \mathbf{x}_\star$.

Deleting one column

Let $\mathbf{X}^- = \big[\mathbf{X}\,[,1:(k-1)\big]\quad \mathbf{X}\,[,(k+1):m]\big] \in \mathbb{R}^{N\times\binom{m-1}{}}$ be the reduced form of $\mathbf{X}$ after the deletion of column $k = \{1,\ldots,m\}$, then

$$\widetilde{\mathbf{R}}_1^- = \big[\mathbf{R}_{1k^-} \quad \mathbf{R}_{1k^+}\big]\,,$$

where $\mathbf{R}_{1k^-} = \mathbf{R}_1[,1:(k-1)]$ and $\mathbf{R}_{1k^+} = \mathbf{R}_1[,(k+1):m]$ are upper trapezoidal. Updated matrix $\mathbf{R}_1^-$ can be obtained by setting to 0 the $m-k$ elements on the sub-diagonal in $\mathbf{R}_{1k^+}$, as shown in Figure 2.D.2. This can be achieved by pre-multiplying matrix $\widetilde{\mathbf{R}}_1^-$ for the sequence of Given rotations given by $\mathbf{G}_m(m-1,m)^\intercal \times \cdots \times \mathbf{G}_{k+1}(k,k+1)^\intercal$, leading to

$$\begin{bmatrix} \mathbf{R}_1^- \\ \mathbf{0}^\intercal \end{bmatrix} = \mathbf{G}_{m-1}(m-1,m)^\intercal \times \cdots \times \mathbf{G}_k(k,k+1)^\intercal \widetilde{\mathbf{R}}_1^-.$$

The number of operations required for this update is $\mathcal{O}\big((m-k)^2\big)$, which becomes 0 if $k = m$. The algorithm is shown in Algorithm 7.

2.D.3 Adding and deleting a block of columns

Here we consider the update of triangular matrix $\mathbf{R}_1$ following the addition of a block of columns $\mathbf{X}_\star \in \mathbb{R}^{N\times d}$ at the end or the deletion of a block of (adjacent and non-adjacent) columns $\mathbf{X}_k \in \mathbb{R}^{N\times d}$ from position $k = \{1,\ldots,m-d+1\}$.

$$\mathbf{R}_1 = \begin{bmatrix} + & + & + & + \\ 0 & + & + & + \\ 0 & 0 & + & + \\ 0 & 0 & 0 & + \end{bmatrix} \rightarrow \begin{bmatrix} + & + & + \\ 0 & + & + \\ 0 & \odot & + \\ 0 & 0 & \odot \end{bmatrix} = \mathbf{R}_1^-$$

Figure 2.D.2: Delete one column with $N = 7$, $m = 4$ and $k = 2$; symbol $\odot$ denotes an element set to zero with a Givens rotation.

Adding a block of columns

The thinQR update of the augmented matrix $\mathbf{X}^+ = \begin{bmatrix} \mathbf{X} & \mathbf{X}_\star \end{bmatrix} \in \mathbb{R}^{N \times (m+d)}$ after the addition of a block of columns $\mathbf{X}_\star \in \mathbb{R}^{N \times d}$ at the end of matrix $\mathbf{X}$ (i.e. at position $m+1$) is

$$\begin{bmatrix} \mathbf{Q}_1^\intercal \\ \mathbf{Q}_2^\intercal \end{bmatrix} \mathbf{X}^+ = \begin{bmatrix} \mathbf{R}_1 & \mathbf{Z}_{\star 1} \\ \mathbf{0} & \mathbf{Z}_{\star 2} \end{bmatrix} = \widetilde{\mathbf{R}}_1^+,$$

where $\mathbf{Z}_{\star 1} = \mathbf{Q}_1^\intercal \mathbf{X}_\star$ and $\mathbf{Z}_{\star 2} = \mathbf{Q}_2^\intercal \mathbf{X}_\star$. Matrix $\mathbf{R}_1^+$ can be obtained through triangularization of matrix $\mathbf{Z}_{\star 2}$, as shown in Figure 2.D.3. This may be achieved by pre-multiplying $\widetilde{\mathbf{R}}_1^+$ by the appropriate set of Givens matrices. However, this procedure requires the evaluation of matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$. In order to avoid such computation, $\mathbf{Z}_{\star 1}$ can be determined by solving the linear system $\mathbf{R}_1^\intercal \mathbf{Z}_{\star 1} = \mathbf{X}^\intercal \mathbf{X}_\star$. Entries $\mathbf{R}_1^+ [m+i, m+j]$, for $i = 1, \ldots, d$ and $j \geq i, \ldots, d$ can be computed by iteratively exploiting the relationship $\mathbf{X}^{+\intercal} \mathbf{X}^+ = \mathbf{R_1}^{+\intercal} \mathbf{R_1}^+$, see Algorithm 8. This update takes $\mathcal{O}(dNm)$ operations.

As for the case $d = 1$, note that the thinQR decomposition only allows the update of $\mathbf{R}_1$ following the addition of the block of columns at the end. However, in our case this is not an issue, as marginal posterior distribution $m(\boldsymbol{\gamma}|\mathbf{y}, \mathbf{X})$ is invariant with respect to the ordering of the variables. Therefore, it is assumed that they are added at the end of the design matrix.

Deleting a block of adjacent columns

Let $\mathbf{X}^- = \begin{bmatrix} \mathbf{X}[, 1 : (k-1)] & \mathbf{X}[, (k+d) : m] \end{bmatrix} \in \mathbb{R}^{N \times (m-d)}$ be the reduced form of $\mathbf{X}$ after the deletion of a block of $d < m$ adjacent columns starting from position $k = \{1, \ldots, m-d+1\}$, then

$$\widetilde{\mathbf{R}}_1^- = \begin{bmatrix} \mathbf{R}_{1k^-} & \mathbf{R}_{1k^+} \end{bmatrix},$$

where $\mathbf{R}_{1k^-} = \mathbf{R}_1[, 1 : (k-1)]$ and $\mathbf{R}_{1k^+} = \mathbf{R}_1[, (k+d) : m]$ are upper trapezoidal. Updated matrix $\mathbf{R}^-$ can be obtained through triangularization of matrix $\mathbf{R}_{1k^+}$, as

$$\widetilde{\mathbf{R}}_1 = \underbrace{\begin{bmatrix} + & + & + & + & \mathbf{Z}_{\star 1}[1,1] & \mathbf{Z}_{\star 1}[1,2] \\ 0 & + & + & + & \mathbf{Z}_{\star 1}[2,1] & \mathbf{Z}_{\star 1}[2,2] \\ 0 & 0 & + & + & \mathbf{Z}_{\star 1}[3,1] & \mathbf{Z}_{\star 1}[3,2] \\ 0 & 0 & 0 & + & \mathbf{Z}_{\star 1}[4,1] & \mathbf{Z}_{\star 1}[4,2] \\ 0 & 0 & 0 & 0 & \mathbf{Z}_{\star 2}[1,1] & \mathbf{Z}_{\star 2}[1,2] \\ 0 & 0 & 0 & 0 & \mathbf{Z}_{\star 2}[2,1] & \mathbf{Z}_{\star 2}[2,2] \\ 0 & 0 & 0 & 0 & \mathbf{Z}_{\star 2}[3,1] & \mathbf{Z}_{\star 2}[3,2] \end{bmatrix}}_{\mathbf{R}_1^+} \rightarrow \begin{bmatrix} + & + & + & + & \mathbf{Z}_{\star 1}[1,1] & \mathbf{Z}_{\star 1}[1,2] \\ 0 & + & + & + & \mathbf{Z}_{\star 1}[2,1] & \mathbf{Z}_{\star 1}[2,2] \\ 0 & 0 & + & + & \mathbf{Z}_{\star 1}[3,1] & \mathbf{Z}_{\star 1}[3,2] \\ 0 & 0 & 0 & + & \mathbf{Z}_{\star 1}[4,1] & \mathbf{Z}_{\star 1}[4,2] \\ 0 & 0 & 0 & 0 & \widetilde{\mathbf{Z}}_{\star 2}[1,1] & \widetilde{\mathbf{Z}}_{\star 2}[1,2] \\ 0 & 0 & 0 & 0 & 0 & \widetilde{\mathbf{Z}}_{\star 2}[2,2] \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} =$$

Figure 2.D.3: Add a block of columns at the end with $N = 7$, $m = 4$ and $d = 2$; matrix $\mathbf{Z}_{\star 1}$ is computed by solving the linear system $\mathbf{R}_1^\intercal \mathbf{Z}_{\star 1} = \mathbf{X}^\intercal \mathbf{X}_\star$.

shown in Figure 2.D.4. This can be achieved by pre-multiplying matrix $\widetilde{\mathbf{R}}_1^-$ for a set of Householder reflections as follows

$$\begin{bmatrix} \mathbf{R}_1^- \\ \mathbf{0}^\intercal \end{bmatrix} = \mathbf{H}_{m-d}(m-d+1, m) \times \cdots \times \mathbf{H}_k(k+1, k+d)\widetilde{\mathbf{R}}_1$$

where $\mathbf{H}_j(l, n)$, $j = k, \ldots, m-d$, is the Householder matrix with normal vector $\mathbf{v}_j(l, n) \in \mathbb{R}^m$ defined as in equation (2.D.1). Eventually, the upper triangular sub-matrix $\mathbf{R}_1^- \in \mathbb{R}^{(m-d)\times(m-d)}$ is selected.

The number of operations required for this update is $\mathcal{O}\big(dm^2\big)$ if $k \in \{1, \ldots, m-d\}$ and 0 if $k = m - d + 1$. The algorithm is show in Algorithm 9.

$$\mathbf{R}_1 = \begin{bmatrix} + & + & + & + & + \\ 0 & + & + & + & + \\ 0 & 0 & + & + & + \\ 0 & 0 & 0 & + & + \\ 0 & 0 & 0 & 0 & + \end{bmatrix} \rightarrow \begin{bmatrix} + & + & + \\ 0 & + & + \\ 0 & \odot & + \\ 0 & \odot & \odot \\ 0 & 0 & \odot \end{bmatrix} = \mathbf{R}_1^-$$

Figure 2.D.4: Delete a block of columns with $N = 7$, $m = 5$, $k = 2$ and $d = 2$; symbol $\odot$ denotes an elements set to zero with Householder reflections.

Deleting a block of non-adjacent columns

Let $\mathbf{X}^- \in \mathbb{R}^{N\times(m-d)}$ be the reduced form of $\mathbf{X}$ after the deletion of $d < m$ non-adjacent columns in positions $k_1, \ldots, k_d$, then updated matrix $\mathbf{R}_1^-$ can be obtained through triangularization of matrix $\mathbf{R}_1$ after the deletion of columns $k_1, \ldots, k_d$. Following the case for d adjacent columns, this can be done by applying either Givens rotations or Householder reflections, depending on the number of elements

below the main diagonal that are set to zero (see Figure 2.D.5). In this case, Givens rotations are applied to $\mathbf{R}^-$ when setting to zero one element below the new diagonal of $\mathbf{R}^-$, whereas Householder reflections are applied when setting to zero more than one element below the main diagonal.

$$\mathbf{R}_1 = \begin{bmatrix} + & + & + & + & + \\ 0 & + & + & + & + \\ 0 & 0 & + & + & + \\ 0 & 0 & 0 & + & + \\ 0 & 0 & 0 & 0 & + \end{bmatrix} \rightarrow \begin{bmatrix} + & + \\ \odot & + \\ 0 & \odot \\ 0 & \odot \\ 0 & \odot \end{bmatrix} = \mathbf{R}_1^-$$

Figure 2.D.5: Delete a block of non-adjacent columns with $N = 7$, $m = 5$, $d = 3$, $k_1 = 1$, $k_2 = 3$ and $k_3 = 4$; symbol $\odot$ denotes an elements set to zero with Givens rotations or Householder reflections.

2.D.4 ThinQR update algorithms

Algorithm 4: Householder reflection, $(\tau, \mathbf{v}, \mu) = \mathsf{householder}\big(a, \mathbf{x}\big)$

Input: $a \in \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^N$;
$s = \|\mathbf{x}\|_2^2, \quad \mathbf{v} = \begin{bmatrix} 1 & \mathbf{x}^\top \end{bmatrix}^\top$;
if $(s == 0)$ & $(a == 0)$ **then** $\tau = 0$;
if $(s == 0)$ & $(a > 0)$ **then**
    $\tau = -2$;
else
    $\mu = \sqrt{s + a^2}$;
    if $(a \leq 0)$ **then**
        $\mathbf{v}\,[1] = a - \mu$;
    else
        $\mathbf{v}\,[1] = -s/(a + \mu)$;
    end
end
$b = (\mathbf{v}[1])^2$, $\tau = 2b/(s + b), \quad \mathbf{v} = \begin{bmatrix} 1 & \big(\mathbf{v}[2 : (N+1)]/\mathbf{v}[1]\big)^\top \end{bmatrix}^\top$;
return $(\tau, \mathbf{v}, \mu)$;

Algorithm 5: Givens rotation, $(c, s) = \mathsf{givens}\big(a, b\big)$

Input: $a \in \mathbb{R},\ b \in \mathbb{R}$;
if $(b == 0)$ **then**
    $c = 1$;
    $s = 0$;
else
    if $(|b| > |a|)$ **then**
        $r = -a/b$;
        $s = 1/\sqrt{1 + r^2}$;
        $c = s * r$;
        if $(b > 0)$ **then** $c = -c,\ s = -s$;
    else
        $r = -b/a$;
        $c = 1/\sqrt{1 + r^2}$;
        $s = c * r$;
        if $(a < 0)$ **then** $c = -c,\ s = -s$;
    end
end
return (c, s);

Algorithm 6: ThinQR update when a column is added at position $k = m + 1$, $\mathbf{R}_1^+ = \mathsf{thinqraddcol}\big(\mathbf{R}_1, \mathbf{X}, \mathbf{u}\big)$

Input: $\mathbf{R}_1 \in \mathbb{R}^{m \times m},\ \mathbf{X} \in \mathbb{R}^{N \times m},\ \mathbf{u} \in \mathbb{R}^N$;
`// add one column`
Solve $\mathbf{R}_1^\intercal \mathbf{r}_{12} = \mathbf{X}^\intercal \mathbf{u}$ with respect to $\mathbf{r}_{12}$ with forward substitution algorithm;
$\mathbf{R}_1 = \begin{bmatrix} \mathbf{R}_1 & \mathbf{r}_{12} \\ \mathbf{0}_{1 \times m} & 0 \end{bmatrix}$;
`// update` $\mathbf{R}_1$
$\mathbf{R}_1\,[m+1, m+1] = \|\mathbf{u}\|_2^2 - \|\mathbf{r}_{12}\|_2^2$;
$\mathbf{R}_1\,[m+1, m+1] = \sqrt{|\mathbf{R}_1\,[m+1, m+1]\,|}$;
return $\mathbf{R}_1$;

Algorithm 7: ThinQR update when a column is deleted at position $1 \leq k \leq m$, $\mathbf{R}_1^- = \mathsf{thinqrdelcol}\big(\mathbf{R}_1, k\big)$

Input: $\mathbf{R}_1 \in \mathbb{R}^{m \times m}$, $k \in \{1, \ldots, m\}$;
// delete one column
if $(k = m)$ **then return** $\mathbf{R}_1\,[1:(m-1), 1:(m-1)]$;
$\mathbf{R}_1\,[, k:(m-1)] = \mathbf{R}_1\,[, (k+1):m]$;
for $\big(i = k;\ i < p;\ i++\big)$ **do**
    $(c, s) = \mathsf{givens}\big(\mathbf{R}_1\,[i, i]\,, \mathbf{R}_1\,[i+1, i]\,\big)$ as in Algorithm 5;
    $\mathbf{G} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$;
    // update $\mathbf{R}_1$
    $\mathbf{R}_1\,[i, i] = c * \mathbf{R}_1\,[i, i] - s * \mathbf{R}_1\,[i+1, i]$;
    $\mathbf{R}_1\,[i+1, i] = 0$;
    if $(i < m-1)$ **then**
        $\mathbf{R}_1\,[i:i+1, (i+1):(m-1)] = \mathbf{G}^\intercal \mathbf{R}_1\,[i:i+1, (i+1):(m-1)]$;
    end
end
return $\mathbf{R}_1 = \mathbf{R}_1\,[1:(m-1), 1:(m-1)]$;

Algorithm 8: ThinQR update when $d \geq 2$ columns are added from position $k = m + 1$ to $k + d - 1$, $\mathbf{R}_1^+ = \mathsf{thinqraddblockcols}\big(\mathbf{R}_1, \mathbf{X}, \mathbf{U}\big)$

Input: $\mathbf{R}_1 \in \mathbb{R}^{m \times m}$, $\mathbf{X} \in \mathbb{R}^{N \times m}$, $\mathbf{U} \in \mathbb{R}^{N \times d}$;

`// add` d `columns`

`// compute` $\mathbf{R}_{12}$

Solve $\mathbf{R}_1^\intercal \mathbf{R}_{12} = \mathbf{X}^\intercal \mathbf{U}$ with respect to $\mathbf{R}_{12}$ with forward substitution algorithm;

`// compute` $\mathbf{R}_{22}$

$\mathbf{R}_{22} = \mathbf{0}_{d \times d}$;

$\mathbf{R}_{22}\,[1,1] = \sqrt{\|\mathbf{U}[,1]\|_2^2 - \|\mathbf{R}_{12}[,1]\|_2^2}$;

$\mathbf{R}_{22}\,[1, 2:d] = \big(\mathbf{U}\,[,1]^\intercal\,\mathbf{U}\,[,2:d] - \mathbf{R}_{12}\,[,1]^\intercal\,\mathbf{R}_{12}\,[,2:d]\,\big)/\mathbf{R}_{22}\,[1,1]$;

for $\big(i = 2;\ i \leq d;\ i++\big)$ **do**

  $\mathbf{R}_{22}\,[i,i] = \sqrt{|\|\mathbf{U}\,[,i]\,\|_2^2 - \|\mathbf{R}_{12}\,[,i]\,\|_2^2 - \|\mathbf{R}_{22}\,[1:(i-1),i]\,\|_2^2|}$;

  if $\big(i < d\big)$ **then** $\mathbf{R}_{22}\,[i,(i+1):d] = \big(\mathbf{U}\,[,i]^\intercal\,\mathbf{U}\,[,(i+1):d] - \mathbf{R}_{12}\,[,i]^\intercal\,\mathbf{R}_{12}\,[,(i+1):d] - \mathbf{R}_{22}\,[1:(i-1),i]^\intercal\,\mathbf{R}_{22}\,[1:(i-1),(i+1):d]\,\big)/\mathbf{R}_{22}\,[i,i]$;

end

$$\mathbf{R}_1 = \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_{12} \\ \mathbf{0}_{d \times m} & \mathbf{R}_{22} \end{bmatrix};$$

return $\mathbf{R}_1$;

Algorithm 9: ThinQR update when $2 \leq d < m$ columns are deleted from position $1 \leq k \leq m-d+1$ to $k+d-1$, $\mathbf{R}_1^- = \mathsf{thinqrdelblockcols}\big(\mathbf{R}_1, k, d\big)$

Input: $\mathbf{R}_1 \in \mathbb{R}^{m \times m}$, $k \in \{1, \ldots, m-d+1\}$, $d \in \{2, \ldots, m+1-k\}$;

// delete d columns

if $(k = m-d+1)$ **then return** $\mathbf{R}_1\,[1:(m-d), 1:(m-d)]$;

// permute columns

$\mathbf{R}_1\,[, k:(m-d)] = \mathbf{R}_1\,[, (k+d):m]$;

for $\big(i = k;\ i \leq m-d-1;\ i++\big)$ **do**

  $(\tau, \mathbf{v}, \mu) = \mathsf{householder}(\mathbf{R}_1\,[i, i]\,, \mathbf{R}_1\,[(i+1):(i+d), i])$ as in Algorithm 4;

  $\mathbf{R}_1\,[i, i] = \mu$;

  $\mathbf{R}_1\,[(i+1):(i+d), i] = \mathbf{0}_d$;

  $\mathbf{R}_1\,[i:(i+d), (i+1):(m-d)] =$

  $\quad \mathbf{R}_1\,[i:(i+d), (i+1):(m-d)] - (\tau * \mathbf{v})\big(\mathbf{v}^\intercal \mathbf{R}_1\,[i:(i+d), (i+1):(m-d)]\big)$;

end

// update $\mathbf{R}_1\,[m-d, m-d]$

$\mathbf{R}_1\,[m-d, m-d] = \sqrt{\|\mathbf{R}_1\,[(m-d):m, m]\,\|_2^2}$;

return $\mathbf{R}_1\,[1:(m-d), 1:(m-d)]$;

Algorithm 10: Apply either Givens rotation or Householder reflection to column i, $\mathbf{R}_1 = \mathsf{thinqrstep}\big(\mathbf{R}_1, i, a\big)$

Input: $\mathbf{R}_1 \in \mathbb{R}^{m \times l}$, $i \in \{1, \ldots, l-1\}$, $a \in \{1, \ldots, m-i\}$;
if $(a > 1)$ **then**
    `// Householder reflection`
    $(\tau, \mathbf{v}, \mu) = \mathsf{householder}\big(\mathbf{R}_1\left[i, i\right], \mathbf{R}_1\left[(i+1) : (i+a), i\right]\big)$ as in Algorithm 4;
    $\mathbf{R}_1\left[i, i\right] = \mu$;
    $\mathbf{R}_1\left[i : (i+a), (i+1) : l\right] = \mathbf{R}_1\left[i : (i+a), (i+1) : l\right] - \big(\tau * \mathbf{v}\big)\big(\mathbf{v}^\intercal \mathbf{R}_1\left[i : (i+a), (i+1) : l\right]\big)$;
    $\mathbf{R}_1\left[(i+1) : (i+a), i\right] = \mathbf{0}_a$;
else
    $(c, s) = \mathsf{givens}\big(\mathbf{R}_1\left[i, i\right], \mathbf{R}_1\left[i+1, i\right]\big)$ as in Algorithm 5;
    $\mathbf{G} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$;
    `// update` $\mathbf{R}_1$
    $\mathbf{R}_1\left[i, i\right] = c * \mathbf{R}_1\left[i, i\right] - s * \mathbf{R}_1\left[i+1, i\right]$;
    $\mathbf{R}_1\left[i+1, i\right] = 0$;
    $\mathbf{R}_1\left[i : (i+1), (i+1) : l\right] = \mathbf{G}^\intercal \mathbf{R}_1\left[i : (i+1), (i+1) : l\right]$;
end
return $\mathbf{R}_1$;

Algorithm 11: ThinQR update when d non-adjacent columns are deleted, $\mathbf{R}_1^- = \mathsf{thinqrdelblockcols\_nonadj}\big(\mathbf{R}_1, \mathbf{k}\big)$

Input: $\mathbf{R}_1 \in \mathbb{R}^{m \times m}$, $\mathbf{k}[i] \in \{1, \ldots, m\}, i = 1, \ldots, d$, $k[i] < k[j] \, \forall \, i < j, i, j = 1, \ldots, d$;
// delete d columns
if $\big(d = 1\big)$ **then return** $\mathsf{thinqrdelcol}\big(\mathbf{R}_1, \mathbf{k}\,[1]\,\big)$ in Algorithm 7;
if $\big(\big(\mathbf{k}\,[d] - \mathbf{k}\,[1]\,\big) = (d-1)\big)$ **then return** $\mathsf{thinqrdelblockcols}\big(\mathbf{R}_1, \mathbf{k}\,[1]\,, d\big)$ in Algorithm 9;
$\mathbf{e} = 1 : m$;
$\overline{\mathbf{k}} = \mathbf{e} \setminus \mathbf{k}$;
$l = \overline{\mathbf{k}}[m - d]$;
$q = d - (m - l)$;
$\mathbf{k} = \mathbf{k}[1 : q]$;
$\mathbf{R}_1 = \mathbf{R}_1\,[1 : l, 1 : l]$;
if $\big(q = 1\big)$ **then return** $\mathsf{thinqrdelcol}\big(\mathbf{R}_1, \mathbf{k}\,[1]\,\big)$ in Algorithm 7;
if $\big(\big(\mathbf{k}\,[q] - \mathbf{k}\,[1]\,\big) = (q-1)\big)$ **then return** $\mathsf{thinqrdelblockcols}\big(\mathbf{R}_1, \mathbf{k}\,[1]\,, q\big)$ in Algorithm 9;
// delete columns
$\mathbf{R}_1 = \mathbf{R}_1\,\big[, \overline{\mathbf{k}}\big]$;
$\overline{\mathbf{k}} = \overline{\mathbf{k}}\,[\mathbf{k}\,[1] : (l - q)]$;
// compute $\mathbf{a}[1]$
$\mathbf{a} = \mathbf{0}_{l-q-\mathbf{k}[1]+1}; \quad \mathbf{a}[1] = \overline{\mathbf{k}}[1] - \mathbf{k}[1]$;
// update $\mathbf{R}_1$
for $\big(i = 1; \, i \leq (l - q - \mathbf{k}[1]); \, i++\big)$ **do**
    $\mathbf{R}_1 = \mathsf{thinqrstep}\big(\mathbf{R}_1, i + \mathbf{k}[1] - 1, \mathbf{a}[i]\big)$ in Algorithm 10;
    $\mathbf{a}[i+1] = \mathbf{a}[i] + \big(\overline{\mathbf{k}}[i+1] - \overline{\mathbf{k}}[i]\big) - 1$
end
$\mathbf{R}_1[l-q-\mathbf{k}[1]+1, l-q-\mathbf{k}[1]+1] = \sqrt{\|\mathbf{R}_1[(l-q-\mathbf{k}[1]+1) : (l-\mathbf{k}[1]+1), l-q-\mathbf{k}[1]+1]\|_2^2}$;
return $\mathbf{R}_1[1 : (l-q),]$;

Chapter 3

Multiple graphical horseshoe estimator for modeling correlated precision matrices

3.1 Introduction

Graphical models are a popular tool used in many scientific fields to analyze and infer networks. In the Gaussian setting, the main challenges in graph estimation are the positive-definiteness constraint on precision matrices (inverse-covariance matrices) and the quadratic growth, with respect to the number of variables included in the analysis, of the number of free parameters. Traditional methods, such as the ones based on pairwise model comparisons, become computationally infeasible as the number of considered variables increases. For exchangeable observations, a collection of the existing methods for high-dimensional covariance matrix estimation is available in Pourahmadi (2011), in which the author proposes to reduce the problem to multiple independent (penalized) least-squares regressions. Other common approaches, such as the Graphical LASSO of Friedman et al. (2008) and the Graphical SCAD of Fan et al. (2009), are based on a penalized likelihood optimization and provide a sparse solution for the precision matrix in high-dimensional settings. A few approaches for the estimation of high-dimensional sparse networks have also been proposed within the Bayesian framework. In particular, the Bayesian version of the Graphical LASSO (Wang, 2012), the spike and slab stochastic search method (Wang, 2015), and the more recent Graphical Horseshoe presented in Li et al. (2019); all Bayesian methods implemented a block Gibbs sampler that has shown good computational performances up to a few hundred variables.

We are interested in settings where observations can be considered exchange-

able only within groups; in these settings, a separate group-specific estimation will reduce the statistical power, while an analysis of data pooled across groups will lead to spurious findings (Peterson et al., 2015). Generalizations of the graphical models, called multiple graphical models, have been proposed with the aim of jointly estimating multiple correlated networks. Among the penalized likelihood approaches, the fused Graphical LASSO and the group Graphical LASSO of Danaher et al. (2014) rely on convex optimization problems and force similar edge values and similar graph structures, respectively. Bayesian approaches have been first proposed to encourage similar network structures across related subgroups (Peterson et al., 2015; Shaddox et al., 2018). More recent attempts, such as the generalization of the Bayesian spike and slab stochastic method of Peterson et al. (2020) and the GemBAG of Yang et al. (2021), focus on shared sparsity structures and precision matrix elements. See Ni et al. (2022) for a recent review of Bayesian approaches for complex graphical models, including methods for multiple groups.

Here we propose a generalization of the Graphical Horseshoe of Li et al. (2019) in the presence of multiple correlated sample groups, which we refer to as the *multiple Graphical Horseshoe* (mGHS). This model works under the multivariate gaussianity assumption with multiple dependent precision matrices. The proposed model is based on a novel prior on multiple covariance matrices that builds upon the Horseshoe prior proposed in Carvalho et al. (2010) and lets the data decide whether borrowing strength across groups and then encouraging similar precision matrices is appropriate. The properties of the Horseshoe prior are well-studied and include the improved Kullback-Leibler risk bound (Carvalho et al., 2010), minimaxity in estimation under the l_2 loss (Van der Pas et al., 2014) and improved risk properties in linear regression (Bhadra et al., 2016). Through simulation studies, we empirically show that the model benefits from the similar structures of the groups and provides better statistical performances than the Graphical Horseshoe applied separately to each group. The model relies on a Metropolis-within-Gibbs sampler where the parameters are updated by sampling from their full-conditional distributions and, in particular, a novel method is introduced in order to sample the local variance parameters. This method scales well with respect to the number of variables and is the first full Bayesian approach (to our knowledge) able to analyze multiple undirected graphs of hundreds of nodes. In Castelletti et al. (2020) the authors propose an efficient full Bayesian approach for multiple networks, however they consider only directed acyclic graphs (DAGs). Finally, we discuss a novel idea for posterior edge selection based on model cuts. The main novelties can be summarized as follow: 1) a novel shrinkage prior for multiple precision matrices, 2) an efficient algorithm that scales exceptionally well, and 3) a novel approach for edge selection based on model cuts.

The paper is organized as follows. In Section 3.2 the proposed sampling model

is introduced. Section 3.3 illustrates how to sample from a three-parameters Gamma distribution ($\mathcal{G}_{3p}$) with a modified rejection sampling approach. Section 3.4 outlines the proposed algorithm in detail. In Section 3.5 we present a novel proposal for model selection. Section 3.6 illustrates comparative simulation studies, whereas in Section 3.7 we present an application to a benchmark bike-sharing dataset. Discussions and comments are presented in Section 3.8.

3.2 The model

In this section, we introduce the sampling model used to infer relationships among variables within each of K possibly related sample groups, each represented by a graph $G_k = (V, E_k)$, where V corresponds to a set of vertices and E_k to a set of group-specific edges. Let $\mathbf{y}_{sk}$ be the p-dimensional random vector related to the observation s in group k, where $s = 1, \ldots, n_k$ and $k = 1, \ldots, K$. Under the multivariate normal distribution, the corresponding sampling model is

$$\mathbf{y}_{sk} \sim \mathcal{N}_p \left(\mathbf{0}, \mathbf{\Sigma}_k \right),$$

where $\mathbf{\Omega}_k \equiv (\omega_{ij}^k)_{p \times p} = \mathbf{\Sigma}_k^{-1}$ is the precision matrix of group k. There is a one-to-one correspondence between the zero patterns in a precision matrix and an undirected graph G_k that, in turn, can be used to learn conditional independencies. Specifically, it can be shown that $\omega_{ij}^k = 0$ if and only if variables i and j are conditionally independent conditioning on the other variables (Dempster, 1972); in this case, the undirected graph G_k will have a missing edge between nodes i and j. Therefore, the goal is the joint estimation of non-zero entries in precision matrices with the aim of capturing significant connections among variables. In high-dimensional settings, the number of parameters to be estimated in $\mathbf{\Omega}_k$ is of order $O\left(p^2\right)$. This task is particularly challenging since these precision matrices, in addition to being very large, are constrained to the cone of symmetric positive definite matrices. Building upon the Graphical Horseshoe proposed by Li et al. (2019), we propose in Sections 3.2.1 and 3.4 model and algorithm, respectively, that use shrinkage priors to perform full Bayesian inference of multiple related high-dimensional undirected graphical models.

3.2.1 An horseshoe prior for multiple related precision matrices

Li et al. (2019) have successfully developed the Graphical Horseshoe prior, a shrinkage prior for (single) precision matrices. In this section, we describe how to extend

the Graphical Horseshoe prior to multiple related precision matrices. The proposed approach will both achieve shrinkage and borrowing strength across related subgroups; as a key modeling feature, our approach will learn from the data which pairs of groups are related and which ones can be considered independent. With respect to the model proposed by Peterson et al. (2020), the only alternative full Bayesian approach that uses a joint prior on related multiple precision matrices, the proposed approach will result in a much more scalable algorithm, as detailed in Section 3.4.

Let $\boldsymbol{\omega}_{ij} = \left(\omega_{ij}^1, \ldots, \omega_{ij}^K\right)^\intercal$ be the vector of precision matrix entries corresponding to edge (i, j) across K groups. Our approach builds upon the Graphical Horseshoe prior (Li et al., 2019), as we shrink non-informative edges ω_{ij}^k with a novel multivariate Horseshoe prior (Carvalho et al., 2010); we assume a non-informative prior for diagonal entries ω_{jj}^k. The joint prior distribution for precision matrices $\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_K$ can be written as

$$\begin{aligned} \pi\left(\omega_{jj}^k\right) &\propto 1, \quad k = 1, \ldots, K, \quad j = 1, \ldots, p \\ \pi\left(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_K | \boldsymbol{\Psi}_{ij} : i < j\right) &\propto \prod_{i<j} \mathcal{N}_K\left(\boldsymbol{\omega}_{ij} | \mathbf{0}, \boldsymbol{\Psi}_{ij}\right) \cdot \mathbb{I}_{\left(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_K \in \mathbb{M}_+^p\right)} \end{aligned}$$

where $\mathbb{M}_+^p$ denotes the space of $p \times p$ positive-definite symmetric matrices. The proposed prior jointly models multiple precision matrices and, specifically, accounts for similarity between groups by imposing a K-variate normal prior distribution for $\boldsymbol{\omega}_{ij}$ with prior covariance matrix specific for each pair ij. As in Peterson et al. (2020), the proposed prior jointly learns both the within-group and across-group associations from the data in a single step, but it is computationally more efficient because it is based on continuous mixtures of multivariate normal distributions. Indeed, there is no need to sample the binary edge inclusion indicators as in Peterson et al. (2020).

Following the *separation strategy* introduced by Barnard et al. (2000), the across-group covariance matrices $\boldsymbol{\Psi}_{ij}$ can be decomposed as $\boldsymbol{\Psi}_{ij} = \boldsymbol{\Delta}_{ij}\mathbf{R}\boldsymbol{\Delta}_{ij}$, where $\boldsymbol{\Delta}_{ij} = \text{diag}\{\delta_{ij,1}, \ldots, \delta_{ij,K}\}$ contains the standard deviations of edge (i, j) and $\mathbf{R} = \{r_{k'k} : k' < k\} \in \mathbb{M}_+^K$ is a valid correlation matrix with diagonal entries equal to one. As suggested by Barnard et al. (2000), we model variances $\delta_{ij,k}$ and correlations $r_{k'k}$ separately since it is generally not clear how these elements interact with each other. We apply the Horseshoe prior from Carvalho et al. (2010) by decomposing $\delta_{ij,k} = \tau_k \lambda_{ij,k}$ and imposing the following priors:

$$\lambda_{ij,k} \sim \mathcal{C}^+(0, 1), \tag{3.1}$$

$$\tau_k \sim \mathcal{C}^+(0, 1), \tag{3.2}$$

where $\mathcal{C}^+$ denotes the positive half-Cauchy distribution. In (3.1) and (3.2), parameters τ_k and $\lambda_{ij,k}$ control the global and local shrinkage of ω_{ij}^k, respectively.

The heavy-tail distribution of $\lambda_{ij,k}$ allows ω_{ij}^k to avoid overshrinkage and lets the coefficients free to reach larger values. The amount of common shrinkage shared by the entries ω_{ij}^k is then controlled by the global scale parameter τ_k. When $K = 1$, the proposed model reduces to the Graphical Horseshoe of Li et al. (2019).

The selection of the prior distribution for correlation matrix $\mathbf{R}$ is often more complicated. Barnard et al. (2000) give an overview of the most common prior for a correlation matrix. Here we follow Peterson et al. (2020) and choose the prior distribution

$$\pi(\mathbf{R}) \propto 1 \cdot \mathbb{I}_{\left(\mathbf{R} \in \mathbb{C}_+^K\right)},$$

where $\mathbb{C}_+^K$ denotes the space of $K \times K$ definite-positive correlation matrices with diagonal entries equal to 1. The matrix $\mathbf{R}$ allows the local variances λ_{ij}^k to share information between each other when the correlations between groups are large. On the contrary, the model reduces to the Graphical Horseshoe of Li et al. (2019) applied separately to each group when $\mathbf{R} = \mathbf{I}_K$ is the identity matrix. In Section 3.3 we introduce a new sampling algorithm for the three-parameter Gamma distribution that will be used within the algorithm for posterior inference detailed in Section 3.4.

3.3 The three-parameter Gamma distribution and a modified rejection sampling algorithm

In this section, we introduce a modified acceptance-rejection method designed to generate samples from the three-parameter Gamma ($\mathcal{G}_{3p}$) distribution. Ahrens and Dieter (1982) and Stadlober (1982) demonstrated how to apply a rejection sampling for a target distribution when no valid proposal distribution is available. In particular, they proposed a modified rejection sampling to sample from a Gamma distribution and a t-Student distribution, respectively. Here the same situation applies since no trivial distribution, such as Gaussian or Gamma distributions, can be used as a valid proposal distribution. Indeed, it can be shown that these densities do not cover the target function on the latter's support, as required by the standard rejection sampling method. Therefore, we propose to overcome this problem by applying a modified rejection sampling with a Gaussian proposal distribution. The technical and theoretical aspects of this approach are detailed in Appendix 3.A, where we also provide a proof that the method proposed in this section draws samples from the target distribution (3.3). For the sake of clarity, the notation used in this section does not relate to the notation used in the other sections.

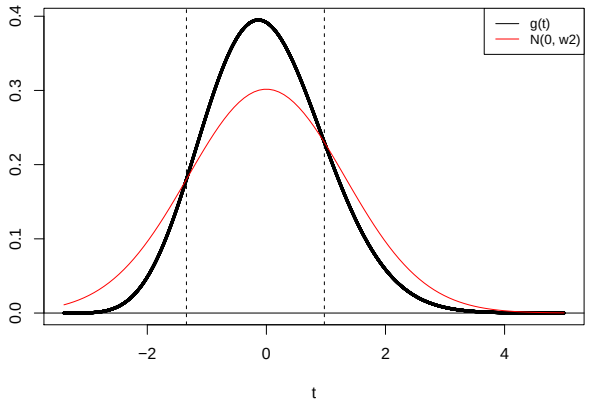


Figure 3.3.1: density g and h with $\gamma = 4$, $\alpha = 2.75$, $\beta = 3.3$; dotted lines represent t_1 and t_2.

Let $X \sim \mathcal{G}_{3p}(\gamma, \alpha, \beta)$, $\alpha, \beta \neq 0$, $\gamma \in \mathbb{N}^+$, a random variable with density

$$f_X(x \mid \gamma, \alpha, \beta) = \frac{e^{-\frac{\beta^2}{8\alpha^2}} (2\alpha^2)^{\frac{\gamma+1}{2}}}{\Gamma(\gamma+1) D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)} x^\gamma e^{-\alpha^2 x^2 + \beta x} \cdot \mathbb{I}_{(x>0)}, \tag{3.3}$$

where $D_a(b)$ is the Parabolic Cylinder function with parameters a and b. The mean and variance of variable X are

$$E(X) \equiv \mu = \frac{\gamma+1}{\alpha\sqrt{2}} \frac{D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}$$

$$Var(X) \equiv \sigma^2 = \frac{(\gamma+1)(\gamma+2)}{2\alpha^2} \frac{D_{-\gamma-3}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)} - \frac{(\gamma+1)^2}{2\alpha^2} \frac{D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)^2}{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)^2}.$$

The density $f(x) \sim \mathcal{G}_{3p}(\gamma, \alpha, \beta)$ is transformed into a standardized distribution $g(t) = \sigma f(\sigma t + \mu)$ by the transformation $t = (x - \mu)/\sigma$, with support on the interval $\left(-\frac{\mu}{\sigma}, \infty\right)$. A new value $t_\star$ can be drawn from $g(t)$ using the modified rejection sampling described below. Finally, the value $x_\star = \sigma t_\star + \mu$ is returned.

Consider the proposal distribution $h(t) \sim \mathcal{N}(0, \omega^2)$ and the ratio

$$\begin{aligned} r(t_\star) = \frac{g(t_\star)}{h(t_\star)} &= \frac{\sigma f(\sigma t_\star + \mu)}{h(t_\star)} \\ &= \omega\sigma C_f \sqrt{2\pi} (\sigma t_\star + \mu)^\gamma e^{-\alpha^2(\sigma t_\star+\mu)^2+\beta(\sigma t_\star+\mu)-\frac{t_\star^2}{2\omega^2}} \cdot \mathbb{I}_{\left(t_\star > -\frac{\mu}{\sigma}\right)} \\ &= \omega\sigma C_f \sqrt{2\pi} (\sigma t_\star + \mu)^\gamma e^{\left(\frac{1}{2\omega^2}-\alpha^2\sigma^2\right)t_\star^2+\left(\beta-2\mu\alpha^2\right)\sigma t_\star+\beta\mu-\alpha^2\mu^2}, \end{aligned} \tag{3.4}$$

where C_f is the normalizing constant of $f(x)$ and $(\beta - 2\mu\alpha^2) < 0$. The analysis of $r(t)$ gives insights on how to correctly choose the variance ω^2 of the proposal distribution $h(t)$, as $r(t)$ needs to be bounded and should go to zero as t increases. For this reason we set the variance to $w^2 = \frac{1}{2\alpha^2\sigma^2}$ and the ratio in (3.4) evaluated at $t_\star$ can be re-written as

$$r(t_\star) = \omega\sigma C_f \sqrt{2\pi} (\sigma t_\star + \mu)^\gamma e^{\left(\beta-2\mu\alpha^2\right)(\sigma t_\star+\mu)+\alpha^2\mu^2},$$

which is analytically tractable. In order to apply a standard rejection sampling, the method requires that $r(t_\star) \leq 1$. However, as shown in Figure 3.3.1, the proposal density $h(t)$ lays below the target density $g(t)$ in the interval $[t_1, t_2]$, with

$$\begin{aligned} t_1 &= \frac{\gamma}{\sigma(\beta - 2\mu\alpha^2)} W_0\left(\frac{(\beta - 2\mu\alpha^2)}{\gamma}\left(\frac{e^{-\alpha^2\mu^2}}{\omega\sigma C_f\sqrt{2\pi}}\right)^{\frac{1}{\gamma}}\right) - \frac{\mu}{\sigma}, \\ t_2 &= \frac{\gamma}{\sigma(\beta - 2\mu\alpha^2)} W_{-1}\left(\frac{(\beta - 2\mu\alpha^2)}{\gamma}\left(\frac{e^{-\alpha^2\mu^2}}{\omega\sigma C_f\sqrt{2\pi}}\right)^{\frac{1}{\gamma}}\right) - \frac{\mu}{\sigma}, \end{aligned}$$

where W denotes the Lambert function. It can be analytically shown that $r(t_{max}) \geq 1$, where $t_{max} = -\frac{\gamma}{\sigma(\beta-2\mu\alpha^2)} - \frac{\mu}{\sigma}$ is the global maximum of the ratio. Therefore, a standard rejection sampling cannot be applied. Noting that in the intervals $\left(-\frac{\mu}{\sigma}, t_1\right)$ and (t_2, ∞) it yields $h(t) > g(t)$, the rejection sampling algorithm can be modified as follows:

- Step 1: generate a sample $t_\star$ from $h(t)$ and immediately accept $x_\star = \sigma t_\star + \mu$ if $t_1 \leq t_\star \leq t_2$;
- Step 2: if $t_\star < t_1$ or $t_\star > t_2$, generate a sample u from a $\mathcal{U}(0, 1)$ density and compute $r(t_\star)$. Accept $x_\star = \sigma t_\star + \mu$ if $u \leq r(t_\star)$. The computation of $r(t_\star)$ can often be avoided if an accurate lower bound for the tails of the ratio is available;
- Step 3: if Step 2 leads to rejection, take a new sample $t'_\star$ from the distribution $d(t) = g(t) - h(t)$, in the interval $[t_1, t_2]$ and return $x'_\star = \sigma t'_\star + \mu$. Sampling

from $d(t)$ can be achieved by means of a standard rejection sampling, as in Ahrens and Dieter (1982), Stadlober (1982). More details about this step can be found in Appendix 3.A.2.

The acceptance probability of each step is discussed in Appendix 3.A.1.

Proposition 3.3.1. *The modified rejection sampling defined by steps 1, 2, and 3 draws a sample from a $\mathcal{G}_{3p}$ distribution with probability 1.*

Proof See Appendix 3.A.3.

The main computational bottleneck of the method is the evaluation of the Parabolic Cylinder function D. This issue can be alleviated by exploiting the following proposition and by the application of sharp approximations.

Proposition 3.3.2. *The Kullback-Leibler divergence (KL) between a distribution $q_x \sim \mathcal{G}_{3p}(\gamma, \alpha, \beta)$ and a distribution $p_x \sim \mathcal{G}(d, c)$, where $d = \gamma + 1$ and $c = -\beta$, goes to zero when $\beta/\alpha \to -\infty$.*

Proof See Appendix 3.B.

Furthermore, when $\beta/\alpha \to \infty$ or $\gamma \to \infty$, the three-parameter Gamma distribution can be conveniently approximated by a Normal distribution. We empirically show that, in these cases, the KL divergence between a distribution $q_x \sim \mathcal{G}_{3p}(\gamma, \alpha, \beta)$ and a distribution $p_x \sim \mathcal{N}(m, s^2)$ asymptotically goes to 0, where estimates of m and s^2 are given in Appendix 3.B. These empirical results, along with proposition 3.3.2, can be used to efficiently evaluate the mean and variance of the target distribution without the need to compute the function D for some combinations of the parameters' value.

3.4 Posterior sampling

We develop an efficient MCMC algorithm to sample from the posterior distribution of the parameters. The algorithm can be divided into three main steps: 1. a Gibbs step for the update of parameters $\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_K$; 2. a Gibbs step for the update of shrinkage parameters $\boldsymbol{\Lambda}_1^2, \ldots, \boldsymbol{\Lambda}_K^2$ and $\boldsymbol{\tau}^2$; 3. a Metropolis-Hastings (MH) step for the update of correlation matrix $\mathbf{R}$. In step 2 we make use of the modified rejection sampler introduced in Section 3.3. The complete algorithm is shown in Appendix 3.C.

1. Sampling $\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_K$**.** The full conditional distribution of $\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_K$ is

$$\pi\left(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_K | \cdot\right) \propto \prod_{k=1}^{K} \left|\boldsymbol{\Omega}_k\right|^{\frac{n_k}{2}} \exp\left\{-\frac{1}{2} tr\left(\mathbf{S}_k \boldsymbol{\Omega}_k\right)\right\} \cdot \prod_{i<j} \exp\left\{-\frac{1}{2} \boldsymbol{\omega}_{ij}^{\intercal} \boldsymbol{\Delta}_{ij}^{-1} \mathbf{R}^{-1} \boldsymbol{\Delta}_{ij}^{-1} \boldsymbol{\omega}_{ij}\right\} \cdot \mathbb{I}_{\left(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_K \in \mathbb{M}_+^K\right)}$$

where $\mathbf{S}_k = \sum_{s=1}^{n_k} \mathbf{y}_{sk} \mathbf{y}_{sk}^{\intercal}$ and $tr\left(\cdot\right)$ denotes the trace. Precision matrices $\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_K$ can be updated by adapting the block Gibbs sampler proposed in Wang (2015) for the estimation of a single precision matrix. Following Peterson et al. (2020), for each sample group $k = 1, \ldots, K$ precision matrix $\boldsymbol{\Omega}_k$ is updated column-wise by sampling from the full-conditional distribution of each column $j = 1, \ldots, p$ conditionally on both the rest of the columns of group k and on the j-th column of the reaming $k-1$ sample groups. Consider the following partition of vector $\boldsymbol{\omega}_{ij}$ and matrices $\boldsymbol{\Delta}_{ij}$ and $\mathbf{R}$:

$$\boldsymbol{\omega}_{ij} = \begin{bmatrix} \boldsymbol{\omega}_{ij}^{-k} \\ \omega_{ij}^{k} \end{bmatrix}, \quad \boldsymbol{\Delta}_{ij} = \begin{bmatrix} \boldsymbol{\Delta}_{ij,-k} & \mathbf{0} \\ \mathbf{0}^{\intercal} & \delta_{ij,k} \end{bmatrix} \quad \text{and} \quad \mathbf{R} = \begin{bmatrix} \mathbf{R}_{-k} & \mathbf{r}_k \\ \mathbf{r}_k^{\intercal} & 1 \end{bmatrix}. \tag{3.5}$$

The full conditional of $\boldsymbol{\Omega}_k$ is:

$$\pi\left(\boldsymbol{\Omega}_k | \cdot\right) \propto \left|\boldsymbol{\Omega}_k\right|^{\frac{n_k}{2}} \exp\left\{-\frac{1}{2} tr\left(\mathbf{S}_k \boldsymbol{\Omega}_k\right)\right\} \prod_{i<j} \exp\left\{-\frac{1}{2 d_{ij}^k} \left(\omega_{ij}^k - \delta_{ij,k} \mathbf{r}_k^{\intercal} \mathbf{R}_{-k}^{-1} \boldsymbol{\Delta}_{ij,-k}^{-1} \boldsymbol{\omega}_{ij}^{-k}\right)^2\right\}, \tag{3.6}$$

where $d_{ij}^k = \delta_{ij,k}^2 \left(1 - \mathbf{r}_k^{\intercal} \mathbf{R}_{-k}^{-1} \mathbf{r}_k\right)$. As proposed in Wang (2015), sampling from (3.6) can be achieved by updating one column of $\boldsymbol{\Omega}_k$ at the time. Without loss of generality, consider the permutation of the columns such that the j-th column becomes the last one. This permutation leads to the following partition:

$$\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_{-j}^k & \mathbf{s}_j^k \\ \left(\mathbf{s}_j^k\right)^{\intercal} & s_{jj}^k \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Omega}_k = \begin{bmatrix} \boldsymbol{\Omega}_{-j}^k & \boldsymbol{\omega}_j^k \\ \left(\boldsymbol{\omega}_j^k\right)^{\intercal} & \omega_{jj}^k \end{bmatrix}.$$

The full-conditional distribution of parameters $\left(\omega_{jj}^k, \boldsymbol{\omega}_j^k\right)$ is

$$\pi\left(\omega_{jj}^k, \boldsymbol{\omega}_j^k | \cdot\right) \propto \left(\omega_{jj}^k - \left(\boldsymbol{\omega}_j^k\right)^{\intercal} \left(\boldsymbol{\Omega}_{-j}^k\right)^{-1} \boldsymbol{\omega}_j^k\right)^{\frac{n_K}{2}} \cdot e^{-\frac{1}{2}\left(\left(\boldsymbol{\omega}_j^k - \mathbf{m}_{j,k}\right)^{\intercal} \mathbf{D}_{j,k}^{-1} \left(\boldsymbol{\omega}_j^k - \mathbf{m}_{j,k}\right) + 2\left(\boldsymbol{\omega}_j^k\right)^{\intercal} \mathbf{s}_j^k + s_{jj}^k \omega_{jj}^k\right)}, \tag{3.7}$$

where $\mathbf{m}_{j,k}$ is the $(p-1)$-dimensional vector with entries $m_{j,k}^i = \delta_{ij,k} \mathbf{r}_k^{\intercal} \mathbf{R}_{-k}^{-1} \boldsymbol{\Delta}_{ij,-k}^{-1} \boldsymbol{\omega}_{ij}^{-k}$ and $\mathbf{D}_{j,k}$ is diagonal with entries d_{ij}^k, $i = 0, \ldots, p, i \neq j$. A closed form for sampling

from (3.7) can be obtained with the transformation $\left(\mathbf{v}_{j,k}, \gamma_{jj}^k\right) \to \left(\boldsymbol{\omega}_j^k, \omega_{jj}^k - \left(\boldsymbol{\omega}_j^k\right)^\intercal \left(\boldsymbol{\Omega}_{-j}^k\right)^{-1} \boldsymbol{\omega}_j^k\right)$, which yields

$$\gamma_{jj}^k|\cdot \sim \mathcal{G}\left(\frac{n_k}{2}+1, \frac{s_{jj}^k}{2}\right),$$
$$\mathbf{v}_{j,k}|\cdot \sim \mathcal{N}_{p-1}\left(\boldsymbol{\Sigma}_{j,k}^{-1}\left(\mathbf{D}_{j,k}^{-1}\mathbf{m}_{j,k} - \mathbf{s}_{jj}^k\right), \boldsymbol{\Sigma}_{j,k}^{-1}\right)$$

where $\mathcal{G}$ denotes the Gamma distribution and $\boldsymbol{\Sigma}_{j,k} = \mathbf{D}_{j,k}^{-1} + s_{jj}^k \left(\boldsymbol{\Omega}_{-j}^k\right)^{-1}$. Therefore, values ω_{jj}^k and $\boldsymbol{\omega}_j^k$ can be updated by first sampling γ_{jj}^k and $\mathbf{v}_{j,k}$ and then applying the inverse transformation.

Computationally, this is the most expensive step of the algorithm due to the need to invert the matrices $\boldsymbol{\Omega}_{-j}^k$ and $\boldsymbol{\Sigma}_{j,k}$. In our implementation of the Gibbs steps for γ_{jj}^k and $\mathbf{v}_{j,k}$, we make use of Shermann-Morrison formula to update $\left(\boldsymbol{\Omega}_{-j}^k\right)^{-1}$ with $O(p^2)$ operations, instead of $O(p^3)$.

2. Sampling $\boldsymbol{\Lambda}_1^2, \ldots, \boldsymbol{\Lambda}_K^2$ and $\boldsymbol{\tau}^2$. Samplers commonly used in conjunction with Horseshoe prior cannot be implemented for the proposed model. Indeed, the positive half-Cauchy distribution is not conjugated to the variance in a multivariate normal means model. Our approach builds upon the data-augmentation scheme proposed Makalic and Schmidt (2016). We introduce the auxiliary variables $\eta_{ij,k}$ and ζ_k such that

- if $\lambda_{ij,k}^2 \mid \eta_{ij,k} \sim \mathcal{IG}\left(\frac{1}{2}, \frac{1}{\eta_{ij,k}}\right)$ and $\eta_{ij,k} \sim \mathcal{IG}\left(\frac{1}{2}, 1\right)$, then $\lambda_{ij,k} \sim \mathcal{C}^+(0,1)$;
- if $\tau_k^2 \mid \zeta_k \sim \mathcal{IG}\left(\frac{1}{2}, \frac{1}{\zeta_k}\right)$ and $\zeta_k \sim \mathcal{IG}\left(\frac{1}{2}, 1\right)$, then $\tau_k \sim \mathcal{C}^+(0,1)$.

After conditioning on the auxiliary variables $\eta_{ij,k}$ and ζ_k, the full conditional distribution of parameters $\boldsymbol{\Lambda}$ and $\boldsymbol{\tau}$ can be written as

$$\pi\left(\boldsymbol{\Lambda}, \boldsymbol{\tau}|\cdot\right) \propto \prod_{i<j} |\boldsymbol{\Delta}_{ij}|^{-1} \exp\left\{-\frac{1}{2}\boldsymbol{\omega}_{ij}^\intercal \left(\boldsymbol{\Delta}_{ij}\mathbf{R}\boldsymbol{\Delta}_{ij}\right)^{-1}\boldsymbol{\omega}_{ij}\right\} \cdot$$
$$\prod_{k=1}^K \tau_k^{-3} \exp\left\{-\frac{1}{\zeta_k \tau_k^2}\right\} \cdot \prod_{i<j} \lambda_{ij,k}^{-3} \exp\left\{-\frac{1}{\eta_{ij,k}\lambda_{ij,k}^2}\right\}.$$

Local shrinkage matrix $\boldsymbol{\Lambda}_k$ is updated column-wise alongside precision matrix $\boldsymbol{\Omega}_k$. Considering the partition of $\boldsymbol{\omega}_{ij}$, $\boldsymbol{\Delta}_{ij}$ and $\mathbf{R}$ in (3.5), the full-conditionals of pa-

rameters $\lambda^2_{ij,k}$ and τ^2_k related to group k are

$$\pi\left(\lambda^2_{ij,k} \mid \cdot\right) \propto \lambda^{-4}_{ij,k} \exp\left\{-\frac{\alpha_{\lambda_{ij,k}}}{\lambda^2_{ij,k}} + \frac{\beta_{\lambda_{ij,k}}}{\lambda_{ij,k}}\right\} \cdot \mathbb{I}_{\left(\lambda^2_{ij,k}>0\right)},$$

$$\alpha_{\lambda_{ij,k}} = \frac{1}{\eta_{ij,k}} + \frac{\left(\omega^k_{ij}\right)^2}{2\tau^2_k \mu_k} \quad \text{and} \quad \beta_{\lambda_{ij,k}} = \frac{\omega^k_{ij}}{\tau_k \mu_k} \mathbf{r}^\intercal_k \mathbf{R}^{-1}_{-k} \mathbf{\Delta}^{-1}_{ij,-k} \boldsymbol{\omega}^{-k}_{ij}, \quad (3.8)$$

$$\pi\left(\tau^2_k \mid \cdot\right) \propto \tau_k^{-\frac{p(p-1)}{2}-3} \exp\left\{-\frac{\alpha_{\tau_k}}{\tau^2_k} + \frac{\beta_{\tau_k}}{\tau_k}\right\} \cdot \mathbb{I}_{\left(\tau^2_k>0\right)},$$

$$\alpha_{\tau_k} = \frac{1}{\zeta_k} + \frac{1}{2}\sum_{i<j} \frac{\left(\omega^k_{ij}\right)^2}{\lambda^2_{ij,k}\mu_k} \quad \text{and} \quad \beta_{\tau_k} = \sum_{i<j} \frac{\omega^k_{ij}}{\lambda_{ij,k}\mu_k} \mathbf{r}^\intercal_k \mathbf{R}^{-1}_{-k} \mathbf{\Delta}^{-1}_{ij,-k} \boldsymbol{\omega}^{-k}_{ij} \quad (3.9)$$

where $\mu_k = 1 - \mathbf{r}^\intercal_k \mathbf{R}^{-1}_{-k} \mathbf{r}_k$. Note that the full conditional distributions show a shared amount of global and local shrinkage, as the model exploits the similarity among groups and learns from the structures of the other graphs. Densities (3.8) and (3.9) are a transformation of $\mathcal{G}_{3p}$ random variables introduced in Section 3.3. Specifically,

$$\begin{aligned} &\text{if} \quad u \sim \mathcal{G}_{3p}\left(1, \alpha_{\lambda_{ij,k}}, \beta_{\lambda_{ij,k}}\right), \quad \text{then} \quad \lambda^2_{ij,k} = 1/u^2, \\ &\text{if} \quad u \sim \mathcal{G}_{3p}\left(p(p-1)/2, \alpha_{\tau_k}, \beta_{\tau_k}\right), \quad \text{then} \quad \tau^2_k = 1/u^2. \end{aligned}$$

We use the sampling algorithm introduced in Section 3.3 to efficiently obtain samples from these distributions. Finally, hyper-parameters $\eta_{ij,k}$ and ζ_k are updated by sampling from the inverse-Gamma distributions $\eta_{ij,k} \sim \mathcal{IG}\left(1, 1 + 1/\lambda^2_{ij,k}\right)$ and $\zeta_k \sim \mathcal{IG}\left(1, 1 + 1/\tau^2_k\right)$.

***3. Sampling* R.** The similarity among groups is captured through correlation matrix $\mathbf{R} \in \mathbb{C}^K_+$. Following Peterson et al. (2020), we implement a modified version of the Metropolis-Hastings sampler proposed by Liu and Daniels (2006), which relies on a candidate prior distribution $\pi^\star(\mathbf{R})$ that is used to define a proposal distribution for correlation matrices. In the first step of this data-augmentation approach a $K \times K$ covariance matrix $\mathbf{\Theta}$ is sampled from an Inverse-Wishart distribution; in the second step, a reduction function is applied to map the covariance matrix to a valid correlation matrix, that is eventually accepted with an MH step.

We introduce a diagonal matrix $\mathbf{V}$ such that $\mathbf{\Theta} = \mathbf{VRV}$; the matrix $\mathbf{V}$ maps the correlation matrix $\mathbf{R}$ to the covariance matrix $\mathbf{\Theta}$. Following Peterson et al. (2020), the transformation from the standard parameter space to the expanded space is achieved as

$$\boldsymbol{\omega}_{ij} = \mathbf{V}^{-1}\boldsymbol{\epsilon}_{ij}, \quad \mathbf{R} = \mathbf{V}^{-1}\mathbf{\Theta}\mathbf{V}^{-1}, \quad (3.10)$$

where $\sum_{i<j} \epsilon_{ijk}^2 = 1$, for $k = 1, \ldots, K$ and $\mathbf{V} = \text{diag}\left\{\sum_{i<j} \left(\omega_{ij}^1\right)^2, \ldots, \sum_{i<j} \left(\omega_{ij}^K\right)^2\right\}$. Let the candidate prior distribution be

$$\pi^\star(\mathbf{R}) \propto \left|\mathbf{R}\right|^{-\frac{K+1}{2}} \cdot \mathbb{I}_{\left(\mathbf{R} \in \mathbb{C}_+^K\right)}, \tag{3.11}$$

then the proposal density for matrix $\mathbf{R}$ is

$$\begin{aligned} q(\mathbf{R} \mid \cdot) &\propto \pi^\star(\mathbf{R}) \cdot \pi(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_K \mid \mathbf{R}) \\ &\propto |\mathbf{R}|^{-\frac{K+1+p(p-1)/2}{2}} \cdot \prod_{i<j} e^{-\frac{1}{2}\boldsymbol{\omega}_{ij}^\intercal \boldsymbol{\Delta}_{ij}^{-1} \mathbf{R}^{-1} \boldsymbol{\Delta}_{ij}^{-1} \boldsymbol{\omega}_{ij}}, \end{aligned}$$

which is conditioned on the current state of the algorithm and accounts for the dependency with parameters $\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_K$, $\boldsymbol{\Lambda}_1, \ldots, \boldsymbol{\Lambda}_K$ and $\boldsymbol{\tau}^2$. Note that (3.11) concentrates its mass around zero when K increases; for this reason, a reasonably small number of sample groups K is required. The Jacobian of the transformation defined in (3.10) is $J = |\mathbf{V}^{-1}|^{\frac{p(p-1)}{2}+K+1}$, thus the proposal distribution for the MH sampler is

$$\begin{aligned} q(\boldsymbol{\Theta} \mid \cdot) &\propto \pi^\star(\boldsymbol{\Theta}) \cdot \pi(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_K \mid \boldsymbol{\Theta}) \\ &\propto |\boldsymbol{\Theta}|^{-\frac{K+1+p(p-1)/2}{2}} \, e^{-\frac{1}{2}\sum_{i<j} \boldsymbol{\epsilon}_{ij}^\intercal \boldsymbol{\Delta}_{ij}^{-1} \boldsymbol{\Theta}^{-1} \boldsymbol{\Delta}_{ij}^{-1} \boldsymbol{\epsilon}_{ij}} \end{aligned} \tag{3.12}$$

which is a $\mathcal{IW}\left(\frac{p(p-1)}{2}, \mathbf{H}\right)$, where $\mathbf{H} = \sum_{i<j} \boldsymbol{\Delta}_{ij}^{-1} \boldsymbol{\epsilon}_{ij} \boldsymbol{\epsilon}_{ij}^\intercal \boldsymbol{\Delta}_{ij}^{-1}$. Therefore, a candidate $\boldsymbol{\Theta}^\star$ is sampled from (3.12) and then mapped to $\mathbf{R}^\star$ via the inverse transformation $\mathbf{R}^\star = \mathbf{V}^{-1}\boldsymbol{\Theta}^\star\mathbf{V}^{-1}$. New correlation matrix $\mathbf{R}^\star$ is accepted with probability

$$\begin{aligned} \alpha &= \min\left\{1, \frac{\pi(\mathbf{R}^\star \mid \cdot) \cdot q(\mathbf{R} \mid \cdot)}{\pi(\mathbf{R} \mid \cdot) \cdot q(\mathbf{R}^\star \mid \cdot)}\right\} \\ &= \min\left\{1, e^{\frac{K+1}{2}(\log|\mathbf{R}^\star| - \log|\mathbf{R}|)}\right\}, \end{aligned}$$

where $p(\mathbf{R} \mid \cdot) \propto \pi(\mathbf{R}) \cdot \pi(\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_K \mid \mathbf{R})$ denotes the full-conditional distribution of $\mathbf{R}$.

3.5 Posterior edge selection

A practical problem with continuous shrinkage priors is model selection since the parameters are shrunk toward zero but never exactly zero. A common method relies on posterior marginal credible intervals. However, Van der Pas et al. (2017) have shown that under the Horseshoe prior in a Normal means problem, this method leads to a conservative variables selection procedure where some of the zero

parameters are falsely selected, whereas some signal is not, due to wide intervals for non-zero parameters. To avoid such a problem, Li et al. (2019) used 50% credible intervals to control the number of false negatives. This choice is in line with the *median probability model* (MPM) of Barbieri and Berger (2004). The MPM model is defined as the model that includes only those edges with marginal posterior probability greater (or equal) than 1/2. In the context of linear regression models, Barbieri and Berger (2004) have shown that this method represents the predictive optimal model under some common but strict hypothesis, such as orthogonality of the covariates. The result is extended to g-type spike and slab priors in Barbieri et al. (2021). This approach is used, among many others, in Wang (2015) and Peterson et al. (2020). A practical example of an MPM-like strategy can be found in Carvalho et al. (2010). The authors show that the Horseshoe estimator is $\beta_j^{\text{HS}} = \lambda_j^2/\left(1+\lambda_j^2\right)\beta_j^{\text{OLS}}$, where λ_j^2 and β_j denote the local shrinkage parameter and the regression parameter of variable j, respectively, and propose to set to zero those variables for which $\lambda_j^2/\left(1+\lambda_j^2\right) < 1/2$.

The cited methods present two main drawbacks. First, the optimality results in Barbieri and Berger (2004) only hold for fixed design $\tilde{\mathbf{X}}$ of prediction point or for stochastic predictors with $\mathbb{E}\left(\tilde{\mathbf{X}}^\intercal\tilde{\mathbf{X}}\right)$, which are often unrealistic assumptions; therefore, the threshold 1/2 does not ensure the optimality of the selected model under the considered framework, where the goal is to analyze the connections between variables. Secondly, the considered selection procedures rely on marginal values and do not account for any posterior correlation among the parameters.

To overcome these problems, we propose a "quasi-bayesian" approach for edge selection that accounts for the posterior dependencies among the parameters. The method relies on a *cut* function that "cuts" the relationship between the parameters to prevent model feedback which could negatively affect the performances of the model (Zigler et al., 2013; Plummer, 2015). Cuts have been used in different contexts (Lunn et al., 2009; Bayarri et al., 2009; McCandless et al., 2010; Blangiardo et al., 2011; Zigler, 2016) either to control the flow of information or to gain a computational advantage. Bayarri et al. (2009) consider the cut function as a "*modularization*" of the model. This approach breaks a bigger model into smaller parts called modules, modifying the magnitude of the interactions between the parameters in different modules.

3.5.1 An extended model and algorithm for edge selection

In this section, we extend the model presented in the previous sections introducing two parameters t_α and $\mathbf{z}$, and an algorithm that updates these parameters with a Metropolis-within-Gibbs step. Notation refers to a single graph and can be easily extended to the case of multiple graphs.

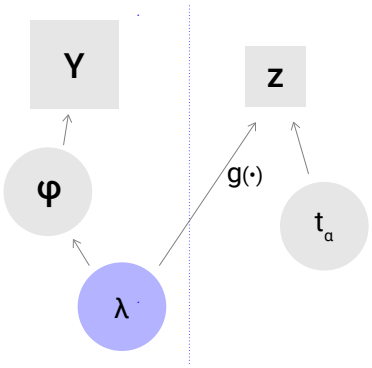


Figure 3.5.1: Graphical representation of the model. The dotted line denotes the cut function, stopping the flows of information from $\mathbf{z}$ to $\boldsymbol{\lambda}$.

The parameter $t_\alpha \in (0,1)$ can be interpreted as a threshold for edge selection, and the latent variable $\mathbf{z}$ is a $p(p-1)/2$-binary vector with generic element $z_{ij} = 1$ if the corresponding edge ω_{ij}, $i < j$, is included in the model, $z_{ij} = 0$ otherwise. Formally, the model is defined as

$$z_{ij} = 1 \text{ if } \kappa_{ij} \geq t_\alpha, \text{ and } \quad z_{ij} = 0 \text{ otherwise,}$$

where $\kappa_{ij} = \lambda_{ij}^2 / \left(1 + \lambda_{ij}^2\right)$. Here the goal is to estimate parameter t_α based on the posterior values of $\boldsymbol{\lambda}$. At the same time, we want to prevent the flow of information from t_α and $\mathbf{z}$ to $\boldsymbol{\lambda}$. The cut function comes in handy to avoid such issues. The modularization of the proposed model is shown in Figure 3.5.1, where $\boldsymbol{\varphi} = (\boldsymbol{\Omega}, \boldsymbol{\tau}, \mathbf{R})$ and parameters $\mathbf{z}$ and $\boldsymbol{\lambda}$ are connected through the reparametrization κ_{ij}.

Different prior distributions can be assumed for t_α; a natural choice is $t^\alpha \sim \mathsf{Beta}(a, b)$. Parameters z_{ij} can be seen as the realization of $p(p-1)/2$ Bernoulli distributions $z_{ij} \mid \kappa_{ij}, \boldsymbol{\varphi}, t_\alpha \sim \mathsf{Ber}\left(q_{ij}^\alpha\right)$, where $q_{ij}^\alpha = 1 - \mathbb{P}\left(\kappa_{ij} \leq t^\alpha \mid \boldsymbol{\varphi}\right)$. The joint likelihood of the model can be factorized as

$$\begin{aligned} \pi\left(\mathbf{Y}, \boldsymbol{\lambda}, \boldsymbol{\varphi}, \mathbf{z}, t_\alpha\right) &\propto \pi\left(\boldsymbol{\varphi} \mid \mathbf{Y}, \boldsymbol{\lambda}\right) \pi\left(\mathbf{z}, t_\alpha \mid \boldsymbol{\kappa}, \boldsymbol{\varphi}\right) \pi\left(\boldsymbol{\lambda}\right), \\ &\propto \pi\left(\boldsymbol{\lambda}, \boldsymbol{\varphi} \mid \mathbf{Y}\right) \pi\left(\mathbf{z}, t_\alpha \mid \boldsymbol{\kappa}, \boldsymbol{\varphi}\right). \end{aligned}$$

The modularization of the model allows us to sample directly from the conditional distributions $\pi\left(\boldsymbol{\lambda}, \boldsymbol{\varphi} \mid \mathbf{Y}\right)$ and $\pi\left(\mathbf{z}, t_\alpha \mid \boldsymbol{\kappa}, \boldsymbol{\varphi}\right)$, thus evaluating parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\varphi}$ without the influence of the unknown quantity $\mathbf{z}$. The joint posterior distribution of parameters t_α and $\mathbf{z}$ is

$$\pi\left(\mathbf{z}, t^\alpha \mid \boldsymbol{\kappa}, \boldsymbol{\varphi}\right) \propto \left(t^\alpha\right)^{a-1} \left(1 - t^\alpha\right)^{b-1} \cdot \prod_{j=1}^{p} \prod_{i<j} \left(q_{ij}^\alpha\right)^{z_{ij}} \left(1 - q_{ij}^\alpha\right)^{1-z_{ij}}. \tag{3.13}$$

We propose a Metropolis-within-Gibbs algorithm in order to sample from (3.13). Parameters z_{ij} are sampled from the full-conditional distribution

$$z_{ij} \mid \kappa_{ij}, \boldsymbol{\varphi}, t^\alpha \sim \mathsf{Ber}\left(q_{ij}^\alpha\right).$$

Under the framework introduced in Section 3.4, the transformation $\kappa_{ij} = \lambda_{ij}^2/\left(1+\lambda_{ij}^2\right)$ with Jacobian $J_{\kappa_{ij}} = (1+\kappa_{ij})^{-2}$ yields

$$\pi\left(\kappa_{ij} \mid \mathbf{Y}, \boldsymbol{\varphi}\right) \propto \kappa_{ij}^{-2} \exp\left\{-\alpha_{\lambda_{ij}} \frac{1-\kappa_{ij}}{\kappa_{ij}} + \beta_{\lambda_{ij}} \sqrt{\frac{1-\kappa_{ij}}{\kappa_{ij}}}\right\} \cdot \mathbb{I}_{(\kappa_{ij} \in (0,1))},$$

where the cumulative density function $F_{\kappa_{ij}|\mathbf{Y},\boldsymbol{\varphi}}\left(t^\alpha\right)$ is available in closed form. Therefore, the quantity q_{ij}^α can be analytically and efficiently computed conditionally on the current state of $\boldsymbol{\varphi}$.

The threshold t^α is then updated with a MH step, where the new values $t_\star^\alpha$ are sampled from the prior distribution. The acceptance probability of this step is

$$\alpha_{\text{MH}} = \min\left\{1, \frac{\pi\left(\mathbf{z}, t_\star^\alpha \mid \boldsymbol{\kappa}, \boldsymbol{\varphi}\right)}{\pi\left(\mathbf{z}, t^\alpha \mid \boldsymbol{\kappa}, \boldsymbol{\varphi}\right)}\right\}.$$

The sampled values of t^α can be used to perform graph selection; specifically, we include in the graph all edges such that $P(\kappa_{ij} \mid \mathbf{Y}, \boldsymbol{\varphi}) > t^\alpha$. Hereafter, we consider both this approach and the MPM method ($t^\alpha = 1/2$; Carvalho et al., 2010) as two alternative approaches to posterior edge selection.

3.6 Simulation studies

We perform simulation studies that cover several scenarios of interest. The performances of the proposed model and competing approaches are tested in four scenarios all comprising $K = 4$ groups:

- **Independence set-up**: the groups are simulated from multivariate Gaussian distributions with a different precision matrix for each group;
- **Coupled set-up**: each pair of groups is simulated from a multivariate Gaussian distribution with the same precision matrix;
- **P2020 set-up**: the groups are simulated following the scheme of Peterson et al. (2020), where each precision matrix is created by adding (deleting) new edges to (from) the other precision matrices;
- **Full-dependence set-up**: the groups are simulated from multivariate Gaussian distributions with equal precision matrices.

The precision matrices are simulated following the approach of Peterson et al. (2020), which relies on a generalization of the method proposed by Danaher et al. (2014). Edges are divided into independent subgroups with size either equal to 5 or 10. Diagonal entries of the precision matrices are set to 1. We test our model against the fused and grouped Graphical LASSO (fJGL and gJGL, respectively) of Danaher et al. (2014), the ordinary Graphical Horseshoe (GHS) of Li et al. (2019) estimated for each group independently, and the group estimation of multiple Bayesian graphical models (GemBAG) from Yang et al. (2021). Among all competing approaches, the proposed approach is the only one that provides uncertainty quantification through posterior inference on all model parameters.

Different combinations of n and p are evaluated, and the results are reported in Tables 3.6.1-3.6.4, where p_0 refers to the mean number of true significant edges across groups. Edge selection is assessed based on accuracy, the Matthews correlation coefficient (MCC), true and false positive rate (TPR and FPR, respectively) and the AUC criterion. We take the mean Frobenius loss among groups matrices to evaluate the goodness of the precision matrices estimates. Subscripts MPM and t_α indicate whether the posterior edge selection is performed based on the MPM method or with the cut-model proposed in Section 3.5, respectively. Hyperparameters a and b of the Beta prior on t_α should reflect prior beliefs in graphs' sparsity; to control the number of false positives, we set $a = 30$ and $b = 25$. For the fused and grouped Graphical LASSO, regulation parameters λ_1 and λ_2 are selected by performing a grid search to find the combination of values that minimizes the AIC (Danaher et al., 2014; Peterson et al., 2020). For GemBAG, hyperparameters related to the two levels of sparseness are set to $p_1 = 0.4$ and $p_2 = 0.8$ for all the considered cases. Prior variances v_0 and v_1 are estimated by minimizing the BIC criterion over a grid of values, as done in Yang et al. (2021).

In all scenarios, see tables 3.6.1-3.6.4, mGHS performs better than GHS applied to each group separately when the groups are actually similar, as it provides better selection performances in all the coupled, P2020 and full-dependence settings. Moreover, our model is the only competitor able to approach the performances of the GHS in the independent set-up. Indeed, in this case the latter shows better performances than all the other competitors for all the considered values of n and p, whereas the Graphical LASSO and GemBAG behave poorly and their selection results worsen as p increases.

The P2020 set-up provides the most realistic scheme, where the groups have similar but different precision matrices. Under these circumstances, the best model is GemBAG, which gives higher values of MCC and AUC for $p \geq 100$. The only competitive model is mGHS, which has the highest AUC when $p = 50$ and it is the only competitor able to approach GemBAG's performances in the other considered cases.

In this simulation study, edge selection based on the cut model completely overtakes the selection procedure based on the MPM model. Indeed, the approach based on cuts strongly reduces the number of false discoveries, resulting in a higher value of the MCC index. Note that the value of the estimated threshold is affected by the choice of the prior distribution of t^α. We used $t^\alpha \sim \mathsf{Beta}(30, 25)$ across all simulation scenarios and data analyses; in our experience, this is a viable option that leads to control of the FPR even though different choices may lead to a different level of sparsity in the estimated graphs.

Finally, the GemBAG and fJGL provide the lowest values of the Frobenius loss. Except for the independent setting, none of the other methods gives better performances in terms of precision matrices estimation. GemBAG is the most efficient method, as it takes an average of only a few hours for the estimation of a network with $p = 500$. On the contrary, the mGHS provides a fully Bayesian inference at the cost of a 10-fold increase in computational time. GHS and Graphical LASSO have not been included in this case, as the computational time increases dramatically.

$n = 50, p = 50$	*Independent* ($p_0 = 82.5$)						*Coupled* ($p_0 = 77.5$)					
	Acc	MCC	TPR	FPR	AUC	Fr Loss	Acc	MCC	TPR	FPR	AUC	Fr Loss
mGHS$_{\text{MPM}}$	0.775	0.299	0.744	0.223	0.824	10.231	0.715	0.230	**0.723**	0.286	**0.789**	8.624
	(0.018)	(0.030)	(0.040)	(0.019)	(0.027)	(1.224)	(0.039)	(0.039)	(0.048)	(0.041)	(0.037)	(1.029)
mGHS$_{t_\alpha}$	0.926	**0.459**	0.544	0.046	0.824	10.231	0.930	**0.392**	0.421	0.035	**0.789**	8.624
	(0.008)	(0.038)	(0.052)	(0.009)	(0.027)	(1.224)	(0.009)	(0.055)	(0.086)	(0.012)	(0.037)	(1.209)
GHS$_{\text{MPM}}$	0.786	0.315	**0.754**	0.211	**0.840**	10.199	0.702	0.204	0.684	0.297	0.760	8.745
	(0.015)	(0.029)	(0.037)	(0.015)	(0.024)	(1.246)	(0.047)	(0.044)	(0.048)	(0.049)	(0.040)	(0.940)
fJGL	0.873	0.384	0.648	0.110	0.769	**9.186**	0.907	0.333	0.437	0.061	0.688	**7.863**
	(0.024)	(0.037)	(0.063)	(0.029)	(0.024)	(0.709)	(0.021)	(0.044)	(0.088)	(0.026)	(0.034)	(0.535)
gJGL	0.874	0.383	0.645	0.109	0.768	9.232	0.906	0.328	0.436	0.062	0.687	7.998
	(0.024)	(0.036)	(0.062)	(0.028)	(0.024)	(0.720)	(0.021)	(0.043)	(0.091)	(0.027)	(0.036)	(0.557)
GemBAG$_{\text{MPM}}$	**0.940**	0.311	0.124	**0.001**	0.791	11.835	**0.940**	0.238	0.081	**0.001**	0.786	8.580
	(0.002)	(0.052)	(0.041)	(0.002)	(0.057)	(1.150)	(0.002)	(0.064)	(0.036)	(0.002)	(0.050)	(0.775)
	*P*2020 ($p_0 = 82.5$)						*Full dependence* ($p_0 = 85$)					
	Acc	MCC	TPR	FPR	AUC	Fr Loss	Acc	MCC	TPR	FPR	AUC	Fr Loss
mGHS$_{\text{MPM}}$	0.796	0.358	**0.822**	0.206	**0.875**	8.498	0.716	0.247	**0.735**	0.285	0.792	8.349
	(0.011)	(0.021)	(0.030)	(0.011)	(0.020)	(1.323)	(0.034)	(0.036)	(0.046)	(0.036)	(0.032)	(1.184)
mGHS$_{t_\alpha}$	0.925	0.532	0.698	0.059	**0.875**	8.498	0.923	**0.408**	0.446	0.041	0.792	8.349
	(0.008)	(0.034)	(0.037)	(0.009)	(0.020)	(1.323)	(0.009)	(0.046)	(0.074)	(0.012)	(0.032)	(1.184
GHS$_{\text{MPM}}$	0.795	0.321	0.748	0.202	0.840	9.371	0.670	0.165	0.631	0.327	0.710	8.616
	(0.013)	(0.027)	(0.037)	(0.013)	(0.022)	(1.216)	(0.055)	(0.046)	(0.046)	(0.059)	(0.043)	(0.954)
fJGL	0.874	0.412	0.697	0.113	0.792	8.205	0.905	0.309	0.373	0.055	0.659	**7.711**
	(0.023)	(0.046)	(0.050)	(0.025)	(0.025)	(0.702)	(0.021)	(0.056)	(0.100)	(0.028)	(0.040)	(0.611)
gJGL	0.864	0.376	0.660	0.121	0.770	8.851	0.902	0.293	0.358	0.057	0.650	7.989
	(0.025)	(0.036)	(0.054)	(0.029)	(0.022)	(0.714)	(0.023)	(0.048)	(0.100)	(0.030)	(0.039)	(0.579)
GemBAG$_{\text{MPM}}$	**0.956**	**0.580**	0.367	**0.001**	0.871	**7.835**	**0.938**	0.318	0.112	**0.000**	**0.838**	7.984
	(0.004)	(0.049)	(0.065)	(0.001)	(0.035)	(1.043)	(0.002)	(0.049)	(0.032)	(0.000)	(0.031)	(0.651)

Table 3.6.1: Simulation results for $n = 50$ and $p = 50$ (50 replicates). Methods mGHS and GHS are evaluated over $B = 10000$ post burn-in samples.

$n = 50, p = 100$	*Independent* ($p_0 = 195$)						*Coupled* ($p_0 = 177.5$)					
	Acc	MCC	TPR	FPR	AUC	Fr Loss	Acc	MCC	TPR	FPR	AUC	Fr Loss
mGHS$_{\text{MPM}}$	0.655	0.146	0.712	0.348	0.759	20.607	0.568	0.082	**0.653**	0.436	0.671	17.547
	(0.024)	(0.018)	(0.030)	(0.025)	(0.023)	(1.444)	(0.028)	(0.020)	(0.036)	(0.028)	(0.037)	(1.283)
mGHS$_{t_\alpha}$	0.953	**0.348**	0.361	0.022	0.759	20.607	0.961	0.228	0.152	0.009	0.671	17.547
	(0.004)	(0.032)	(0.044)	(0.005)	(0.023)	(1.444)	(0.003)	(0.050)	(0.060)	(0.005)	(0.037)	(1.283)
GHS$_{\text{MPM}}$	0.669	0.155	**0.715**	0.333	**0.769**	20.594	0.563	0.074	0.638	0.439	0.655	17.545
	(0.023)	(0.019)	(0.030)	(0.024)	(0.024)	(1.453)	(0.029)	(0.020)	(0.036)	(0.029)	(0.036)	(1.283)
fJGL	0.931	0.315	0.451	0.049	0.701	**19.892**	0.952	0.234	0.226	0.021	0.603	**16.296**
	(0.012)	(0.028)	(0.054)	(0.014)	(0.022)	(0.988)	(0.009)	(0.033)	(0.074)	(0.012)	(0.032)	(0.694)
gJGL	0.929	0.312	0.456	0.051	0.702	19.921	0.952	**0.229**	0.219	0.021	0.599	16.689
	(0.013)	(0.029)	(0.058)	(0.015)	(0.023)	(1.055)	(0.009)	(0.034)	(0.074)	(0.011)	(0.032)	(0.722)
GemBAG$_{\text{MPM}}$	**0.962**	0.179	0.052	**0.001**	0.698	23.012	**0.965**	0.143	0.034	**0.001**	**0.708**	16.986
	(0.001)	(0.043)	(0.026)	(0.001)	(0.069)	(2.174)	(0.001)	(0.046)	(0.016)	(0.000)	(0.044)	(0.894)
	*P*2020 ($p_0 = 182.5$)						*Full dependence* ($p_0 = 185$)					
	Acc	MCC	TPR	FPR	AUC	Fr Loss	Acc	MCC	TPR	FPR	AUC	Fr Loss
mGHS$_{\text{MPM}}$	0.720	0.215	**0.808**	0.284	0.853	18.878	0.589	0.107	**0.692**	0.415	0.714	17.127
	(0.013)	(0.014)	(0.025)	(0.013)	(0.016)	(1.944)	(0.030)	(0.021)	(0.036)	(0.031)	(0.035)	(1.491)
mGHS$_{t_\alpha}$	0.948	0.459	0.625	0.040	0.853	18.878	0.958	**0.285**	0.230	0.013	0.714	17.127
	(0.004)	(0.022)	(0.030)	(0.005)	(0.016)	(1.944)	(0.004)	(0.046)	(0.071)	(0.006)	(0.035)	(1.491)
GHS$_{\text{MPM}}$	0.710	0.181	0.733	0.291	0.800	20.299	0.564	0.072	0.626	0.438	0.647	17.256
	(0.016)	(0.015)	(0.026)	(0.017)	(0.016)	(1.650)	(0.029)	(0.022)	(0.040)	(0.029)	(0.039)	(1.261)
fJGL	0.935	0.393	0.588	0.052	0.768	18.557	0.955	0.240	0.201	0.016	0.593	**16.103**
	(0.010)	(0.030)	(0.042)	(0.011)	(0.018)	(1.070)	(0.007)	(0.042)	(0.079)	(0.010)	(0.035)	(1.027)
gJGL	0.923	0.335	0.540	0.062	0.739	20.104	0.955	0.223	0.182	0.015	0.583	16.772
	(0.011)	(0.024)	(0.043)	(0.013)	(0.018)	(1.107)	(0.008)	(0.038)	(0.071)	(0.010)	(0.031)	(0.924)
GemBAG$_{\text{MPM}}$	**0.975**	**0.550**	0.321	**0.000**	**0.869**	**15.676**	**0.966**	0.277	0.084	**0.000**	**0.808**	16.411
	(0.002)	(0.041)	(0.052)	(0.000)	(0.015)	(1.264)	(0.001)	(0.038)	(0.022)	(0.000)	(0.037)	(1.044)

Table 3.6.2: Simulation results for $n = 50$ and $p = 100$ (50 replicates). Methods mGHS and GHS are evaluated over $B = 10000$ post burn-in samples.

$n = 100, p = 250$	*Independent* ($p_0 = 532.5$)						*Coupled* ($p_0 = 477.5$)					
	Acc	MCC	TPR	FPR	AUC	Fr Loss	Acc	MCC	TPR	FPR	AUC	Fr Loss
mGHS$_{\text{MPM}}$	0.632	0.115	0.808	0.371	0.830	34.766	0.556	0.077	**0.761**	0.447	0.761	31.934
	(0.012)	(0.006)	(0.015)	(0.012)	(0.012)	(1.629)	(0.010)	(0.007)	(0.021)	(0.010)	(0.018)	(1.062)
mGHS$_{t_\alpha}$	0.976	**0.420**	0.524	0.015	0.830	34.766	0.983	**0.350**	0.308	0.007	0.761	31.934
	(0.002)	(0.016)	(0.021)	(0.002)	(0.012)	(1.629)	(0.001)	(0.019)	(0.036)	(0.002)	(0.018)	(1.062)
GHS$_{\text{MPM}}$	0.639	0.118	**0.812**	0.364	**0.835**	**34.721**	0.551	0.069	0.729	0.451	0.732	32.946
	(0.007)	(0.005)	(0.015)	(0.007)	(0.011)	(1.596)	(0.010)	(0.007)	(0.023)	(0.010)	(0.019)	(1.024)
fJGL	0.956	0.345	0.617	0.038	0.790	37.527	0.971	0.307	0.423	0.021	0.701	31.616
	(0.005)	(0.016)	(0.025)	(0.006)	(0.011)	(1.235)	(0.004)	(0.020)	(0.030)	(0.004)	(0.014)	(0.839)
gJGL	0.956	0.344	0.618	0.038	0.790	37.581	0.970	0.292	0.407	0.022	0.693	32.616
	(0.005)	(0.016)	(0.024)	(0.006)	(0.010)	(1.169)	(0.005)	(0.017)	(0.040)	(0.005)	(0.018)	(0.917)
GemBAG$_{\text{MPM}}$	**0.985**	0.344	0.147	**0.000**	0.697	46.156	**0.986**	0.326	0.130	**0.000**	**0.836**	**30.824**
	(0.000)	(0.018)	(0.013)	(0.000)	(0.020)	(1.840)	(0.000)	(0.022)	(0.014)	(0.000)	(0.010)	(0.927)
	*P*2020 ($p_0 = 482.5$)						*Full dependence* ($p_0 = 485$)					
	Acc	MCC	TPR	FPR	AUC	Fr Loss	Acc	MCC	TPR	FPR	AUC	Fr Loss
mGHS$_{\text{MPM}}$	0.654	0.132	**0.863**	0.350	0.885	26.270	0.575	0.095	**0.811**	0.429	0.815	30.133
	(0.007)	(0.004)	(0.013)	(0.007)	(0.008)	(1.321)	(0.011)	(0.006)	(0.017)	(0.011)	(0.014)	(1.058)
mGHS$_{t_\alpha}$	0.972	0.460	0.699	0.024	0.885	26.270	0.981	0.406	0.440	0.011	0.815	30.133
	(0.002)	(0.013)	(0.017)	(0.002)	(0.008)	(1.321)	(0.002)	(0.017)	(0.034)	(0.002)	(0.014)	(1.058)
GHS$_{\text{MPM}}$	0.659	0.123	0.817	0.344	0.850	29.366	0.552	0.068	0.725	0.451	0.728	32.782
	(0.007)	(0.005)	(0.015)	(0.007)	(0.010)	(1.298)	(0.010)	(0.007)	(0.022)	(0.010)	(0.018)	(0.948)
fJGL	0.963	0.416	0.717	0.033	0.842	31.347	0.972	0.395	0.559	0.021	0.769	26.937
	(0.004)	(0.018)	(0.021)	(0.004)	(0.009)	(1.244)	(0.004)	(0.022)	(0.040)	(0.004)	(0.019)	(1.079)
gJGL	0.954	0.347	0.655	0.041	0.807	37.849	0.969	0.291	0.413	0.023	0.695	32.168
	(0.006)	(0.020)	(0.023)	(0.006)	(0.010)	(1.392)	(0.005)	(0.017)	(0.042)	(0.006)	(0.019)	(0.913)
GemBAG$_{\text{MPM}}$	**0.992**	**0.713**	0.516	**0.000**	**0.893**	**18.421**	**0.989**	**0.534**	0.293	**0.000**	**0.893**	**26.442**
	(0.000)	(0.010)	(0.013)	(0.000)	(0.007)	(0.865)	(0.000)	(0.017)	(0.017)	(0.000)	(0.008)	(0.976)

Table 3.6.3: Simulation results for $n = 100$ and $p = 250$ (50 replicates). Methods mGHS and GHS are evaluated over $B = 10000$ post burn-in samples.

$n = 100, p = 500$	*Independent* ($p_0 = 271.25$)						*Coupled* ($p_0 = 279.5$)					
	Acc	MCC	TPR	FPR	AUC	Fr Loss	Acc	MCC	TPR	FPR	AUC	Fr Loss
mGHS$_{\text{MPM}}$	0.518	0.034	**0.845**	0.482	**0.832**	41.906	0.525	0.043	**0.929**	0.476	0.922	39.677
	(0.003)	(0.002)	(0.023)	(0.003)	(0.018)	(1.633)	(0.004)	(0.002)	(0.014)	(0.004)	(0.010)	(1.851)
mGHS$_{t_\alpha}$	0.997	**0.430**	0.461	0.001	**0.832**	41.906	0.994	0.425	0.747	0.006	0.922	39.677
	(0.000)	(0.021)	(0.030)	(0.000)	(0.018)	(1.633)	(0.003)	(0.064)	(0.022)	(0.003)	(0.010)	(1.851)
GemBAG$_{\text{MPM}}$	**0.998**	0.379	0.172	**0.000**	0.799	**40.747**	**0.999**	**0.740**	0.634	**0.000**	**0.962**	**33.100**
	(0.000)	(0.032)	(0.023)	(0.000)	(0.013)	(1.495)	(0.000)	(0.015)	(0.036)	(0.000)	(0.014)	(3.257)
	*P*2020 ($p_0 = 270.5$)						*Full Dependence* ($p_0 = 273$)					
	Acc	MCC	TPR	FPR	AUC	Fr Loss	Acc	MCC	TPR	FPR	AUC	Fr Loss
mGHS$_{\text{MPM}}$	0.522	0.041	**0.915**	0.479	0.909	38.121	0.523	0.043	**0.938**	0.478	0.933	38.741
	(0.004)	(0.002)	(0.018)	(0.003)	(0.015)	(2.107)	(0.004)	(0.002)	(0.016)	(0.004)	(0.013)	(2.042)
mGHS$_{t_\alpha}$	0.979	0.274	0.770	0.020	0.909	38.121	0.979	0.286	0.819	0.020	0.933	38.741
	(0.013)	(0.085)	(0.020)	(0.013)	(0.015)	(2.107)	(0.013)	(0.080)	(0.019)	(0.013)	(0.013)	(2.042)
GemBAG$_{\text{MPM}}$	**0.999**	**0.839**	0.723	**0.000**	**0.966**	**23.992**	**0.999**	**0.873**	0.771	**0.000**	**0.979**	**22.112**
	(0.000)	(0.012)	(0.030)	(0.000)	(0.011)	(4.506)	(0.000)	(0.014)	(0.028)	(0.000)	(0.006)	(3.407)

Table 3.6.4: Simulation results for $n = 100$ and $p = 500$ (25 replicates). Method mGHS is evaluated over $B = 10000$ post burn-in samples.

3.7 Application to a bike-sharing dataset

We perform an analysis of the Capital Bikeshare system data[1], a benchmark dataset previously analyzed in Zhu and Foygel Barber (2015) and Yang et al. (2021). This is the first analysis of this dataset with a full Bayesian graphical model. The dataset contains records of bike rentals in a bicycle sharing system with more than 500 stations located in the Washington D.C. area, where each ride is labeled as *casual* (paying for a single day) or *member* (membership payment). Data from years 2016, 2017, and 2018 are used, for a total of $n = 1092$ registered days. Only the $p = 239$ most active stations are selected. Therefore, for $i = 1, \dots, 1092$ and $j = 1, \dots, 239$, let y_{ij}^c and y_{ij}^m be the number of registered casual and member trips initiated at station j on day i, respectively. After correcting for the seasonal trend, each station data is marginally standardized and transformed with the Yeo-Johnson transformation (Yeo and Johnson, 2000) to better approximate a Gaussian distribution. Finally, the data are divided by year and rider membership for a total of $K = 6$ groups. Matrices $\mathbf{Y}_k$, $k = 1, \dots, 6$ are marginally standardized such that $\boldsymbol{\mu}_k = \mathbf{0}$ and the standard deviations are equal to 1 for each group.

For each class, 80% of the observations are used as training set and the remaining 20% as test set. For $k = 1, \dots, 6$, let $\hat{\boldsymbol{\Omega}}_k$ be the estimated precision matrix of the k-th training set. Here we take the posterior mean. Following Fan et al. (2009), the observations of each test set is partitioned as $\mathbf{y}_i^k = \left(\mathbf{y}_{i,j_1}^k, \mathbf{y}_{i,j_2}^k\right)$, where $\mathbf{y}_{i,j_1}^k = \left(y_{i,1}^k, \dots, y_{i,120}^k\right)$ and $\mathbf{y}_{i,j_2}^k = \left(y_{i,121}^k, \dots, y_{i,239}^k\right)$, $i = 1, \dots, n_k$. The corresponding partition for $\boldsymbol{\Omega}_k$ and $\boldsymbol{\Sigma}_k$ are

$$\boldsymbol{\Omega}_k = \begin{bmatrix} \boldsymbol{\Omega}_{k_{11}} & \boldsymbol{\Omega}_{k_{12}} \\ \boldsymbol{\Omega}_{k_{21}} & \boldsymbol{\Omega}_{k_{22}} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}_k = \begin{bmatrix} \boldsymbol{\Sigma}_{k_{11}} & \boldsymbol{\Sigma}_{k_{12}} \\ \boldsymbol{\Sigma}_{k_{21}} & \boldsymbol{\Sigma}_{k_{22}} \end{bmatrix}.$$

The performances of the models are evaluated by predicting $\mathbf{y}_{i,j_2}^k$ based on $\mathbf{y}_{i,j_1}^k$ and $\hat{\boldsymbol{\Omega}}_k$. Under the Gaussian assumption, the best linear predictor is

$$\hat{\mathbf{y}}_{i,j_2}^k = \mathbb{E}\left(\mathbf{y}_{i,j_2}^k \mid \mathbf{y}_{i,j_1}^k\right) = \hat{\boldsymbol{\Sigma}}_{k_{21}} \hat{\boldsymbol{\Sigma}}_{k_{11}}^{-1} \mathbf{y}_{i,j_1}^k.$$

To assess the prediction performances of the methods we rely on the *average absolute forecast error* (AAFE), defined as

$$\text{AAFE}_k = \frac{1}{119} \frac{1}{|\mathbb{T}_k|} \sum_{i \in \mathbb{T}_k} \sum_{j=121}^{239} |y_{ij}^k - \hat{y}_{ij}^k|,$$

where $\mathbb{T}_k$ denotes the test set indexes for group k. We denote the mean AAFE across groups as mAAFE.

[1]Data are available at http://www.capitalbikeshare.com/system-data

The multiple Graphical Horseshoe is tested against the ordinary Graphical Horseshoe of Li et al. (2019) and the GemBAG of Yang et al. (2021). For the estimation of the threshold in the mGHS model we set the hyperparameter to $a = 30$ and $b = 25$, whereas in GemBAG we estimated hyperparameters v_0 and v_1 according to the BIC criterion as in Section 3.6. For computational reasons, the joint Graphical LASSO of Danaher et al. (2014) is excluded from the analysis.

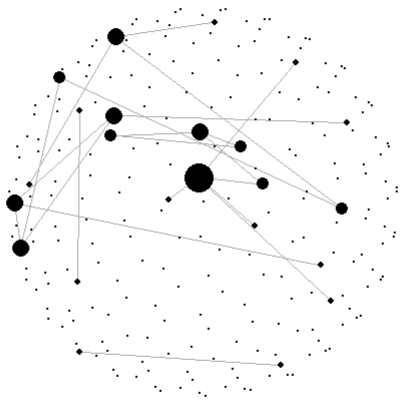
With mAAFE = 0.596, the best predictive model is the mGHS, whereas the ordinary GHS shows similar predictive performance (mAAFE = 0.600). The latter, however, provides a sparser model: regardless of the method used for selecting the edges a posteriori, the mGHS always estimates denser networks, including connections between stations that the GHS is not able to capture. Finally, the GemBAG provides at the same time the sparsest model and the worst predictive performance, with mAAFE = 0.613.

To further understand how the connections between stations work among the casual and member users, we plot the estimated networks for each group for both GHS and mGHS (Figures 3.D.2 and 3.D.1 in Appendix 3.D), where we select those edges with a posterior inclusion probability higher than 0.9. The estimated networks for casual users are denser in both models, suggesting a higher activity of casual rides. However, the number of edges shared across the years is higher for the registered users, implying more regular activities of those who choose to pay a seasonal ticket. The intersection of the estimated networks across three years for the registered and casual users is shown in Figure 3.7.1 for both GHS and mGHS, where the size of the nodes depends on the number of edges associated with the related stations. The two models estimate similar networks for both types of users, however, mGHS gives more importance to the stations identified by GHS and includes some additional ones.

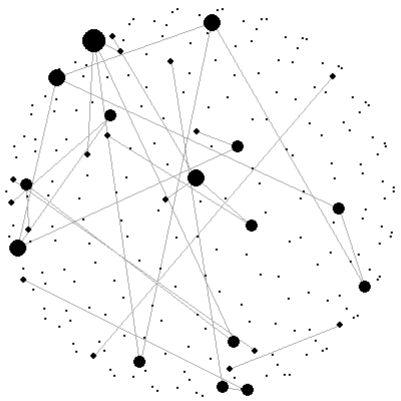
casual 2016	casual 2017	casual 2018	member 2016	member 2017	member 2018	
1.000	0.969	0.893	0.479	0.515	0.483	casual 2016
0.969	1.000	0.958	0.518	0.562	0.526	casual 2017
0.893	0.958	1.000	0.461	0.502	0.475	casual 2018
0.479	0.518	0.461	1.000	0.984	0.971	member 2016
0.515	0.562	0.502	0.984	1.000	0.980	member 2017
0.483	0.526	0.475	0.971	0.980	1.000	member 2018

(3.14)

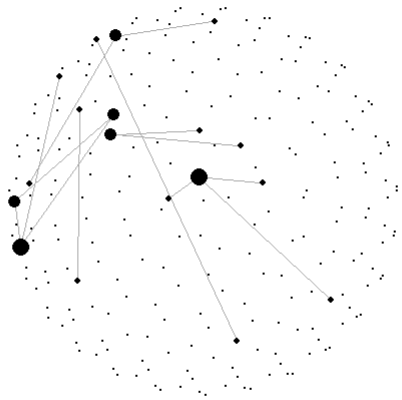
The hypothesis of a more regular behaviour of the registered users is supported also by 3.14, which reports the estimated correlation matrix between groups. The correlation is high across the years for both types of users. In particular, it remains close to 1 even after two years for the rides with membership payment (correlation between 2016 and 2018 is 0.971). On the contrary, the decrease is larger for the casual rides, with a correlation of 0.893.



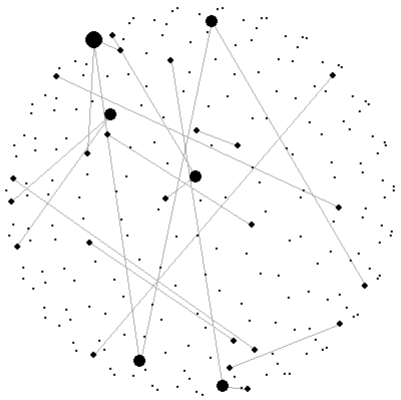
(a) Casual network estimated by mGHS



(b) Member network estimated by mGHS



(c) Casual network estimated by GHS



(d) Member network estimated by GHS

Figure 3.7.1: Intersection of the estimated networks across three years; the size of the nodes depends on the number of edges associated to the related station

3.8 Conclusion

In this paper, we have introduced a novel fully Bayesian method for the analysis of high-dimensional dependent precision matrices. In particular, we provided an efficient approach that works up to hundreds of variables. We empirically showed that the model is able to borrow information between groups when appropriately supported by the data. Simulation studies empirically demonstrated that the proposed approach has good performances in terms of edge selection; the proposed joint model performs at least as well as the separate analysis of each group with the ordinary Graphical Horseshoe (Li et al., 2019). We applied our method to a benchmark dataset with a slight improvement in prediction performance. Compared to the ordinary Graphical Horseshoe, the proposed model borrowed information across groups and selected a higher number of common edges across the years. Moreover, the estimation of correlation matrix $\mathbf{R}$ provided unique insights about the behavior of bike-sharing users.

We proposed a new approach for posterior edge selection that accounts for posterior dependencies between parameters $\lambda^2_{ij,k}$'s. This method can be easily extended to other common frameworks, for example, variable selection in regression models. The simulation results of our approach are promising, however, among other unexplored properties, it is not clear whether the proposal distribution accounts for easy control of false discoveries, a desirable feature for every posterior selection method. This can be assessed by evaluating different proposal distributions and comparing our procedure against the alternative solutions proposed by Chandra et al. (2022) and Lee et al. (2023). Further improvements in the proposal may concern the introduction of different thresholds t^{α}_{ij} specific for each edge, the application of adaptive procedures or the construction of a distribution which reflects the methods of Muller et al. (2007), Chandra et al. (2022) and Lee et al. (2023). The proposed cut model provides only an approximation of the posterior distribution, and, in models with cuts in general, the algorithm may fail to converge to a well-defined distribution (Plummer, 2015). To this aim, the author propose an approximate solution called tempered cut algorithm with the goal of overcoming the problem of convergence. Whereas cut models can outperform fully Bayesian models in terms of performance and computational efficiency, a careful assessment of the output produced by models with cuts should be always performed. We leave the cited improvements of our edge selection method to future works.

Note that very recently Lingjaerde et al. (2022) have proposed an approach, alternative to the one presented in this paper, for the analysis of multiple graphical models with horseshoe priors, termed the joint graphical horseshoe. The approach proposed in this paper, with respect to the joint graphical horseshoe, is characterized by a few important and unique features, since it provides full Bayesian

inference, it adapts well to setting with heterogeneous levels of network similarity, it learns the level of network similarity across groups from the data, and it has been successfully applied to networks with large p (up to 500 nodes).

Among possible extensions, we may consider a spike-and-slab type of prior on the off-diagonal elements of the correlation matrix $\mathbf{R}$. This approach would not only give a deeper insight into the similarity across the groups, but it would speed the model up when the groups are not significantly related: the $\mathcal{G}_{3p}$ distribution would reduce to an Inverse-Gamma when the k-th row of the matrix $\mathbf{R}$ is zero, avoiding the need of the rejection sampling discussed in Section 3.3.

A main challenge, and still a limitation, of the proposed approach, is the computational complexity of the algorithm since it becomes infeasible when the number of covariates p is extremely large, e.g., in the thousands. Alternative computational approaches that could be explored include the thresholding approach of Johndrow et al. (2020) that could be adapted to sample from multivariate Normal distributions under the Horseshoe prior, and eventually lead to a significant reduction in computational times.

The R code for mGHS model, simulations studies and application to bike-sharing dataset is available at https://github.com/cbusatto/mGHS.

Appendix

Appendix 3.A The three-parameter Gamma distribution

3.A.1 Technical details of the modified rejection sampling method

The acceptance probability of each step of the algorithm is compute as follows:

- Step 1: the probability of immediate acceptance is

$$P(E_1) = \Phi_{0,\omega^2}(t_2) - \Phi_{0,\omega^2}(t_1),$$

  where $\Phi_{\mu,\sigma^2}(\cdot)$ denotes the cumulative density function of a Gaussian distribution with mean μ and variance σ^2;

- Step 2: the acceptance probability of Step 2 is

$$P(E_2) = 1 - P(E_1) - P(E_3),$$

  where $P(E_3)$ is the acceptance probability of Step 3.

- Step 3: the probability of acceptance this step is

$$\begin{aligned} P(E_3) &= \int_{-\infty}^{t_1} h(t)dt + \int_{t_2}^{\infty} h(t)dt - \int_{-\frac{\mu}{\sigma}}^{t_1} g(t)dt + \int_{t_2}^{\infty} g(t)dt \\ &= \int_{-\infty}^{\infty} h(t)dt - \int_{t_1}^{t_2} h(t)dt - \int_{-\frac{\mu}{\sigma}}^{\infty} g(t)dt + \int_{t_1}^{t_2} g(t)dt \\ &= \int_{t_1}^{t_2} g(t) - h(t)dt. \end{aligned}$$

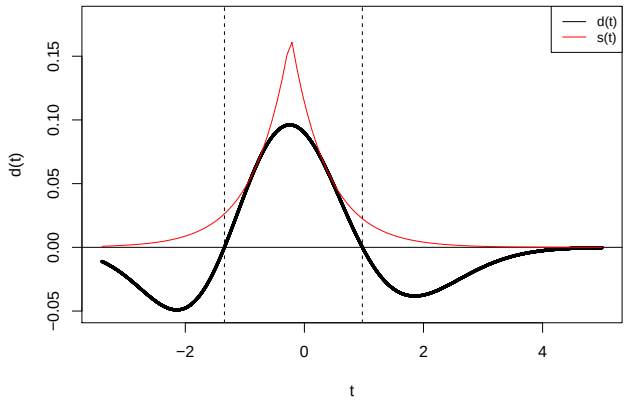


Figure 3.A.1: distributions $d(t)$ and $s(t)$; dotted lines represent t_1 and t_2.

3.A.2 Rejection sampling for sampling from the difference distribution $d(t)$

Sampling from $d(t)$ in Step 3 can be achieved by means of a standard rejection sampling. Let $s(t)$ be the proposal distribution, we adapt a double-exponential (Laplace) distribution of the form

$$s(t) = \frac{c}{\sqrt{2\pi}} e^{-\frac{|t-b|}{\delta}}, \quad -\infty < t < \infty,$$

in order to minimize the area between $s(t)$ and $d(t)$ (Ahrens and Dieter, 1982; Stadlober, 1982). This happens when the hat function $s(t)$ touches $d(t)$ at two different points L and R, with $R > L$. As explained in Dieter (1981), if $d(t)$ is covered by a double-exponential distribution, optimal parameters c, b, and δ can be estimated within two steps: first, points L, R and parameter δ are computed simultaneously (for instance by Newton iteration) as

$$\begin{aligned} d'(L) &= \frac{1}{\delta} d(L) \\ d'(R) &= -\frac{1}{\delta} d(R) \\ \delta &= \frac{1}{2}(R - L), \end{aligned}$$

whereas parameters c and b are calculated as

$$b = \frac{1}{2}\left(L + R + \delta ln\left(\frac{d(R)}{d(L)}\right)\right)$$
$$c = e\sqrt{2\pi d(R)d(L)}.$$

Figure 3.A.1 shows the difference function $d(t)$ and its optimal hat function $s(t)$.

The algorithm can be further sped up by noting that the quantities t_1, t_2, b, c and δ only depend on the ratio β/α. The computation of these parameters, which involve iterative methods, can be avoided by tabulating the needed quantities for a restricted grid of the parameters γ, α, and β.

3.A.3 Proof of Proposition 3.3.1

Recalling that $t = (x - \mu)/\sigma$, where $x > 0$, the acceptance probability of the first two steps of the algorithm can be computed as

$$\begin{aligned}
\mathbb{P}(T_{acc}) &= \mathbb{P}\left(U \leq \frac{g(t)}{h(t)}\right) \\
&= \int_{-\frac{\mu}{\sigma}}^{\infty} \mathbb{P}\left(U \leq \frac{g(t)}{h(t)} \middle| T = t\right) h(t)dt \\
&= \int_{-\frac{\mu}{\sigma}}^{t_1} \frac{g(t)}{h(t)} h(t)dt + \int_{t_1}^{t_2} h(t)dt + \int_{t_2}^{\infty} \frac{g(t)}{h(t)} h(t)dt \\
&= \int_{-\frac{\mu}{\sigma}}^{t_1} g(t)dt + \int_{t_1}^{t_2} h(t)dt + \int_{t_2}^{\infty} g(t)dt.
\end{aligned}$$

Thus, the probability of rejection is $\mathbb{P}(T_{rej}) = 1 - \mathbb{P}(T_{acc}) = \int_{t_1}^{t_2} g(t) - h(t)dt$. Since the Step 3 draws a sample from $\int_{t_1}^{t_2} g(t) - h(t)dt$, the acceptance probability of the method is exactly 1.

To show that the distribution of accepted values follows the target density $g(t)$, the cumulative density function $\mathbb{P}\left(T \leq u \middle| T_{acc}\right) = \frac{\mathbb{P}(T \leq u, T_{acc})}{\mathbb{P}(T_{acc})} = \mathbb{P}(T \leq u, T_{acc})$ has to be equal to $F_{g(t)}(u) = \int_{-\frac{\mu}{\sigma}}^{u} g(t)dt$. Three different cases are studied:

- **Case** $u < t_1$:

$$\begin{aligned}
\mathbb{P}(T \leq u, T_{acc}) &= \int_{-\frac{\mu}{\sigma}}^{u} \mathbb{P}\left(U \leq \frac{g(t)}{h(t)}\right) h(t)dt \\
&= \int_{-\frac{\mu}{\sigma}}^{u} g(t)dt \\
&= F_{g(t)}(u);
\end{aligned}$$

- **Case** $u \in [t_1, t_2]$:

$$\begin{aligned}
\mathbb{P}\left(T \leq u, T_{acc}\right) &= \mathbb{P}\left(T \leq t_1, T_{acc}\right) + \mathbb{P}\left(t_1 < T \leq u, T_{acc}\right) \\
&= F_{g(t)}(t_1) + \int_{t_1}^{u} \mathbb{P}\left(U \leq \frac{g(t)}{h(t)}\right) h(t)dt + \int_{t_1}^{u} g(t) - h(t)dt \\
&= F_{g(t)}(t_1) + \int_{t_1}^{u} h(t)dt + \int_{t_1}^{u} g(t) - h(t)dt \\
&= F_{g(t)}(u);
\end{aligned}$$

- **Case** $t_2 < u$:

$$\begin{aligned}
\mathbb{P}\left(T \leq u, T_{acc}\right) &= \mathbb{P}\left(T \leq t_2, T_{acc}\right) + \mathbb{P}\left(t_2 < T \leq u, T_{acc}\right) \\
&= F_{g(t)}(t_2) + \int_{t_2}^{u} \mathbb{P}\left(U \leq \frac{g(t)}{h(t)}\right) h(t)dt \\
&= F_{g(t)}(t_2) + \int_{t_2}^{u} g(t)dt \\
&= F_{g(t)}(u).
\end{aligned}$$

Therefore, the method actually samples from the target distribution.

Appendix 3.B KL divergence for the three-parameter Gamma distribution

Here the asymptotic behaviour of a $\mathcal{G}_{3p}$ distribution for limit cases $\beta/\alpha \to -\infty$, $\beta/\alpha \to \infty$ and $\gamma \to \infty$ is described. The analysis relies on the KL divergence. In the first case, $\beta/\alpha \to -\infty$ the $\mathcal{G}_{3p}$ distribution is compared to a Gamma distribution and yields a closed-form result, whereas when $\beta/\alpha \to \infty$ and $\gamma \to \infty$ the target density is approximated with a Gaussian distribution based on empirical results.

- **Proof of Proposition 3.3.2**:
  The KL divergence between distribution $q_x \sim \mathcal{G}_{3p}\left(\gamma, \alpha, \beta\right)$ and distribution $p_x \sim \mathsf{Ga}\left(d, c\right)$ is

$$\mathrm{KL}\left(p \| q\right) = \int_0^\infty p_x \log\left(\frac{p_x}{q_x}\right) dx = \int_0^\infty p_x \log\left(p_x\right) dx - \int_0^\infty p_x \log\left(q_x\right) dx. \tag{3.15}$$

Denoting the two integrals in (3.15) as $I(d,c) = \int_0^\infty p_x \log(p_x)\, dx$ and $I(d,c,\gamma,\alpha,\beta) = \int_0^\infty p_x \log(q_x)\, dx$, it yields

$$\begin{aligned}
I(d,c) &= \int_0^\infty \log\left(\frac{c^d}{\Gamma(d)} e^{-cx} x^{d-1}\right) \frac{c^d}{\Gamma(d)} e^{-cx} x^{d-1} dx \\
&= \log\left(\frac{c^d}{\Gamma(d)}\right) - \frac{c^{d+1}}{\Gamma(d)} \int_0^\infty e^{-cx} x^d dx + \frac{c^d(d-1)}{\Gamma(d)} \int_0^\infty \log(x) e^{-cx} x^{d-1} dx \\
&= \log\left(\frac{c^d}{\Gamma(d)}\right) - \frac{c^{d+1}}{\Gamma(d)} \frac{\Gamma(d+1)}{c^{d+1}} + \frac{c^d(d-1)}{\Gamma(d)} \frac{\Gamma(d)}{c^d} \left(\frac{\Gamma'(d)}{\Gamma(d)} - \log c\right) \\
&= \log\left(\frac{c^d}{\Gamma(d)}\right) - d + (d-1)\left(\frac{\Gamma'(d)}{\Gamma(d)} - \log c\right) \\
&= \log(c) + (d-1)\frac{\Gamma'(d)}{\Gamma(d)} - d - \log(\Gamma(d))
\end{aligned}$$

and

$$\begin{aligned}
I(d,c,\gamma,\alpha,\beta) &= \int_0^\infty \log\left(\frac{(2\alpha^2)^{\frac{\gamma+1}{2}} e^{-\frac{\beta^2}{8\alpha^2}}}{\gamma! D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)} e^{-\alpha^2 x^2 + \beta x} x^\gamma\right) \frac{c^d}{\Gamma(d)} e^{-cx} x^{d-1} dx \\
&= \log\left(\frac{(2\alpha^2)^{\frac{\gamma+1}{2}} e^{-\frac{\beta^2}{8\alpha^2}}}{\gamma! D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}\right) + \frac{c^d}{\Gamma(d)} \int_0^\infty \left(-\alpha^2 x^2 + \beta x\right) e^{-cx} x^{d-1} dx \\
&\quad + \frac{c^d \gamma}{\Gamma(d)} \int_0^\infty \log(x) e^{-cx} x^{d-1} dx \\
&= \log\left(\frac{(2\alpha^2)^{\frac{\gamma+1}{2}} e^{-\frac{\beta^2}{8\alpha^2}}}{\gamma! D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}\right) - \frac{\alpha^2 c^d \Gamma(d+2)}{c^{d+2}\Gamma(d)} + \frac{\beta c^d \Gamma(d+1)}{c^{d+1}\Gamma(d)} + \\
&\quad \gamma\left(\frac{\Gamma'(d)}{\Gamma(d)} - \log(c)\right) \\
&= \log\left(\frac{(2\alpha^2)^{\frac{\gamma+1}{2}} e^{-\frac{\beta^2}{8\alpha^2}}}{\gamma! D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}\right) - \frac{\alpha^2 d(d+1)}{c^2} + \frac{\beta d}{c} + \gamma\left(\frac{\Gamma'(d)}{\Gamma(d)} - \log(c)\right).
\end{aligned}$$

Thus,

$$
\begin{aligned}
I\left(d,c\right)-I\left(d,c,\gamma,\alpha,\beta\right) = & \log(c)+(d-1)\frac{\Gamma'\left(d\right)}{\Gamma(d)}-d-\log\left(\Gamma(d)\right)- \\
& \log\left(\frac{\left(2\alpha^{2}\right)^{\frac{\gamma+1}{2}}e^{-\frac{\beta^{2}}{8\alpha^{2}}}}{\gamma!D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}\right)+\frac{\alpha^{2}d(d+1)}{c^{2}}-\frac{\beta d}{c}- \\
& \gamma\left(\frac{\Gamma'\left(d\right)}{\Gamma(d)}-\log\left(c\right)\right) \\
= & \left(\gamma+1\right)\log\left(c\right)+\left(d-1-\gamma\right)\frac{\Gamma'\left(d\right)}{\Gamma(d)}-d\left(1+\frac{\beta}{c}-\frac{\alpha^{2}(d+1)}{c^{2}}\right) \\
& +\log\left(\frac{\Gamma(\gamma+1)}{\Gamma(d)}\right)-\log\left(\frac{\left(2\alpha^{2}\right)^{\frac{\gamma+1}{2}}e^{-\frac{\beta^{2}}{8\alpha^{2}}}}{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}\right).
\end{aligned}
\tag{3.16}
$$

Let $d=\frac{\mu^{2}}{\sigma^{2}}$ and $c=\frac{\mu}{\sigma^{2}}$ so that the Gamma distribution has the same mean and variance of the $\mathcal{G}_{3p}$ distribution. Exploiting the properties of the Parabolic Cylinder functions it yields

$$
\begin{aligned}
\lim_{\frac{\beta}{\alpha}\to-\infty}d = & \lim_{\frac{\beta}{\alpha}\to-\infty}\frac{\left(\gamma+1\right)^{2}D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)^{2}}{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}\left(\frac{\gamma+2}{\gamma+1}D_{-\gamma-3}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)-\right. \\
& \left.\frac{D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)^{2}}{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}\right)^{-1} \\
= & \frac{1}{\lim_{\frac{\beta}{\alpha}\to-\infty}\frac{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}{(\gamma+1)D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)^{2}}\left(\left(\gamma+2\right)D_{-\gamma-3}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)-\left(\gamma+1\right)\frac{D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)^{2}}{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}\right)} \\
= & \frac{1}{\lim_{\frac{\beta}{\alpha}\to-\infty}\frac{(\gamma+2)D_{-\gamma-3}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}{(\gamma+1)D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)^{2}}-1} \\
= & \frac{1}{\frac{\gamma+2}{\gamma+1}-1}=\gamma+1
\end{aligned}
$$

and

$$\begin{aligned}
\lim_{\frac{\beta}{\alpha}\to-\infty} c &= \lim_{\frac{\beta}{\alpha}\to-\infty} \frac{d}{\mu} \\
&= (\gamma+1) \lim_{\frac{\beta}{\alpha}\to-\infty} \frac{1}{\mu} \\
&= (\gamma+1) \lim_{\frac{\beta}{\alpha}\to-\infty} \frac{\alpha\sqrt{2}}{\gamma+1} \frac{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}{D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)} \\
&= \lim_{\frac{\beta}{\alpha}\to-\infty} \alpha\sqrt{2} \frac{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}{D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)} \quad \left(D_v(z) = zD_{v-1}(z) - (v-1)\, D_{v-2}(z)\right) \\
&= \lim_{\frac{\beta}{\alpha}\to-\infty} \alpha\sqrt{2} \left(-\frac{\beta}{\alpha\sqrt{2}} + (\gamma+2) \frac{D_{-\gamma-3}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}{D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)} \right) \\
&= -\beta.
\end{aligned}$$

Plugging these results into (3.16) yields

$$\begin{aligned}
KL_{\frac{\beta}{\alpha}\to-\infty}(q,p) &= (\gamma+1)\log(-\beta) - (\gamma+1)\left(\frac{\alpha^2(d+1)}{\beta^2}\right) - \log\left(\frac{(2\alpha^2)^{\frac{\gamma+1}{2}} e^{-\frac{\beta^2}{8\alpha^2}}}{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}\right) \\
&= \log\left(\left(\frac{-\beta}{\alpha\sqrt{2}}\right)^{\gamma+1} \frac{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}{e^{-\frac{\beta^2}{4(2\alpha^2)}}}\right) - (\gamma+1)\left(\frac{\alpha^2(d+1)}{\beta^2}\right) = 0,
\end{aligned}$$

since $\lim_{z\to\infty} \frac{D_{-v}(z)}{z^{-v}e^{\frac{-z^2}{4}}} = 1$

- **Asymptotic behaviour when $\frac{\beta}{\alpha} \to +\infty$ or $\gamma \to +\infty$:**
  When $\frac{\beta}{\alpha} \to +\infty$ the Gamma-3p is approximated with a $\mathcal{N}(\mu, \sigma^2)$ distribution, with

$$\begin{aligned}
\lim_{\frac{\beta}{\alpha}\to+\infty} \mu &= \lim_{\frac{\beta}{\alpha}\to+\infty} \frac{\gamma+1}{\alpha\sqrt{2}} \frac{D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)} \quad \left(\lim_{x\to-\infty} v\frac{D_{-v-1}(z)}{D_{-v}(z)} = -z\right) \\
&= \frac{\beta}{2\alpha^2} \qquad (3.17)
\end{aligned}$$

and

$$\lim_{\frac{\beta}{\alpha}\to+\infty}\sigma^2 = \lim_{\frac{\beta}{\alpha}\to+\infty}\frac{\gamma+1^2}{2\alpha^2}\frac{D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}\left(\frac{\gamma+2}{\gamma+1}\frac{D_{-\gamma-3}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}{D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)} - \frac{D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}\right)$$
$$= \frac{1}{2\alpha^2}. \tag{3.18}$$

Following Segura (2021), when $v \to \infty$ a sharp approximation for the ratio of Parabolic Cylinder functions is $v\frac{D_{-v-1}(z)}{D_{-v}(z)} \approx -z + \frac{1}{2}\left(z + \sqrt{z^2+4v-2}\right)$. Therefore, the mean and variance of the Gaussian approximation become

$$\lim_{\gamma\to+\infty}\mu = \lim_{\gamma\to+\infty}\frac{\gamma+1}{\alpha\sqrt{2}}\frac{D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}$$
$$= \frac{\beta}{4\alpha^2} + \frac{1}{\alpha\sqrt{8}}\sqrt{\frac{\beta^2}{2\alpha^2}+4\gamma+2} \tag{3.19}$$

and

$$\lim_{\gamma\to+\infty}\sigma^2 = \lim_{\gamma\to+\infty}\frac{\gamma+1^2}{2\alpha^2}\frac{D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}\left(\frac{\gamma+2}{\gamma+1}\frac{D_{-\gamma-3}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}{D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)} - \frac{D_{-\gamma-2}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}{D_{-\gamma-1}\left(-\frac{\beta}{\alpha\sqrt{2}}\right)}\right)$$
$$= \frac{\gamma+1}{4\alpha^2}\left(\frac{\beta}{\alpha\sqrt{2}} + \sqrt{\frac{\beta^2}{2\alpha^2}+4\gamma+6}\right) - \lim_{\gamma\to+\infty}\mu^2$$
$$= \frac{\gamma+1}{4\alpha^2}\left(\frac{\beta}{\alpha\sqrt{2}} + \sqrt{\frac{\beta^2}{2\alpha^2}+4\gamma+6}\right) - \left(\frac{\beta}{4\alpha^2} + \frac{1}{\alpha\sqrt{8}}\sqrt{\frac{\beta^2}{2\alpha^2}+4\gamma+2}\right)^2. \tag{3.20}$$

Tables 3.B.1 and 3.B.2 show the KL divergence for increasing values of the ratio β/α and γ. The integral is numerically approximated with the command KLD from package LaplacesDemon for software R. The approximated KL divergence is evaluated over the interval $(\mu - 5\sigma, \mu + 5\sigma)$. Values of the parameters higher than those shown in the table 3.B.1 give overflow problems. The results in the tables below depend only on the values of γ and the ratio β/α, that is, for different values of α the KL divergence between $q \sim \mathcal{G}_{3p}(\gamma, \alpha, \beta)$ and $p \sim \mathcal{N}(\mu, \sigma^2)$ does not change. The sequence of KL divergence is always decreasing in Table 3.B.2, for both KL($q\|p$) and KL($p\|q$). In Table 3.B.1 the sequence is decreasing only for KL($p\|q$), however the mean between the two is decreasing.

KL	$\frac{\beta}{\alpha}=0.002$	$\frac{\beta}{\alpha}=0.2$	$\frac{\beta}{\alpha}=0.5$	$\frac{\beta}{\alpha}=1$	$\frac{\beta}{\alpha}=3$	$\frac{\beta}{\alpha}=5$	$\frac{\beta}{\alpha}=8$	$\frac{\beta}{\alpha}=0.002$	$\frac{\beta}{\alpha}=0.2$	$\frac{\beta}{\alpha}=0.5$	$\frac{\beta}{\alpha}=1$	$\frac{\beta}{\alpha}=3$	$\frac{\beta}{\alpha}=5$	$\frac{\beta}{\alpha}=8$
$\gamma=1$	0.284	0.273	0.257	0.227	0.105	0.041	0.016	0.411	0.394	0.372	0.329	0.139	0.047	0.016
$\gamma=3$	1.206	1.164	1.100	0.983	0.545	0.281	0.127	2.365	2.288	2.153	1.906	0.886	0.355	0.139
$\gamma=5$	2.185	2.115	2.007	1.820	1.104	0.636	0.318	5.032	4.858	4.577	4.064	1.992	0.871	0.366
$\gamma=10$	4.633	4.505	4.319	3.995	2.736	1.810	1.038	13.207	12.743	12.057	10.804	5.698	2.811	1.303
$\gamma=15$	6.875	6.740	6.516	6.124	4.501	3.206	1.990	22.597	21.884	20.721	18.653	10.295	5.419	2.672
$\gamma=30$	10.882	10.839	10.751	10.537	9.160	7.568	5.517	53.998	52.316	49.941	45.377	27.042	15.766	8.645
$\gamma=50$	12.710	12.739	12.749	12.708	12.090	11.137	9.542	97.765	95.186	90.974	83.411	52.121	32.489	19.310
$\gamma=100$	14.162	14.225	14.278	14.349	14.163	13.755	13.110	208.476	203.439	195.323	180.003	117.316	77.428	49.973

Table 3.B.1: KL divergence when β/α increases: $\mathrm{KL}(q\|p)$ (left) and $\mathrm{KL}(p\|q)$ (right) where $q \sim \mathcal{G}_{3p}(\gamma, \alpha, \beta)$ and $p \sim \mathcal{N}(\mu, \sigma^2)$, with μ and σ^2 computed as in (3.17)-(3.18).

KL	$\frac{\beta}{\alpha}=0.002$	$\frac{\beta}{\alpha}=0.2$	$\frac{\beta}{\alpha}=0.5$	$\frac{\beta}{\alpha}=1$	$\frac{\beta}{\alpha}=3$	$\frac{\beta}{\alpha}=5$	$\frac{\beta}{\alpha}=8$	$\frac{\beta}{\alpha}=0.002$	$\frac{\beta}{\alpha}=0.2$	$\frac{\beta}{\alpha}=0.5$	$\frac{\beta}{\alpha}=1$	$\frac{\beta}{\alpha}=3$	$\frac{\beta}{\alpha}=5$	$\frac{\beta}{\alpha}=8$
$\gamma=1$	0.022	0.021	0.018	0.016	0.011	0.007	0.004	0.023	0.022	0.020	0.017	0.010	0.007	0.003
$\gamma=3$	0.015	0.013	0.012	0.010	0.005	0.004	0.002	0.025	0.023	0.020	0.015	0.005	0.004	0.002
$\gamma=5$	0.010	0.009	0.008	0.006	0.003	0.002	0.002	0.016	0.015	0.013	0.010	0004	0.002	0.002
$\gamma=10$	0.005	0.004	0.004	0.003	0.002	0.001	0.001	0.006	0.006	0.005	0.004	0.002	0.001	0.001
$\gamma=15$	0.003	0.003	0.003	0.002	0.001	0.001	<0.001	0.003	0.003	0.003	0.003	0.001	0.001	<0.001
$\gamma=30$	0.001	0.001	0.001	0.001	<0.001	<0.001	<0.001	0.002	0.001	0.001	0.001	<0.001	<0.001	<0.001
$\gamma=50$	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
$\gamma=100$	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001

Table 3.B.2: KL divergence when γ increases: $\mathrm{KL}(q\|p)$ (left) and $\mathrm{KL}(p\|q)$ (right) where $q \sim \mathcal{G}_{3p}(\gamma, \alpha, \beta)$ and $p \sim \mathcal{N}(\mu, \sigma^2)$, with μ and σ^2 computed as in (3.19)-(3.20).

Appendix 3.C Pseudo-code for mGHS algorithm

Algorithm 12: Multiple Graphical Horseshoe algorithm

Input: $\mathbf{S}_1, \ldots, \mathbf{S}_K \in \mathbb{R}^{p\times p}$, $K, p, B, bn \in \mathbb{N}$, $\mathbf{n} \in \mathbb{N}^K$;
set $\boldsymbol{\Omega}_k = \mathbf{I}_p$, $\boldsymbol{\Sigma}_k = \mathbf{I}_p$, $\boldsymbol{\Lambda}_k = \mathbf{1}_{p\times p}$, $\boldsymbol{\eta}_k = \mathbf{1}_{p\times p}$, $\boldsymbol{\tau} = \mathbf{1}_K$, $\boldsymbol{\zeta} = \mathbf{1}_K$, $\mathbf{R} = \mathbf{I}_K$ and $\boldsymbol{\mu} = \mathbf{0}_K$;
for $b = 1$ *to* B **do**
  for $k = 1$ *to* K **do**
    for $j = 1$ *to* p **do**
      compute $\mathbf{t}_{j,k} := \left\{t^i_{j,k} = \mathbf{r}_k^\intercal \mathbf{R}^{-1}_{-k} \boldsymbol{\Delta}^{-1}_{ij,-k} \boldsymbol{\omega}^{-k}_{ij},\ i = 1, \ldots, p, i \neq j\right\}$;
      compute $\mathbf{m}_{j,k} := \left\{m^i_{j,k} = t^i_{j,k}\sqrt{\tau_k \lambda^k_{ij}},\ i = 1, \ldots, p, i \neq j\right\}$;
      compute $\mathbf{D}_{j,k} := \left\{D^{ii}_{j,k} = \mu_k \tau_k \lambda^k_{ij},\ i = 1, \ldots, p, i \neq j\right\}$;
      compute $\mathbf{O}_{j,k} = \boldsymbol{\Sigma}^k_{-j} - \boldsymbol{\sigma}^k_j \left(\boldsymbol{\sigma}^k_j / \sigma^k_{jj}\right)^\intercal$ and $\mathbf{W}_{j,k} = \mathbf{D}_{j,k} + s^k_{jj}\mathbf{O}_{j,k}$;
      sample $\gamma^k_{jj} \sim \mathcal{IG}\left(n_k/2 + 1, s^k_{jj}/2\right)$;
      sample $\mathbf{v}_{j,k} \sim \mathcal{N}_{p-1}\left(\mathbf{W}^{-1}_{j,k}\left(\mathbf{D}^{-1}_{j,k}\mathbf{m}_{j,k} - \mathbf{s}^k_{-j}\right), \mathbf{W}^{-1}_{j,k}\right)$;
      sample $\mathbf{e}_{j,k} := \left\{e^i_{j,k} \sim \mathcal{IG}\left(1, 1 + 1/l^i_{j,k}\right),\ i = 1, \ldots, p, i \neq j\right\}$;
      sample $\mathbf{l}_{j,k} := \left\{l^i_{j,k} \sim \mathcal{G}_{3p}\left(1, \alpha_{\lambda_{ij,k}}, \beta_{\lambda_{ij,k}}\right)^{-2},\ i = 1, \ldots, p, i \neq j\right\}$, where $\alpha_{\lambda_{ij,k}} = \left(\frac{v_i^2}{2\tau_k\mu_k} + \frac{1}{\eta^k_{ij}}\right)^{1/2}$ and $\beta_{\lambda_{ij,k}} = \frac{v_i}{\sqrt{\tau_k}\mu_k} t^i_{j,k}$;
      set $\boldsymbol{\Sigma}^k_{-j} = \mathbf{O}_{j,k} + \mathbf{O}_{j,k}\mathbf{v}_{j,k}\mathbf{v}^\intercal_{j,k}\mathbf{O}_{j,k}/\gamma^k_{jj}$, $\boldsymbol{\sigma}^k_j = -\mathbf{O}_{j,k}\mathbf{v}_{j,k}/\gamma^k_{jj}$, $\sigma^k_{jj} = 1/\gamma^k_{jj}$, $\boldsymbol{\omega}^k_j = \mathbf{v}_{j,k}$, $\omega^k_{jj} = \gamma^k_{jj} + \mathbf{v}^\intercal_{j,k}\mathbf{O}_{j,k}\mathbf{v}_{j,k}$, $\boldsymbol{\lambda}^k_j = \mathbf{l}_{j,k}$ and $\boldsymbol{\eta}^k_j = \mathbf{e}_{j,k}$;
    end
  end
  compute $\mathbf{T}_k := \left\{T^{ij}_k = \mathbf{r}_k^\intercal \mathbf{R}^{-1}_{-k} \boldsymbol{\Delta}^{-1}_{ij,-k} \boldsymbol{\omega}^{-k}_{ij},\ k = 1, \ldots, K, j = 2, \ldots, p, i < j\right\}$;
  sample $\boldsymbol{\tau} := \left\{\tau_k \sim \mathcal{G}_{3p}\left(p(p-1)/2, \alpha_{\tau_k}, \beta_{\tau_k}\right)^{-2},\ k = 1, \ldots, K\right\}$, where $\alpha_{\tau_k} = \left(\frac{1}{\zeta_k} + \sum_{i<j} \frac{\left(\omega^k_{ij}\right)^2}{2\lambda^k_{ij}\mu_k}\right)^{1/2}$ and $\beta_{\tau_k} = \sum_{i<j} \frac{\omega^k_{ij}}{\sqrt{\lambda^k_{ij}}\mu_k} \mathbf{T}^{ij}_k$;
  sample $\boldsymbol{\zeta} := \left\{\zeta_k \sim \mathcal{IG}\left(1, 1 + 1/\tau_k\right),\ k = 1, \ldots, K\right\}$;
  sample $\mathbf{R}_\star = \mathrm{diag}\left(\boldsymbol{\Psi}\right)^{-1/2} \boldsymbol{\Psi} \mathrm{diag}\left(\boldsymbol{\Psi}\right)^{-1/2}$, where $\boldsymbol{\Psi} \sim \mathcal{IW}\left(p(p-1)/2, \mathbf{H}\right)$ and $\mathbf{H} := \left\{H_{ij} = \sum_{k=1}^K \left(\sum_{i<j} \omega^k_{ij}\right)^{-1/2} \frac{\omega^k_{ij}}{\sqrt{\tau_k \lambda^k_{ij}}}\right\}$;
  if $\mathcal{U}(0,1) < e^{(K+1)/2(\log|\mathbf{R}_\star| - \log|\mathbf{R}|)}$ **then** set $\mathbf{R} = \mathbf{R}_\star$ and compute $\boldsymbol{\mu} := \left\{\mu_k = 1 - \mathbf{r}_k^\intercal \mathbf{R}^{-1}_{-k}\mathbf{r}_k,\ k = 1, \ldots, K\right\}$;
end
return $\boldsymbol{\Omega}_1 \ldots, \boldsymbol{\Omega}_K, \boldsymbol{\Lambda}_1, \ldots, \boldsymbol{\Lambda}_K, \boldsymbol{\tau}$ and $\mathbf{R}$;

Appendix 3.D Additional details for the Bike-sharing dataset application

Here we report further features of our real data application. We provide an overview of the the thresholds t_k^α estimated with the cut-model (see Table 3.D.1) and the inferred Bikesharing networks for each group with both mGHS and GHS in Figure 3.D.1 and 3.D.2, respectively (black edges denote those edges included in all three years for both member and casual users). For the former problem, we run 4 MCMC chains starting from different values. As shown in Table 3.D.1, the posterior mean is basically equal to the prior mean of t_k^α, therefore suggesting the need of improvements of the cut model: possible solutions are either a new proposal distribution for the MH step or the application of the tempered cut algorithm proposed by Plummer (2015) is required.

	casual 2016	casual 2017	casual 2018	member 2016	member 2017	member 2018
chain 1	0.546	0.548	0.547	0.547	0.545	0.543
chain 2	0.546	0.543	0.542	0.544	0.546	0.546
chain 3	0.547	0.546	0.546	0.544	0.545	0.547
chain 4	0.548	0.547	0.544	0.545	0.547	0.545

Table 3.D.1: Posterior mean of t_k^α for 4 different MCMC chain.

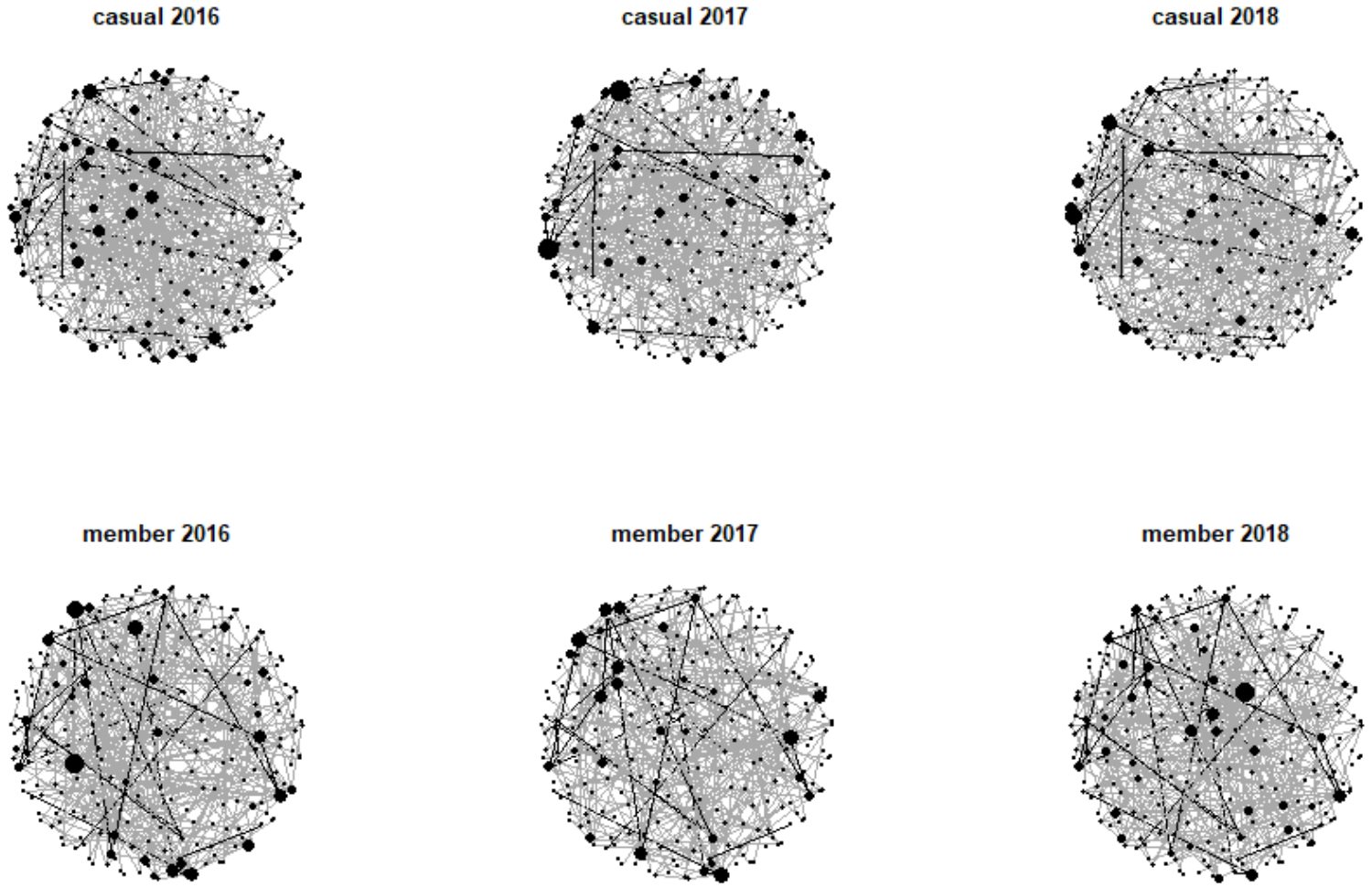


Figure 3.D.1: Estimated graph by mGHS for each group; Black edges denote those edges included in all three years for both member and casual users and the size of the nodes depends on the number of edges associated to the related station.

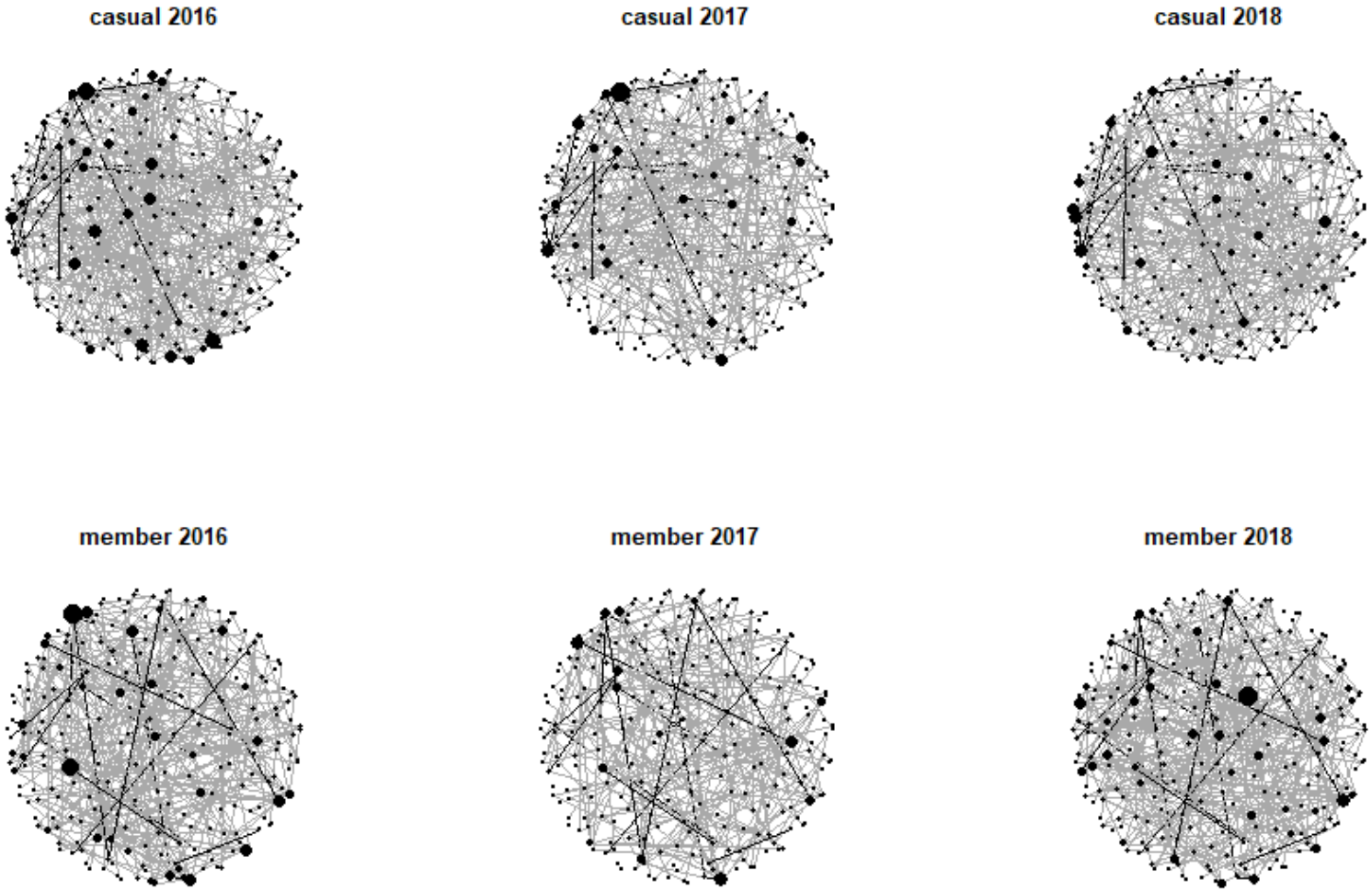


Figure 3.D.2: Estimated graph by GHS for each group; black edges denote those edges included in all three years for both member and casual users and the size of the nodes depends on the number of edges associated to the related station.

Chapter 4

Informative co-data learning for high-dimensional Horseshoe regression

4.1 Introduction

The analysis of high-dimensional data sets is a main interest in many scientific fields. In particular, clinical research often deals with a huge amount of data, such as genes expression or genome-wide methylation levels, for relatively few samples, due to budget or practical constraints. We consider regression models to study a clinical outcome with these data. Since the number of parameters, p, overwhelms sample size, n, we develop an approach to improve the overall performance of the model by incorporating (prior) external knowledge in the estimating process. Such an external source of information is referred to as co-data (complementary data; Neuenschwander et al., 2016), as it provides additional information about the covariates. We consider two different types: continuous, such as p-values from previous studies, or categorical, such as membership to a group, e.g. a chromosome.

Several methods allow incorporating one source of auxiliary information in a regression framework (Tai and Pan, 2007; Boonstra et al., 2013). A popular method is the (sparse) group LASSO (Yuan and Lin, 2006; Simon et al., 2013), which penalizes groups of variables using one common hyperparameter for all groups. Such a solution is attractive when the number of covariate groups is large, but lacks flexibility and fails to adapt locally in other settings, leading to sub-optimal results (Münch et al., 2019). More recent work focuses on the estimation of adaptive penalties with prior variances specific for each group (Van de Wiel et al., 2019; Velten and Huber, 2019; Münch et al., 2019). These methods, however, are re-

strictive in use as they either deal with just one (discrete) co-data source or handle one specific type of outcome only. In Van Nee et al. (2021), instead, a Ridge regression method is proposed that allows for multiple co-data sources by regressing the local variances on the co-data. In this work the co-data regression parameters are estimated independently for each co-data source and are eventually combined using a vector of weights, where each parameter is related to the importance of a single co-data source.

Here, we present a novel Bayesian method for both linear and binary regression that accounts for multiple co-data information. In particular, we introduce a generalization of Horseshoe regression (Carvalho et al., 2010), referred to as the *informative Horseshoe regression* model (infHS). We regress the local variances on the co-data variables, following the work of Van Nee et al. (2021). In contrast to Kpogbezan et al. (2019), where the Horseshoe prior is used with only a single two-group co-data source, our model is flexible with respect to the co-data type, as it allows for both continuous and discrete co-data predictors. Moreover, it extends to binary outcome via probit regression (Albert and Chib, 1993). Unlike Van Nee et al. (2021), it tackles the sparse setting and co-data regression parameters related to different sources are estimated jointly, avoiding multiple regressions for each co-data source separately.

We first propose a Gibbs sampler for iteratively updating posterior parameters. We introduce a novel rejection sampling method for sampling from the non-analytical full-conditional distribution of the local variances. When the number of variables increases, we rely on the computational methods presented in Bhattacharya et al. (2016) to sample efficiently from a multivariate normal density. To make the method applicable to particularly large p settings, we develop a Variational Bayes approximation to the joint posterior distribution, using techniques from Münch et al. (2019) to efficiently optimize the target density of the variational distribution, rendering an algorithm with computation time linear in p. With simulations and two data applications, we show that both prediction and variable selection benefit from the inclusion of co-data information, the latter being particularly relevant under the Horseshoe setting.

The paper in organized as follows. Section 4.2 introduces the hierarchical structure of the model and discusses the parametrizations. In Sections 4.3 and 4.4 we develop the Gibbs sampling algorithm, including the rejection sampling method to update the local variances. In Section 4.5 we propose the Variational Bayes approximation to the joint posterior distribution. Section 4.6 illustrates the benefit of co-data information on variable selection with a simulation study, whereas Section 4.7 presents applications of our model to two data sets, one from genetics and one from cancer genomics. We conclude with discussions and possible extensions in Section 4.8.

4.2 The model

Let $\mathbf{y} \in \mathbb{R}^n$ be the response vector and $\mathbf{X} \in \mathbb{R}^{n\times(p+1)}$ the design matrix, with the first column of ones. We regress $\mathbf{y}$ on $\mathbf{X}$ using a generalized linear model (GLM) with regression coefficient vector $\boldsymbol{\beta} = [\beta_0\ \beta_1 \dots \beta_p]^\intercal$.

$$\begin{aligned} Y_i \mid \mathbf{x}_i, \boldsymbol{\beta} &\overset{ind}{\sim} p\left(Y_i \mid \mathbf{x}_i, \boldsymbol{\beta}\right) \\ E_{Y_i|\mathbf{x}_i,\boldsymbol{\beta}}\left(Y_i\right) &= g^{-1}\left(\mathbf{x}_i^\intercal \boldsymbol{\beta}\right), \quad i = 1, \dots, n, \end{aligned}$$

where $\mathbf{x}_i = [1\ x_{i1} \dots x_{ip}]^\intercal$ is the set of covariates related to observation i.

Following Carvalho et al. (2010), the Horseshoe prior locally shrinks regression parameters toward zero and provides a sparse solution for $\boldsymbol{\beta}$. We assume a normal prior distribution for each β_j, where the variance is decomposed in a global scale parameter τ and a local shrinkage parameter λ_j. Formally,

$$\begin{aligned} \beta_0 \mid \sigma^2, \tau, \lambda_0 &\sim \mathcal{N}\left(0, \sigma^2\tau^2\lambda_0^2\right), \\ \lambda_0 &\sim \mathcal{C}^+(0,1), \\ \beta_j \mid \sigma^2, \tau, \lambda_j &\sim \mathcal{N}\left(0, \sigma^2\tau^2\lambda_j^2\right), \quad j = 1, \dots, p. \end{aligned}$$

Suppose that D different co-data sources $\mathbf{Z}_d \in \mathbb{R}^{p\times m_d}$ are available, where $\sum_d m_d = M$ and $d = 1, \dots, D$. In order to capture the external information effect, Van Nee et al. (2021) introduce parameters $\omega_d > 0$ and $\boldsymbol{\gamma}_d \in \mathbb{R}^{m_d}$, $d = 1, \dots, D$, to model covariate-specific shrinkage λ_j, $j = 1, \dots, p$, as a function of $\mathbf{Z} = [\mathbf{Z}_1, \dots, \mathbf{Z}_D]$. Parameters $\boldsymbol{\gamma}_d$ represent the regression coefficient vector related to matrix $\mathbf{Z}_d$ and are estimated separately for each group d, whereas co-data weights $\omega_d > 0$ model the relative importance of group d and are introduced to combine the different co-data sources. Here, parameter $\boldsymbol{\gamma} = [\boldsymbol{\gamma}_1^\intercal \dots \boldsymbol{\gamma}_D^\intercal]^\intercal$ is update jointly by sampling from its full-conditional distribution. This way the co-data sources are naturally combined and grouping weights ω_d can be excluded from the model. Therefore, the hierarchical set of prior distributions in our model is

$$\begin{aligned} \lambda_j \mid \mathbf{Z}, \boldsymbol{\gamma} &\sim \mathcal{C}\left(\sum_{d=1}^{D} \left(\mathbf{z}_j^d\right)^\intercal \boldsymbol{\gamma}_d, s_0^2\right) \cdot \mathbb{I}_{(\lambda_j>0)}, \quad j = 1, \dots, p, \\ \boldsymbol{\gamma}_d \mid \boldsymbol{\Sigma}_{\gamma_d} &\sim \mathcal{N}_{m_d}\left(\mathbf{0}, \boldsymbol{\Sigma}_{\gamma_d}\right), \quad d = 1, \dots, D, \\ \tau &\sim \mathcal{C}^+\left(0, 1\right), \\ \sigma^2 &\sim \mathcal{IG}\left(v, q\right), \end{aligned} \tag{4.1}$$

where $\mathcal{C}$ denotes the Cauchy distribution and $\mathcal{C}^+$ its half-positive part. Here we model the location parameter of the prior local variances mainly because it allows

to sample co-data coefficients $\boldsymbol{\gamma}$ jointly from a multivariate normal distribution (See Section 4.3). Note that when the co-data are not informative and $\boldsymbol{\gamma}_d \to 0$ for each d, the model reduces to the ordinary Horseshoe prior from Carvalho et al. (2010) with $s_0^2 = 1$.

In Van de Wiel et al. (2019) the authors estimate prior covariance matrices $\boldsymbol{\Sigma}_d$ from the data with an Empirical Bayes estimator separately for each source. This approach, however, is computationally burdensome, as it implies the implementation of multiple MCMC chains until convergence. Here we consider a group-specific scale parameter κ_d^2 and a ridge-like prior $\boldsymbol{\Sigma}_d = \kappa_d^2 \mathbf{I}_{m_d}$. The Zellner's g-prior $\boldsymbol{\Sigma}_{\gamma_d} = c\left(\mathbf{Z}_d^\intercal \mathbf{Z}_d\right)^{-1}$ (Zellner (1986)) would be redundant as the model already accounts for collinearity in $\mathbf{Z}$ (see Section 4.3).

A main advantage of this approach is the computational efficiency, since only prior scale parameters κ_d^2 have to be updated. A conjugated prior distribution for κ_d^2 is

$$\kappa_d^2 \sim \mathcal{IG}\left(a_d, b_d\right). \tag{4.2}$$

Parameters κ_d^2 act deep in the model and one can argue that they should have a small impact on the global estimation process. For this reason, a non-informative choice for a_d and b_d should suffice.

4.3 Posterior inference

In this section we introduce a Gibbs sampler that iteratively updates the parameters by sampling from their full-conditional distributions. We show the details of the algorithm for the linear regression model, under the assumption

$$y_i = \mathbf{x}_i^\intercal \boldsymbol{\beta} + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}\left(0, \sigma^2\right), \quad i = 1, \ldots, n.$$

However, Pólya-Gamma latent variables (Polson et al., 2013) or probit GLM (Albert and Chib, 1993) can be introduced to augment the model and reach a gaussian full-conditional distribution for $\boldsymbol{\beta}$ in a binary regression model.

1. Sampling $\boldsymbol{\beta}$ and σ^2. The full-conditional distributions of $\boldsymbol{\beta}$ and σ^2 are

$$\boldsymbol{\beta} \mid \mathbf{y}, \mathbf{X}, \sigma^2, \tau^2, \lambda_0^2, \boldsymbol{\lambda} \sim \mathcal{N}_{p+1}\left(\boldsymbol{\Sigma}_\beta^\star \mathbf{X}^\intercal \mathbf{y}, \sigma^2 \boldsymbol{\Sigma}_\beta^\star\right), \quad \boldsymbol{\Sigma}_\beta^\star = \left(\mathbf{X}^\intercal \mathbf{X} + \tau^{-2} \boldsymbol{\Lambda}^{-2}\right)^{-1},$$

$$\sigma^2 \mid \mathbf{y}, \mathbf{X}, \boldsymbol{\beta}, \tau^2, \lambda_0^2, \boldsymbol{\lambda} \sim \mathcal{IG}\left(v + \frac{n+p+1}{2}, q + \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \frac{\beta_0^2}{2\tau^2\lambda_0^2} + \frac{1}{2\tau^2}\sum_{j=1}^{p}\frac{\beta_j^2}{\lambda_j^2}\right), \tag{4.3}$$

where $\mathbf{\Lambda} = \text{diag}\,(\lambda_0, \lambda_1, \ldots, \lambda_p)$. Sampling from (4.3) requires the inversion of the $p \times p$ covariance matrix $\mathbf{\Sigma}^{\star}_{\beta}$, which becomes computationally infeasible when the number of covariates p increases (the naive inversion is of order $O\,(p^3)$). For high-dimensional problems we rely on the strategies introduced in Bhattacharya et al. (2016) and Johndrow et al. (2020) to reduce the computational costs of sampling regression parameters $\boldsymbol{\beta}$ to $O\,(n^2 p)$ operations.

2. Sampling λ_0^2**.** The half-Cauchy prior for the local shrinkage parameter λ_0 of the intercept β_0 is not conjugated to the variance in a linear regression model with normal errors. We rely on the data-augmentation step proposed in Makalic and Schmidt (2016) in order to easily and efficiently update parameter λ_0. The authors point out that the half-Cauchy distribution can be written as a scale mixture of inverse-Gamma distributions, which allows conjugate updates of λ_0^2. Therefore, the prior distribution can be rewritten as

$$\begin{aligned} \lambda_0^2 \mid \psi_0 &\sim \mathcal{IG}\left(\frac{1}{2}, \frac{1}{\psi_0}\right), \\ \psi_0 &\sim \mathcal{IG}\left(\frac{1}{2}, 1\right). \end{aligned}$$

The inverse-Gamma distribution is conjugated to itself and to the local scale parameter, therefore a closed-form full-conditional is available and a Gibbs step can be implemented. The full-conditional distributions of λ_0^2 and ψ_0 are

$$\begin{aligned} \lambda_0^2 \mid \beta_0, \tau^2, \sigma^2 &\sim \mathcal{IG}\left(1, \frac{1}{\psi_0} + \frac{\beta_0^2}{2\sigma^2\tau^2}\right), \\ \psi_0 \mid \lambda_0^2 &\sim \mathcal{IG}\left(1, 1 + \frac{1}{\lambda_0^2}\right). \end{aligned}$$

3. Sampling $\boldsymbol{\lambda}$**,** $\boldsymbol{\gamma}$ ***and*** $\boldsymbol{\kappa}^2$**.** The prior distributions for λ_j and $\boldsymbol{\gamma}$ in (4.1) are not conjugated. To this aim, we rely on the data-augmentation step proposed in Geweke (1993) to reach a conjugated framework and jointly update parameter $\boldsymbol{\gamma}$ by sampling from a multivariate normal distribution. Note that the distributions $\mathcal{C}\,(m, s^2)$ and $\mathtt{t}_v\,(m, s^2)$ are equivalent if $v = 1$, where t_v denotes the Student-t distribution. At this point we can rely on the result from Geweke (1993), which states that a Student-t distribution can be formulated as a mixture of Normal and Inverse-Gamma distributions. Formally, let

$$\lambda_j \mid \mathbf{Z}, \boldsymbol{\gamma}, \varphi_j^2 \sim \mathcal{N}\left(\sum_{d=1}^{D} \left(\mathbf{z}_j^d\right)^{\intercal} \boldsymbol{\gamma}_d, s_0^2 \varphi_j^2\right) \cdot \mathbb{I}_{(\lambda_j > 0)} \quad \text{and} \quad \varphi_j^2 \sim \mathcal{IG}\left(\frac{1}{2}, \frac{1}{2}\right), \quad \text{for} \quad j = 1, \ldots, p,$$

then $\lambda_j \mid \mathbf{Z}, \boldsymbol{\gamma} \sim \mathtt{t}_1\left(\sum_{d=1}^{D}\left(\mathbf{z}_j^d\right)^\intercal \boldsymbol{\gamma}_d, s_0^2\right)$. Therefore, after the introduction of a latent factor $\varphi_j^2 \sim \mathcal{IG}(1/2, 1/2)$, the prior distribution of λ_j can be conveniently re-written as a normal distribution truncated at 0. This way the normal prior for co-data regression coefficients $\boldsymbol{\gamma}$ is conjugated and the parameters can easily be updated by sampling from a multivariate normal distribution. The set of prior distributions for parameters λ_j, φ_j^2, $\boldsymbol{\gamma}_d$ can be re-written as

$$\begin{aligned}
\lambda_j \mid \mathbf{Z}, \boldsymbol{\gamma}, \varphi_j^2 &\sim \mathcal{N}\left(\sum_{d=1}^{D}\left(\mathbf{z}_j^d\right)^\intercal \boldsymbol{\gamma}_d, s_0^2\varphi_j^2\right) \cdot \mathbb{I}_{(\lambda_j>0)}, \\
\varphi_j^2 &\sim \mathcal{IG}\left(\frac{1}{2}, \frac{1}{2}\right), \quad j = 1, \ldots, p, \\
\boldsymbol{\gamma}_d \mid \kappa_d^2 &\sim \mathcal{N}_{m_d}\left(\mathbf{0}, \kappa_d^2 \mathbf{I}_{m_d}\right), \\
\kappa_d^2 &\sim \mathcal{IG}\left(a_d, b_d\right), \quad d = 1, \ldots, D.
\end{aligned}$$

Let $\mu_j = \sum_{d=1}^{D}\left(\mathbf{z}_j^d\right)^\intercal \boldsymbol{\gamma}_d$, $\boldsymbol{\lambda} = [\lambda_1 \ldots \lambda_p]^\intercal$ and $\boldsymbol{\Phi}^2 = \operatorname{diag}(\boldsymbol{\varphi})$, with $\boldsymbol{\varphi} = \left[\varphi_1^2, \ldots, \varphi_p^2\right]^\intercal$. The full-conditional distributions of λ_0, λ_j, $\boldsymbol{\gamma}$, φ_j^2 and κ_d^2 in the augmented model are

$$\begin{aligned}
\pi\left(\lambda_j \mid \mathbf{Z}, \beta_j, \sigma^2, \tau^2, \boldsymbol{\gamma}, \varphi_j^2\right) &\propto \lambda_j^{-1} e^{-\frac{\beta_j^2}{2\sigma^2\tau^2\lambda_j^2} - \frac{\lambda_j^2}{2s_0^2\varphi_j^2} + \frac{\mu_j\lambda_j}{s_0^2\varphi_j^2}} \cdot \mathbb{I}_{(\lambda_j>0)}, \\
\varphi_j^2 \mid \mathbf{Z}, \lambda_j, \boldsymbol{\gamma} &\sim \mathcal{IG}\left(1, \frac{1}{2} + \frac{(\lambda_j - \mu_j)^2}{2s_0^2}\right), \\
\boldsymbol{\gamma} \mid \mathbf{Z}, \boldsymbol{\lambda}, \boldsymbol{\varphi}, \boldsymbol{\kappa}^2 &\sim \mathcal{N}_M\left(\boldsymbol{\Sigma}_\gamma^\star\left(\mathbf{Z}^\intercal\boldsymbol{\Phi}^{-2}\boldsymbol{\lambda}\right), s_0^2\boldsymbol{\Sigma}_\gamma^\star\right), \\
\kappa_d^2 \mid \boldsymbol{\gamma}_d &\sim \mathcal{IG}\left(a_d + \frac{m_d}{2}, b_d + \frac{\boldsymbol{\gamma}_d^\intercal\boldsymbol{\gamma}_d}{2}\right),
\end{aligned} \tag{4.4}$$

where $\boldsymbol{\Sigma}_\gamma^\star = \left(\mathbf{Z}^\intercal\boldsymbol{\Phi}^{-2}\mathbf{Z} + s_0^2\mathbf{D}_\kappa^{-1}\right)^{-1}$ and $\mathbf{D}_\kappa = \operatorname{diag}(\kappa_1^2\mathbf{1}_{m_1}, \ldots, \kappa_D^2\mathbf{1}_{m_D})$. Details for sampling parameters λ_j without computing the unknown normalizing constant are given in Section 4.4.

The introduced framework presents a computational advantage: the local variances λ_j^2 can be computed in parallel, potentially improving the efficiency of the model.

4. Sampling τ^2. As for the scale parameter λ_0^2, the half-Cauchy prior for the global scale parameter τ is not conjugated to the prior variance of $\boldsymbol{\beta}$. Therefore, we rely on the strategy proposed by Makalic and Schmidt (2016) in order to update

τ and we assume the prior distributions

$$\tau^2 \mid \zeta \sim \mathcal{IG}\left(\frac{1}{2}, \frac{1}{\zeta}\right),$$
$$\zeta \sim \mathcal{IG}\left(\frac{1}{2}, 1\right)$$

The full-conditional distributions for τ^2 and ζ are available in closed-form. In particular the parameters are updated by sampling from the following densities:

$$\tau^2 \mid \boldsymbol{\beta}, \sigma^2, \lambda_0^2, \boldsymbol{\lambda}, \zeta \sim \mathcal{IG}\left(\frac{p}{2} + 1, \frac{1}{\zeta} + \frac{\beta_0^2}{2\sigma^2\lambda_0^2} + \frac{1}{2\sigma^2}\sum_{j=1}^{p}\frac{\beta_j^2}{\lambda_j^2}\right),$$
$$\zeta \mid \tau^2 \sim \mathcal{IG}\left(1, 1 + \frac{1}{\tau^2}\right).$$

4.4 Rejection sampling for parameters λ_j

In this section we propose a novel rejection sampling algorithm to sample shrinkage parameters λ_j, $j = 1, \ldots, p$, from the full-conditional distribution without knowing the normalizing constant. The rejection sampling allows to draw a new value $x_\star$ from a density f by sampling it from a proposal distribution g and accepting it with a probability proportional to the ratio $r(x_\star) = f(x_\star)/g(x_\star)$. In particular, g must be chosen such that the support of f is a subset of the support of g. Let

$$k = \sup_x \frac{f(x)}{g(x)} < \infty$$

and accept $x_\star$ with probability $a = f(x_\star)/(kg(x_\star))$. It can be shown that the acceptance probability of the algorithm is $1/k$. Therefore, the goal is to find a proposal distribution g such that k is small. The rejection sampling works also if the normalizing constant of f is unknown, as long as a sampling method for g is available.

Consider the following density

$$f(x) = c_f x^{-1} e^{-\psi/x^2 - \alpha^2 x^2 + \beta x} \cdot \mathbb{I}_{(x>0)},$$

where $\psi, \alpha^2 > 0$, $\beta \in \mathbb{R}$ and c_f is the unknown normalizing constant. New values are sampled from the proposal distribution $g(x) \sim \mathcal{G}_{3p}(\gamma, \alpha, \beta)$, with density $g(x) = c_g x^\gamma e^{-\alpha^2 x^2 + \beta x}$, where $\gamma \in \mathbb{N}^+$. We set parameters α and β equal in both

Algorithm 13: Gibbs sampler for Informative Horseshoe regression

Input: $B, bn \in \mathbb{N}$, $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{X} \in \mathbb{R}^{n\times(p+1)}$, $\mathbf{Z}_1, \ldots, \mathbf{Z}_D \in \mathbb{R}^{p\times m_d}$, $\mathbf{a}, \mathbf{b} \in \mathbb{R}_+^D$, $s_0^2 \in \mathbb{R}_+$;

// Set $\lambda_0 = 1$, $\boldsymbol{\lambda} = \mathbf{1}_p$, $\boldsymbol{\gamma} = \mathbf{0}_M$ and $\sigma^2, \tau^2 = 1$ and sample parameters $\boldsymbol{\varphi}$, $\boldsymbol{\kappa}^2$ and ζ from their prior distributions

for $b = 1$ *to* B **do**

  sample $\boldsymbol{\beta} \mid \mathbf{y}, \mathbf{X}, \sigma^2, \tau^2, \lambda_0^2, \boldsymbol{\lambda} \sim \mathcal{N}_{p+1}\left(\boldsymbol{\Sigma}_\beta^\star \mathbf{X}^\intercal \mathbf{y}, \sigma^2 \boldsymbol{\Sigma}_\beta^\star\right)$, where $\boldsymbol{\Sigma}_\beta^\star = \left(\mathbf{X}^\intercal \mathbf{X} + \tau^{-2}\boldsymbol{\Lambda}^{-2}\right)^{-1}$;

  sample $\lambda_0^2 \mid \beta_0, \tau^2, \sigma^2 \sim \mathcal{IG}\left(1, \frac{1}{\psi_0} + \frac{\beta_0^2}{2\sigma^2\tau^2}\right)$;

  sample $\psi_0 \mid \lambda_0^2 \sim \mathcal{IG}\left(1, 1 + \frac{1}{\lambda_0^2}\right)$;

  for $j = 1$ *to* p **do**

    sample $\lambda_j \mid \mathbf{Z}, \beta_j, \sigma^2, \tau^2, \boldsymbol{\gamma}, \varphi_j^2 \propto \lambda_j^{-1} e^{-\frac{\beta_j^2}{2\sigma^2\tau^2\lambda_j^2} - \frac{\lambda_j^2}{2s_0^2\varphi_j^2} + \frac{\mu_j\lambda_j}{s_0^2\varphi_j^2}} \cdot \mathbb{I}_{(\lambda_j>0)}$ following the procedure in Section 4.4;

    sample $\varphi_j^2 \mid \mathbf{Z}, \lambda_j, \boldsymbol{\gamma} \sim \mathcal{IG}\left(1, \frac{1}{2} + \frac{(\lambda_j-\mu_j)^2}{2s_0^2}\right)$;

  end

  sample $\boldsymbol{\gamma} \mid \mathbf{Z}, \boldsymbol{\lambda}, \boldsymbol{\varphi}, \boldsymbol{\kappa}^2 \sim \mathcal{N}_M\left(\boldsymbol{\Sigma}_\gamma^\star \left(\mathbf{Z}^\intercal \boldsymbol{\Phi}^{-2}\boldsymbol{\lambda}\right), s_0^2\boldsymbol{\Sigma}_\gamma^\star\right)$, where $\boldsymbol{\Sigma}_\gamma^\star = \left(\mathbf{Z}^\intercal\boldsymbol{\Phi}^{-2}\mathbf{Z} + s_0^2\mathbf{D}_\kappa^{-1}\right)^{-1}$ and $\mathbf{D}_\kappa = \mathrm{diag}\left(\kappa_1^2\mathbf{1}_{m_1}, \ldots, \kappa_D^2\mathbf{1}_{m_D}\right)$;

  for $d = 1$ *to* D **do**

    sample $\kappa_d^2 \mid \boldsymbol{\gamma}_d \sim \mathcal{IG}\left(a_d + \frac{m_d}{2}, b_d + \frac{\boldsymbol{\gamma}_d^\intercal\boldsymbol{\gamma}_d}{2}\right)$;

  end

  sample $\tau^2 \mid \boldsymbol{\beta}, \sigma^2, \lambda_0^2, \boldsymbol{\lambda}, \zeta \sim \mathcal{IG}\left(\frac{p}{2} + 1, \frac{1}{\zeta} + \frac{\beta_0^2}{2\sigma^2\lambda_0^2} + \frac{1}{2\sigma^2}\sum_{j=1}^p \frac{\beta_j^2}{\lambda_j^2}\right)$;

  sample $\zeta \mid \tau^2 \sim \mathcal{IG}\left(1, 1 + \frac{1}{\tau^2}\right)$;

  sample $\sigma^2 \mid \mathbf{y}, \mathbf{X}, \boldsymbol{\beta}, \tau^2, \lambda_0^2, \boldsymbol{\lambda} \sim \mathcal{IG}\left(v + \frac{n+p+1}{2}, q + \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \frac{\beta_0^2}{2\tau^2\lambda_0^2} + \frac{1}{2\tau^2}\sum_{j=1}^p \frac{\beta_j^2}{\lambda_j^2}\right)$;

  if $b > bn$ **then**

    save $\boldsymbol{\beta}$, σ^2, τ^2, λ_0^2, $\boldsymbol{\lambda}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\kappa}^2$

  end

end

return saved values;

f and g in order for the ratio $r(x)$ to be analytically tractable and easily maximized. Eventually, the acceptance probability is optimized through the choice of parameter γ. This choice of α and β yields

$$r(x) = \frac{c_f}{c_g} x^{-\gamma-1} e^{-\psi/x^2} \cdot \mathbb{I}_{(x>0)},$$

which has one positive maximum at $\dot{x} = \sqrt{2\psi/(\gamma+1)}$. Details for sampling from a $\mathcal{G}_{3p}$ distribution are shown in Section 3.3.

The best theoretical way to choose the parameter γ is to differentiate $k = r(\dot{x})c_f/c_g$ with respect to γ and find the optimal value. However, this can only be done with iterative methods, which negatively affects the computational efficiency of the method. An alternative solution is to set parameter γ such that distributions f and g have the maximum at the same value x_{max}. That is, the maximum of f is computed by solving a quartic equation and the parameter γ is estimated as

$$\gamma = x_{max}\left(2\alpha^2 x_{max} - \beta\right) > -1.$$

Since $\gamma \in \mathbb{N}_0^+$, the closest non-negative integer is chosen. An example of the method is shown in Figure 4.4.1, where the normalizing constant c_f is computed by numerical integration.

For some settings of the parameters ψ, α and β the acceptance probability of the proposed algorithm decreases toward 0, affecting the efficiency of the algorithm. For this reason, and for avoiding the explicit inversion of covariance matrix $\boldsymbol{\Sigma}_\beta^\star$, in the next section we present a Variational Bayes algorithm which overcomes these problems.

4.5 Variational Bayes approximation

When the number of covariates is huge the method introduced in Section 4.2 becomes computationally infeasible. In this section an efficient approximation of the joint posterior distribution is discussed.

Variational inference (VI) is a deterministic optimization method to approximate the target density $\pi(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{X}, \mathbf{Z})$, with $\boldsymbol{\theta} = (\boldsymbol{\beta}, \lambda_0^2, \boldsymbol{\lambda}, \psi_0, \boldsymbol{\varphi}, \boldsymbol{\gamma}, \boldsymbol{\kappa}, \tau^2, \zeta, \sigma^2)$, with another variational distribution $q(\boldsymbol{\theta})$ and reduces Bayesian inference to an optimization problem (Salimans et al., 2015; Lee, 2022). The goal is to find $q(\boldsymbol{\theta})$ that minimizes the Kullback-Leibler divergence (KL) between the target density and the variational distribution. The minimization problem in is eventually reduced to the maximization of the variational lower bound, defined as $\mathcal{L} = \mathbb{E}_q[\log \pi(\boldsymbol{\theta}, \mathbf{y}, \mathbf{X}, \mathbf{Z})] - \mathbb{E}_q[\log q(\boldsymbol{\theta})]$.

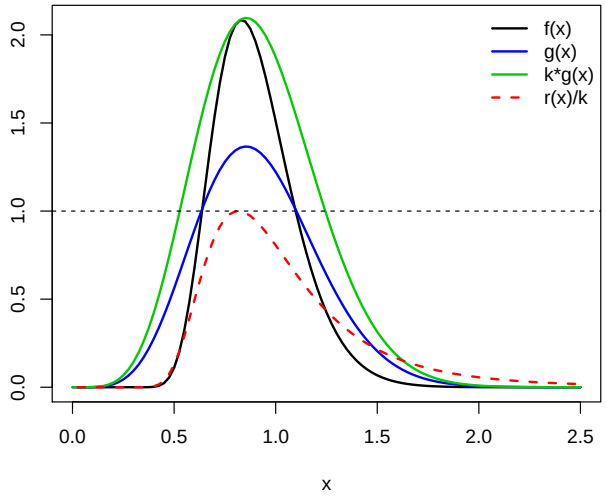


Figure 4.4.1: comparison of $f(x)$ and $g(x)$ with $\psi = 2$, $\alpha^2 = 2.25$ and $\beta = -2$. Estimated parameter $\gamma = 5$; the acceptance probability is $a = 0.65$.

A common factorization for $q(\boldsymbol{\theta})$ is the so-called *mean-field Variational approximation* (Jordan et al., 1999; Beal, 2003), which is a compromise between computational tractability and accuracy of the performances. The variational family $q(\boldsymbol{\theta})$ is assumed to be the product of independent marginal variational factors $q_k(\theta_k)$, $k = 1, \ldots, K$. We rely on the *Coordinate Ascent Variational Inference* algorithm (CAVI) (Bishop and Nasrabadi, 2006; Blei et al., 2017) to solve the optimization problem above. Until convergence of the lower bound $\mathcal{L}$, the CAVI algorithm iteratively updates the parameters of the variational factors $q_k(\theta_k)$, $k = 1, \ldots, K$, based on prior distributions' hyperparameters and the current expectation of factor $q_{-k}(\theta_{-k})$, considered fixed (Lee, 2022). This way the model is able to account for non-linear dependencies among the parameters. Under the mean field approximation, where the components are assumed to be independent, the optimal solution is given by

$$q^{\star}(\theta_k) \propto \exp\left\{\mathbb{E}_{q_{-k}}\left[\log \pi\left(\theta_k \mid \theta_{-k}, \mathbf{y}, \mathbf{X}, \mathbf{Z}\right)\right]\right\}.$$

While the assumption of independence between factors is particularly strict, the CAVI algorithm provides a flexible approach and is ensured to converge to a local optimum (Blei et al., 2017). Note that, when working with exponential families in a conjugated framework, variational factor $q(\theta_k)$ has the same kernel of the tractable distribution $\pi\left(\theta_k \mid \theta_{-k}, \mathbf{y}, \mathbf{X}, \mathbf{Z}\right)$.

Under the assumptions of the model introduced in Section 4.2, the mean field approximation yields

$$q(\boldsymbol{\theta}) = q(\boldsymbol{\beta}) \cdot q\left(\lambda_0^2\right) \cdot q(\lambda_1) \cdot \ldots \cdot q(\lambda_p) \cdot q(\psi_0) \cdot q\left(\varphi_1^2\right) \cdot \ldots \cdot q\left(\varphi_p^2\right) \cdot q(\boldsymbol{\gamma}) \cdot q\left(\kappa_1^2\right) \cdot \ldots \cdot q\left(\kappa_D^2\right) \cdot q\left(\tau^2\right) \cdot q(\zeta) \cdot q\left(\sigma^2\right),$$

where $q(\boldsymbol{\beta})$ and $q(\boldsymbol{\gamma})$ denote the joint variational distribution of $\beta_0, \ldots, \beta_p$ and $\gamma_1, \ldots, \gamma_M$, respectively. At each iteration of the algorithm, the variational factors are updated as

$$
\begin{aligned}
q^\star(\boldsymbol{\beta}) &= \mathcal{N}_{p+1}\left(\boldsymbol{\mu}_\beta^\star, \mathbb{E}_{\sigma^2}\left[\sigma^2\right]\boldsymbol{\Sigma}_\beta^\star\right), \\
&\quad \boldsymbol{\mu}_\beta^\star = \boldsymbol{\Sigma}_\beta^\star \mathbf{X}^\intercal \mathbf{y}, \quad \boldsymbol{\Sigma}_\beta^\star = \left(\mathbf{X}^\intercal\mathbf{X} + \mathbb{E}_{\lambda_0^2\cdot\boldsymbol{\lambda}\cdot\tau^2}\left[\tau^{-2}\boldsymbol{\Lambda}^{-2}\right]\right)^{-1}, \\
q^\star\left(\lambda_0^2\right) &= \mathcal{IG}\left(1, a_0^\star\right), \\
&\quad a_0^\star = \mathbb{E}_\zeta\left[\psi_0^{-1}\right] + \frac{1}{2}\mathbb{E}_{\beta_0\cdot\sigma^2\cdot\tau^2}\left[\frac{\beta_0^2}{\sigma^2\tau^2}\right], \\
q^\star(\psi_0) &= \mathcal{IG}\left(1, k_0^\star\right), \\
&\quad k_0^\star = 1 + \mathbb{E}_{\lambda_0^2}\left[\lambda_0^{-2}\right], \\
q^\star(\lambda_j) &\propto \lambda_j^{-1}\exp\left\{-\frac{a_j^\star}{\lambda_j^2} - b_j^\star\lambda_j^2 + c_j^\star\lambda_j\right\}\cdot\mathbb{I}_{(\lambda_j>0)}, \quad j = 1, \ldots, p, \\
&\quad a_j^\star = \frac{1}{2}\mathbb{E}_{\beta\cdot\sigma^2\cdot\tau^2}\left[\frac{\beta_j^2}{\sigma^2\tau^2}\right], \quad b_j^\star = \frac{1}{2s_0^2}\mathbb{E}_{\varphi^2}\left[\varphi_j^{-2}\right], \quad c_j^\star = \frac{1}{s_0^2}\mathbf{z}_j^\intercal\mathbb{E}_{\gamma\cdot\varphi^2}\left[\frac{\boldsymbol{\gamma}}{\varphi_j^2}\right], \\
q^\star\left(\varphi_j^2\right) &= \mathcal{IG}\left(1, d_j^\star\right), \\
&\quad d_j^\star = \frac{1}{2} + \frac{1}{2s_0^2}\mathbb{E}_{\lambda\cdot\gamma}\left[\left(\lambda_j - \mathbf{z}_j^\intercal\boldsymbol{\gamma}\right)^2\right], \\
q^\star(\boldsymbol{\gamma}) &= \mathcal{N}_M\left(\boldsymbol{\mu}_\gamma^\star, s_0^2\boldsymbol{\Sigma}_\gamma^\star\right), \\
&\quad \boldsymbol{\mu}_\gamma^\star = \boldsymbol{\Sigma}_\gamma^\star\mathbf{Z}^\intercal\mathbb{E}_{\varphi^2\cdot\gamma}\left[\boldsymbol{\Phi}^{-2}\boldsymbol{\lambda}\right], \quad \boldsymbol{\Sigma}_\gamma^\star = \left(\mathbf{Z}^\intercal\mathbb{E}_{\varphi^2}\left[\boldsymbol{\Phi}^{-2}\right]\mathbf{Z} + s_0^2\mathbb{E}_{\kappa^2}\left[\mathbf{D}_\kappa^{-1}\right]\right)^{-1}, \\
q^\star\left(\kappa_d^2\right) &= \mathcal{IG}\left(e_d^\star, f_d^\star\right), \quad d = 1, \ldots, D, \\
&\quad e_d^\star = a_d + \frac{m_d}{2}, \quad f_d^\star = b_d + \frac{1}{2}\mathbb{E}_\gamma\left[\boldsymbol{\gamma}_d^\intercal\boldsymbol{\gamma}_d\right], \\
q^\star\left(\tau^2\right) &= \mathcal{IG}\left(\frac{p}{2} + 1, g^\star\right), \\
&\quad g^\star = \mathbb{E}_\zeta\left[\zeta^{-1}\right] + \frac{1}{2}\mathbb{E}_{\beta_0\cdot\lambda_0^2\cdot\tau^2}\left[\frac{\beta_0^2}{\tau^2\lambda_0^2}\right] + \frac{1}{2}\sum_{j=1}^p\mathbb{E}_{\beta\cdot\sigma^2\cdot\boldsymbol{\lambda}}\left[\frac{\beta_j^2}{\sigma^2\lambda_j^2}\right], \\
q^\star(\zeta) &= \mathcal{IG}\left(1, h^\star\right), \\
&\quad h^\star = 1 + \mathbb{E}_{\tau^2}\left[\tau^{-2}\right], \\
q^\star\left(\sigma^2\right) &= \mathcal{IG}\left(v + \frac{n+p+1}{2}, l^\star\right), \\
&\quad l^\star = q + \frac{1}{2}\left(\mathbb{E}_\beta\left[\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2\right] + \mathbb{E}_{\beta_0\cdot\lambda_0^2\cdot\tau^2}\left[\frac{\beta_0^2}{\tau^2\lambda_0^2}\right] + \sum_{j=1}^p\mathbb{E}_{\beta\cdot\boldsymbol{\lambda}\cdot\tau^2}\left[\frac{\beta_j^2}{\tau^2\lambda_j^2}\right]\right),
\end{aligned}
$$

where the expectations are taken with respect to the updated variational family $q^\star(\boldsymbol{\theta})$. The variational lower bound $\mathcal{L}$ of the marginal distribution $p(\mathbf{y})$ is computed as

$$
\begin{aligned}
\mathcal{L} &= \mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\theta}, \mathbf{y}, \mathbf{X}, \mathbf{Z}\right)\right] - \mathbb{E}_{q^\star}\left[\log q^\star\left(\boldsymbol{\theta}\right)\right] \\
&= \mathbb{E}_{q^\star}\left[\log p\left(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}, \sigma^2\right)\right] + \mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\beta} \mid \sigma^2, \tau^2, \lambda_0^2, \boldsymbol{\lambda}\right)\right] + \mathbb{E}_{q^\star}\left[\log \pi\left(\lambda_0^2 \mid \psi_0\right)\right] + \\
&\quad \mathbb{E}_{q^\star}\left[\log \pi\left(\psi_0\right)\right] + \mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\lambda} \mid \mathbf{Z}, \boldsymbol{\gamma}, \boldsymbol{\varphi}^2\right)\right] + \mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\varphi}^2\right)\right] + \mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\gamma} \mid \boldsymbol{\kappa}^2\right)\right] + \\
&\quad \mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\kappa}^2\right)\right] + \mathbb{E}_{q^\star}\left[\log \pi\left(\tau^2 \mid \zeta\right)\right] + \mathbb{E}_{q^\star}\left[\log \pi\left(\zeta\right)\right] + \mathbb{E}_{q^\star}\left[\log \pi\left(\sigma^2\right)\right] - \\
&\quad \mathbb{E}_{q^\star}\left[\log q^\star\left(\boldsymbol{\beta}\right)\right] - \mathbb{E}_{q^\star}\left[\log q^\star\left(\lambda_0^2\right)\right] - \mathbb{E}_{q^\star}\left[\log q^\star\left(\psi_0\right)\right] - \mathbb{E}_{q^\star}\left[\log q^\star\left(\boldsymbol{\lambda}\right)\right] - \mathbb{E}_{q^\star}\left[\log q^\star\left(\boldsymbol{\varphi}^2\right)\right] - \\
&\quad \mathbb{E}_{q^\star}\left[\log q^\star\left(\boldsymbol{\gamma}\right)\right] - \mathbb{E}_{q^\star}\left[\log q^\star\left(\boldsymbol{\kappa}^2\right)\right] - \mathbb{E}_{q^\star}\left[\log q^\star\left(\tau^2\right)\right] - \mathbb{E}_{q^\star}\left[\log q^\star\left(\zeta\right)\right] - \mathbb{E}_{q^\star}\left[\log q^\star\left(\sigma^2\right)\right] \\
&\propto \frac{1}{2}\left(\log|\boldsymbol{\Sigma}_\beta^\star| + \log|\boldsymbol{\Sigma}_\gamma^\star|\right) + \frac{p+1}{2}\mathbb{E}_{\sigma^2}\left[\log \sigma^2\right] - \left(v + \frac{n+p+1}{2}\right)\log l^\star + \\
&\quad \sum_{j=1}^{p}\left(\log s_j - \log k_j + a_j^\star \mathbb{E}_\lambda\left[\lambda_j^{-2}\right] + b_j^\star \mathbb{E}_\lambda\left[\lambda_j^2\right] - c_j^\star \mathbb{E}_\lambda\left[\lambda_j\right] - \log d_j^\star\right) - \\
&\quad \sum_{d=1}^{D} e_d^\star \log f_d^\star - \left(\frac{p}{2}+1\right)\log g^\star - \log h^\star - \log a_0^\star - \log k_0^\star, \qquad (4.5)
\end{aligned}
$$

where $k_j = 1 - \mathbb{P}\left(\mathcal{N}\left(\mathbf{z}_j^\intercal \boldsymbol{\gamma}, s_0^2\varphi_j^2\right) < 0\right)$ and s_j is the unknown normalizing constant of parameters λ_j in (4.4). Each component is derived in Appendix 4.A.1.

As opposed to the Gibbs sampler, the Variational algorithm does not require the explicit inversion of the $p \times p$ matrix $\boldsymbol{\Sigma}_\beta^\star$, as only the quantities $\boldsymbol{\Sigma}_\beta^\star \mathbf{X}^\intercal \mathbf{y}$, $\operatorname{diag}\left(\boldsymbol{\Sigma}_\beta^\star\right)$ and $\mathbf{X}\boldsymbol{\Sigma}_\beta^\star\mathbf{X}^\intercal$ are needed. These can be efficiently evaluated with complexity $O\left(n^2 p\right)$ following the work of Münch et al. (2019). The details are shown in Appendix 4.B. The bottleneck of the algorithm is the computation of the terms $\log s_j$, $\mathbb{E}_\lambda\left[\lambda_j\right]$, $\mathbb{E}_\lambda\left[\lambda_j^2\right]$ and $\mathbb{E}_\lambda\left[\lambda_j^{-2}\right]$, since no closed-form is available. These can be evaluated by numerical integration, for example with the Gaussian quadrature rule or its adaptive variation called Gauss-Kronrod quadrature formula. Depending on the posterior parameters $a_j^\star$, $b_j^\star$ and $c_j^\star$, this strategy is prone to numerical instability. To avoid overflow problem, we evaluate these integrals with the following step:

Step 0: assume we have to evaluate $\int_0^\infty f(x)dx$, where $f(x) \propto x^\nu \exp\left\{-dx^{-2} - bx^2 + cx\right\} \cdot \mathbb{I}_{(\lambda_j > 0)}$, where $\nu = \{-3, -1, 0, 1\}$;

Step 1: compute the maximum $\dot{x}$ by solving a quartic equation and evaluate $f(\dot{x})$ on the log-scale;

Step 2: evaluate $i_0 = \int_0^\infty \exp\left\{\log f(x) - \log f(\dot{x})\right\} dx$ numerically;

Algorithm 14: Variational Bayes approximation for informative Horseshoe regression

Input: $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{X} \in \mathbb{R}^{n \times (p+1)}$, $\mathbf{Z}_1, \dots, \mathbf{Z}_D \in \mathbb{R}^{p \times m_d}$, $v, q \in \mathbb{R}^+$, $\mathbf{a}, \mathbf{b} \in \mathbb{R}_+^D$, $s_0^2 \in \mathbb{R}_+$;
// Set $b = 1$, $\epsilon = 10^{-3}$, $\mathcal{L}^{(0)} = -\infty$ and initialize all the needed moments
while $\mathcal{L}^{(b)} - \mathcal{L}^{(b-1)} > \epsilon$ **do**
  Update parameter $\boldsymbol{\mu}_\beta^\star$ and compute the quantities diag $\left(\boldsymbol{\Sigma}_\beta^\star\right)$, $|\boldsymbol{\Sigma}_\beta^\star|$ and $\mathbf{X}\boldsymbol{\Sigma}_\beta^\star\mathbf{X}^\intercal$ as in Appendix 4.B;
  Update parameters $a_0^\star$ and $k_0^\star$;
  Update parameters $a_j^\star$, $b_j^\star$, $c_j^\star$ and $d_j^\star$ and evaluate the normalizing constant s_j, $\mathbb{E}_\lambda[\lambda_j]$, $\mathbb{E}_\lambda\left[\lambda_j^2\right]$ and $\mathbb{E}_\lambda\left[\lambda_j^{-2}\right]$ with numerical integration, for $j = 1, \dots, p$;
  Update parameters $\boldsymbol{\mu}_\gamma^\star$ and $\boldsymbol{\Sigma}_\gamma^\star$;
  Update parameters $e_d^\star$ and $f_d^\star$, for $d = 1, \dots, D$;
  Update parameters $g^\star$ and $h^\star$;
  Update parameter $l^\star$;
  Compute $\mathcal{L}^{(b)}$ and set $b = b + 1$;
end
return $\boldsymbol{\mu}_\beta^\star$, $\boldsymbol{\Sigma}_\beta^\star$, $a_0^\star$, $\mathbf{a}^\star$, $\mathbf{b}^\star$, $\mathbf{c}^\star$, $\boldsymbol{\mu}_\gamma^\star$, $\boldsymbol{\Sigma}_\gamma^\star$, $\mathbf{e}^\star$, $\mathbf{f}^\star$, $g^\star$ and $l^\star$;

Step 3: return $\exp\{\log i_0 + \log f(\dot{x})\}$.

As for the Gibbs sampler in Section 4.3, the efficiency of the model can be improved by evaluating these quantities in parallel.

4.6 Simulation study

In this section we empirically show the quality of the Variational approximation to the joint posterior distribution and that variable selection benefits from the co-data sources with a model-based simulation study. In particular, we assess the effectiveness of the approximation on low to moderate p problems, whereas variable selection is evaluated with the Variational algorithm on higher dimensional frameworks.

For all the considered cases we rely on the following simulation scheme. Let $\boldsymbol{\beta}^0$ be the true $(p+1)$-dimensional regression parameter vector. We set the number of true non-zero coefficients to p_0 (intercept excluded). The entries of design matrices

are sampled independently as $x_{ij} \sim \mathcal{N}(0,1)$, $i = 1, \ldots, n$, $j = 1, \ldots, p$, whereas the response variable $\mathbf{y}$ and the $(p+1)$-dimensional true regression coefficient vector are sampled following a modified version of the scheme in Johnson (2013). Specifically

$$y_i = \mathbf{x}_i^\intercal \boldsymbol{\beta}^0 + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0,1), \quad i = 1, \ldots, n,$$

$$\beta_j^0 = \begin{cases} v_0|t| & \text{if } j = 0, \\ (-1)^u \left(v^2 \log(n)/\sqrt{n} + v|t|\right) & \text{if } j = 1, \ldots, p_0, \\ 0 & \text{otherwise,} \end{cases}$$

where $v_0^2 = 0.5$, $v^2 = 0.75$, $u \sim \mathcal{B}(0.4)$ and $t \sim \mathcal{N}(0,1)$. In order to study how the model borrows information from the co-data, we simulated co-data sources representing different degrees of information.

Variable selection is evaluated with the average area under the ROC curve (AUC) of the selected variable, where the posterior inclusion probability of the generic covariate j is evaluated with the quantity $\mathbb{E}_{\lambda_j}\left[\lambda_j^2/\left(1+\lambda_j^2\right)\right]$ (Carvalho et al., 2010).

The hyperparameters for σ^2 and κ_1^2 are set to $(v, q) = (1, 10)$ and $(a_1, b_1) = (1, 10)$, respectively. The Gibbs sampler is run for $B = 5000$ iterations with a burn-in period of $bn = 2500$ iterations, while the Variational algorithm is stopped either if the increase of the lower bound is less than $\epsilon = 0.001$ or if the maximum number of iteration $B = 1000$ is reached.

1. Performance of variable selection.
Here we analyse how the model behave as the degree of prior information varies. We study the cases $n = (50, 100, 150)$ and $p = (500, 1000, 1500)$ and we set the number of true non-zero coefficients to $p_0 = 30$. We consider five degrees of prior information:

$G0)$ **no co-data** set-up: we include in the co-data matrix only the intercept, therefore $\mathbf{Z} = \mathbf{1}_p$;

$G1)$ **non-informative** set-up: a binary co-data source is included in the model by randomly selecting 100 regressors, therefore the co-data matrix $\mathbf{Z}$ is created from the binary vector $\mathbf{z} \in \{0,1\}^p$, where $z_j = 1$ if the j-th variable is selected, $z_j = 0$ otherwise;

$G2)$ **weakly informative** set-up: a binary co-data source is included in the model by randomly selecting 20 of the true non-zero regressors and 80 of the true zero regressors, therefore the co-data matrix $\mathbf{Z}$ is created from the binary vector $\mathbf{z} \in \{0,1\}^p$, where $z_j = 1$ if the j-th variable is selected, $z_j = 0$ otherwise;

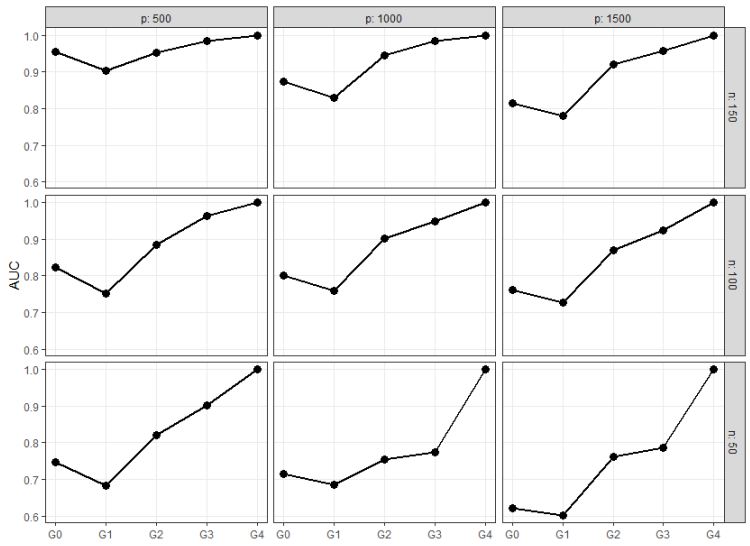


Figure 4.6.1: variable selection with Variational algorithm; the AUC is evaluated over 25 replicates of the experiments.

$G3$) **informative** set-up: a binary co-data source is included in the model by randomly selecting 20 of the true non-zero regressors and 10 of the true zero regressors, therefore the co-data matrix $\mathbf{Z}$ is created from the binary vector $\mathbf{z} \in \{0,1\}^p$, where $z_j = 1$ if the j-th variable is selected, $z_j = 0$ otherwise;

$G4$) **perfect co-data information** set-up: the co-data matrix $\mathbf{Z}$ is created from the binary vector $\mathbf{z} \in \{0,1\}^p$, where $z_j = 1$ if $\beta_j^0 \neq 0$, $z_j = 0$ otherwise.

The results are shown in Figure 4.6.1. The model is able to learn from the auxiliary information and the variable selection performance improves when the co-data is actually informative. The AUC, indeed, increases alongside the magnitude of prior information and the variables are perfectly selected when the co-data provides perfect information (case $G4$). On the other hand, random co-data is associated with a loss of performance and leads to lower scores of AUC. In particular, if we compare the case with random co-data ($G1$) and no co-data ($G0$), the latter performs slightly better. This difference, however, vanishes as the number of covariates increases. Finally, the smallest increase in AUC, as expected, is between $G2$ and $G3$, since these cases represent the most similar degree of co-data information.

2. Gibbs sampler vs Variational inference.
The accuracy of the Variational approximation is evaluated by comparing it to the Gibbs sampler in terms of variable selection and mean squared error (MSE) between $\boldsymbol{\beta}$ estimates and the true vector $\boldsymbol{\beta}^0$. We analysed the cases $n = (50, 100)$

$MSE_{\boldsymbol{\beta}} = \|\boldsymbol{\beta} - \boldsymbol{\beta}^0\|_2^2$							
	$p = 75$		$p = 125$		$p = 200$		
	FB	VB	FB	VB	FB	VB	
$G0$	0.165	0.168	0.473	0.246	0.236	0.208	$n = 50$
$G1$	0.162	0.164	0.302	0.242	0.206	0.207	
$G2$	0.147	0.139	0.238	0.191	0.231	0.149	
$G3$	0.043	0.037	0.188	0.025	0.165	0.029	
$G0$	0.025	0.025	0.055	0.030	0.045	0.040	$n = 100$
$G1$	0.025	0.025	0.026	0.029	0.045	0.039	
$G2$	0.020	0.020	0.020	0.013	0.013	0.014	
$G3$	0.008	0.008	0.012	0.005	0.003	0.003	

Table 4.6.1: Mean of $MSE_{\boldsymbol{\beta}}$ evaluated over 10 replicates of the experiments.

and $p = (75, 125, 200)$ and we set $p_0 = 30$. We considered four degrees of prior information and we refer to Appendix 4.C for the simulation scheme of the co-data.

The results of the variable selection are shown in Figure 4.C.1 in Appendix 4.C. The two methods behave similarly in all the considered cases. In particular, the results are equal for the lowest dimensional case, whereas the Variational approximation approaches the performance of the Gibbs sampler when p increases. Table 4.6.1 reports the details of the comparison between the two methods in terms of $\boldsymbol{\beta}$ estimate. By evaluating the mean of the MSE between the estimated $\boldsymbol{\beta}$ and the true regressor coefficients vector $\boldsymbol{\beta}^0$, we see that the Variational estimate does not deteriorate as p increases and it provides a equal or better solution compared to the Gibbs sampler. In particular, when n is large and the co-data are strongly informative the two methods provide almost the same estimate.

The Gibbs sampler, however, presents one main drawback: for some settings of n and p, the sampling method presented in Section 4.4 suffers from low acceptance probability. Therefore, the efficiency of the Gibbs sampler is heavily affected in a negative way as shown in Figure 4.C.2 in Appendix 4.C: even if the Gibbs is fairly efficient overall, there are some extreme points, suggesting that the sampling algorithm got stuck for some settings. Given the good results of the Variational approximation and the poor performance in terms of computational efficiency of the Gibbs sampler, we rely on the former for the following applications.

4.7 Application to real data

In this section we present two genomics applications. The first allows to evaluate our method for binary co-data in a multiple linear regression context. The second involves multiple co-data sources in a classification setting with a large number of variables.

In Section 4.6 we show that the thresholding variable selection works well when the covariates are sampled independently. However, the posterior probabilities are treated separately and the optimal threshold is (almost) never 0.5. For this reason in this section we also consider the more recent selection procedure called *decoupling shrinkage and selection* (DSS) proposed by Hahn and Carvalho (2015). This method was introduced to deal with potentially very strong correlations, as they are present in many genomics datasets. The authors propose a posterior variable selection summary based on the posterior mean of the predictors which results in a sequence of sparse models. Shrinkage can be achieved with any prior distribution and is 'decoupled' from the selection approach based on the posterior distribution. DSS method relies on the optimization of a loss function which balances the prediction error and the sparseness of the solution. Given an estimate $\hat{\boldsymbol{\beta}}$, the solution is obtained via adaptive LASSO by solving the following optimization problem,

$$\boldsymbol{\theta}^{DSS} = \text{argmin}_{\boldsymbol{\theta}} \frac{1}{n}\|\mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda \sum_{j=0}^{p} \frac{|\theta_j|}{|\hat{\beta}_j|},$$

where the smoothing parameter λ operates as a thresholding parameter and can be estimated with cross-validation over a set of values (grid search). The authors advocate this method over thresholding mainly because it naturally handles multicollinearity.

4.7.1 Case study 1: p38MAPK pathway

Following Kpogbezan et al. (2019), the model is tested on the p38MAPK pathway dataset. We investigate the effect of single nucleotide polymorphisms (SNPs) on the genes in the pathway. A subset of the data collected in the GEUVADIS project (Lappalainen et al., 2013) is considered. For each of the 99 genes we collect a different number p_t of SNPs, $t = 1, \ldots, 99$, with minimum $p_t = 56$ and maximum $p_t = 1169$. After a first sub-selection, $n = 373$ RNA-Seq samples are obtained from the 1000 Genomes Project. Since it is believed that SNPs within the gene's range have stronger influence on the gene's expression, a binary co-data source $\mathbf{z}_t \in \{0, 1\}^{p_t}$ is included in the analysis, where $z_{t,j} = 1$ if the j-th SNP related to gene t is located inside the gene's range and $z_{t,j} = 0$ otherwise.

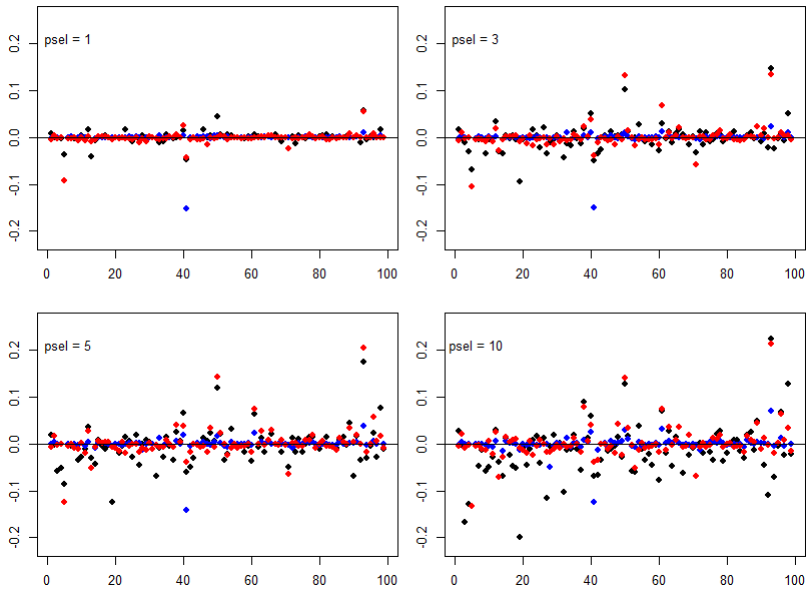


Figure 4.7.1: relative reduction of MSE (rrMSE) of LASSO (black dots), infHS regression with DSS selection procedure (red dots) and infHS regression with thresholding selection procedure (blue dots) for all the 99 genes. For each panel, the maximum number of selected SNPs is fixed to 1, 3, 5, and 10.

The hyperparameters for σ^2 and κ_1^2 are set to $(v, q) = (1, 10)$ and $(a_1, b_1) = (1, 10)$. The algorithm is stopped either if the increase of the variational lower bound is less than $\epsilon = 0.001$ or if the maximum number of iteration $B = 1000$ is reached.

To evaluate the prediction performance of the model, the data are divided in training-set ($n_1 = 249$) and test-set ($n_2 = 124$). We rely on the relative reduction of the mean squared error (rrMSE) for all the 99 genes. The rrMSE for the t-th regression model is defined as

$$\text{rrMSE}_t = 1 - \frac{\text{MSE}_t}{\text{MSE}_0},$$

where MSE_0 and MSE_t are the mean squared error of the null model and the mean squared error of the t-th linear model, respectively. The mean of $\mathbf{y}$ is centered around zero, therefore we do not include the intercept in the null model. Note that larger values of rrMSE_t are associated with better predictive performance. On the contrary, negative values denote worse predictive performance than the null model.

We test our method against LASSO regression (Tibshirani, 1996). The results are shown is Figure 4.7.1 for different degrees of sparseness. For the infHS regres-

sion, the SNPs are selected with both the thresholding procedure used in Section 4.6 (blue dots) and the DSS method (red dots). The latter provides a better prediction for all genes when compared with the thresholding variable selection. This is justified by the strong correlations among the SNPs. When comparing infHS and LASSO regressions, instead, the considerations made in Kpogbezan et al. (2019) hold. The results of the LASSO are more noisy, likely due to less shrinkage of the near-zero coefficients, whereas most of the rrMSE_t are concentrated around zero for the infHS model. The effect of the SNPs are relevant only for a small number of genes: the largest values of rrMSE_t are associated to genes 50, 61 and 93. Both infHS and LASSO are able to capture these effects, however the latter is more prone to negative values of rrMSE_t. The LASSO gives better results for gene 98. Note that the method by Kpogbezan et al. (2019) gives fairly similar results to ours, as it is also based on the Horseshoe. That method, however, is much more limited in use, as it only handles one discrete co-data source and only continuous outcomes.

Our method take 68 minutes to estimate all the 99 regressions and to evaluate the different variable selection approaches. The algorithm is run on a x64 Windows 11 operating system.

4.7.2 Case study 2: methylation data

Let $\mathbf{y} \in \{0,1\}^n$ be a binary vector and $\mathbf{X} \in \mathbb{R}^{n\times(p+1)}$ be the design matrix. Following Albert and Chib (1993) we introduce the latent variable $\mathbf{w} \in \mathbb{R}^n$ to reach a conjugated framework for updating regression coefficient vector $\boldsymbol{\beta}$. The assumptions of the probit model are

$$
\begin{aligned}
y_i \mid \mathbf{w} &= \begin{cases} 1 & \text{if } w_i > 0 \\ 0 & \text{if } w_i \leq 0, \end{cases} \\
w_i \mid \mathbf{X}, \boldsymbol{\beta} &\sim \mathcal{N}\left(\mathbf{x}_i^\intercal \boldsymbol{\beta}, 1\right), \quad i = 1, \ldots, n, \\
\beta_0 \mid \tau^2, \lambda_0^2 &\sim \mathcal{N}\left(0, \tau^2\lambda_0^2\right) \\
\beta_j \mid \tau, \lambda_j &\sim \mathcal{N}\left(0, \tau^2\lambda_j^2\right), \quad j = 1, \ldots, p.
\end{aligned}
$$

The prior distributions for parameters $\boldsymbol{\lambda}, \boldsymbol{\gamma}$ and τ are the same of Section 4.2 and the algorithm follows the same steps of Section 4.3. The introduction of the latent variable $\mathbf{w}$ allows to formulate the problem as a normal regression model on the latent variables w_i. The normal prior distribution for the regression parameters vector $\boldsymbol{\beta}$ is conjugated to the the normal distribution of $\mathbf{w}$. The joint posterior distribution of the model is

$$
\pi\left(\boldsymbol{\theta}, \mathbf{w}, \mathbf{y}, \mathbf{X}, \mathbf{Z}\right) \propto p\left(\mathbf{y} \mid \mathbf{w}\right) \cdot \pi\left(\mathbf{w} \mid \mathbf{X}, \boldsymbol{\theta}\right) \cdot \pi\left(\boldsymbol{\theta} \mid \mathbf{Z}\right),
$$

where $\boldsymbol{\theta} = (\boldsymbol{\beta}, \lambda_0^2, \boldsymbol{\lambda}, \psi_0, \boldsymbol{\varphi}, \boldsymbol{\gamma}, \boldsymbol{\kappa}, \tau^2, \zeta)$. Under the mean field approximation, the variational factors for parameters $\mathbf{w}$ and $\boldsymbol{\beta}$ are updated as

$$
\begin{aligned}
q^\star(w_i) &= \begin{cases} \mathcal{N}_+\left(\mu_i^\star, 1\right) & \text{if } y_i = 0 \\ \mathcal{N}_-\left(\mu_i^\star, 1\right) & \text{if } y_i = 0, \end{cases} \\
q^\star(\boldsymbol{\beta}) &= \mathcal{N}_p\left(\boldsymbol{\mu}_\beta^\star, \boldsymbol{\Sigma}_\beta^\star\right), \\
&\qquad \boldsymbol{\mu}_\beta^\star = \boldsymbol{\Sigma}_\beta^\star \mathbf{X}^\intercal \mathbb{E}_w[\mathbf{w}], \quad \boldsymbol{\Sigma}_\beta^\star = \left(\mathbf{X}^\intercal\mathbf{X} + \mathbb{E}_{\lambda_0^2 \cdot \boldsymbol{\lambda} \cdot \tau^2}\left[\tau^{-2}\boldsymbol{\Lambda}^{-2}\right]\right)^{-1},
\end{aligned}
$$

where $\mu_i^\star = \mathbf{x}_i^\intercal \mathbb{E}_\beta[\boldsymbol{\beta}]$ and $\mathcal{N}_+$ and $\mathcal{N}_-$ denote a normal distribution left and right truncated at 0, respectively. The expectation of latent variables w_i is

$$
\mathbb{E}_w[w_i] = \begin{cases} \mu_i^\star + \frac{\phi_i}{1-\Phi_i} & \text{if } y_i = 0 \\ \mu_i^\star - \frac{\phi_i}{\Phi_i} & \text{if } y_i = 0, \end{cases}
$$

where $\phi_i = \phi(-\mu_i^\star)$ and $\Phi_i = \Phi(-\mu_i^\star)$ are the normal density and the normal cumulative density functions, respectively. The variational lower bound is computed as

$$
\begin{aligned}
\mathcal{L} &= \mathbb{E}_{q^\star}[\log \pi(\boldsymbol{\theta}, \mathbf{w}, \mathbf{y}, \mathbf{X}, \mathbf{Z})] - \mathbb{E}_{q^\star}[\log q^\star(\boldsymbol{\theta}, \mathbf{w})] \\
&= \mathbb{E}_{q^\star}[\log p(\mathbf{y} \mid \mathbf{w})] + \mathbb{E}_{q^\star}[\log \pi(\mathbf{w} \mid \mathbf{X}, \boldsymbol{\beta})] + \mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\beta} \mid \tau^2, \lambda_0^2, \boldsymbol{\lambda}\right)\right] + \\
&\quad \mathbb{E}_{q^\star}\left[\log \pi\left(\lambda_0^2 \mid \psi_0\right)\right] + \mathbb{E}_{q^\star}[\log \pi(\psi_0)] + \mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\lambda} \mid \mathbf{Z}, \boldsymbol{\gamma}, \boldsymbol{\varphi}^2\right)\right] + \mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\varphi}^2\right)\right] + \\
&\quad \mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\gamma} \mid \boldsymbol{\kappa}^2\right)\right] + \mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\kappa}^2\right)\right] + \mathbb{E}_{q^\star}\left[\log \pi\left(\tau^2 \mid \zeta\right)\right] + \mathbb{E}_{q^\star}[\log \pi(\zeta)] - \\
&\quad \mathbb{E}_{q^\star}[\log q^\star(\mathbf{w})] - \mathbb{E}_{q^\star}[\log q^\star(\boldsymbol{\beta})] - \mathbb{E}_{q^\star}\left[\log q^\star\left(\lambda_0^2\right)\right] - \mathbb{E}_{q^\star}[\log q^\star(\psi_0)] - \mathbb{E}_{q^\star}[\log q^\star(\boldsymbol{\lambda})] - \\
&\quad \mathbb{E}_{q^\star}\left[\log q^\star\left(\boldsymbol{\varphi}^2\right)\right] - \mathbb{E}_{q^\star}[\log q^\star(\boldsymbol{\gamma})] - \mathbb{E}_{q^\star}\left[\log q^\star\left(\boldsymbol{\kappa}^2\right)\right] - \mathbb{E}_{q^\star}\left[\log q^\star\left(\tau^2\right)\right] - \mathbb{E}_{q^\star}[\log q^\star(\zeta)] \\
&\propto \sum_{i=1}^{n}\left(y_i \log(1-\Phi_i) + (1-y_i)\log(\Phi_i)\right) + \frac{1}{2}\left(\log|\boldsymbol{\Sigma}_\beta^\star| + \log|\boldsymbol{\Sigma}_\gamma^\star|\right) + \\
&\quad \sum_{j=1}^{p}(\log s_j - \log k_j) + \sum_{j=1}^{p}\left(a_j^\star \mathbb{E}_\lambda\left[\lambda_j^{-2}\right] + b_j^\star \mathbb{E}_\lambda\left[\lambda_j^2\right] - c_j^\star \mathbb{E}_\lambda[\lambda_j] - \log d_j^\star\right) - \\
&\quad \sum_{d=1}^{D} e_d^\star \log f_d^\star - \left(\frac{p}{2} + 1\right)\log g^\star - \log h^\star - \log a_0^\star - \log k_0^\star, \qquad (4.6)
\end{aligned}
$$

where all the quantities are defined in Appendix 4.A.2. The method requires the computation of $\boldsymbol{\Sigma}_\beta^\star \mathbf{X}^\intercal \mathbb{E}_w[w]$ and $\operatorname{diag}\left(\boldsymbol{\Sigma}_\beta^\star\right)$, which can efficiently be evaluated by applying the strategy in Appendix 4.B.

The model is tested on the methylation dataset of Verlaat et al. (2018), which contains methylation profiles of self-collected cervicovaginal lavages corresponding

Algorithm 15: Variational Bayes approximation for probit informative Horseshoe regression

Input: $\mathbf{y} \in \{0,1\}^n$, $\mathbf{X} \in \mathbb{R}^{n\times(p+1)}$, $\mathbf{Z}_1, \dots, \mathbf{Z}_D \in \mathbb{R}^{p\times m_d}$, $\mathbf{a}, \mathbf{b} \in \mathbb{R}_+^D$, $s_0^2 \in \mathbb{R}_+$;
// Set $b = 1$, $\epsilon = 10^{-3}$, $\mathcal{L}^{(0)} = -\infty$ and initialize all the needed moments
while $\mathcal{L}^{(b)} - \mathcal{L}^{(b-1)} > \epsilon$ **do**
  Update parameter $\mu_i^\star$, for $i = 1, \dots, n$;
  Update parameter $\boldsymbol{\mu}_\beta^\star$ and compute the quantities diag $\left(\boldsymbol{\Sigma}_\beta^\star\right)$ and $|\boldsymbol{\Sigma}_\beta^\star|$ as in Appendix 4.B;
  Update parameters $a_0^\star$ and $k_0^\star$;
  Update parameters $a_j^\star$, $b_j^\star$, $c_j^\star$ and $d_j^\star$ and evaluate the normalizing constant s_j, $\mathbb{E}_\lambda[\lambda_j]$, $\mathbb{E}_\lambda\left[\lambda_j^2\right]$ and $\mathbb{E}_\lambda\left[\lambda_j^{-2}\right]$ with numerical integration, for $j = 1, \dots, p$;
  Update parameters $\boldsymbol{\mu}_\gamma^\star$ and $\boldsymbol{\Sigma}_\gamma^\star$;
  Update parameters $e_d^\star$ and $f_d^\star$, for $d = 1, \dots, D$;
  Update parameters $g^\star$ and $h^\star$;
  Compute $\mathcal{L}^{(b)}$ and set $b = b + 1$;
end
return $\boldsymbol{\mu}_\beta^\star$, $a_0^\star$, $\mathbf{a}^\star$, $\mathbf{b}^\star$, $\mathbf{c}^\star$, $\boldsymbol{\mu}_\gamma^\star$, $\boldsymbol{\Sigma}_\gamma^\star$, $\mathbf{e}^\star$, $\mathbf{f}^\star$, $g^\star$;

to 28 women with normal cervix and 36 women with high-grade precursor lesions (CIN3), for a total of $n = 64$ samples. In order to improve the diagnostic classification we include in the analysis $D = 5$ co-data sources: the standard deviations of the probes, p-values from the previous study, a binary variable differentiating the hypo-methylated and hyper-methylated probes, a categorical variable with 6 categories denoting the genomic region of each probe and the probes' means in cancer cells. Based on the previous results, we consider the probes with an external p-value $p_\alpha < 0.005$, resulting in a total of $p = 11251$ probes, where each probe measures the methylation of a unique location on the genome.

We test our model against the ordinary Ridge regression and the LASSO. We include in the analysis also the thresholding and the DSS versions of infHS model. The former do not exclude any of the covariates from the model, since all the posterior inclusion probabilities were greater than 0.5. Therefore the results of this approach are not shown. The hyperparameters are set to $(a_d, b_d) = (1, 10)$ for $d = 1, \dots, 5$ and the predictive performances are evaluated by leave-one-out cross-validation (LOOCV). As for case study 1, the algorithm is stopped either if the

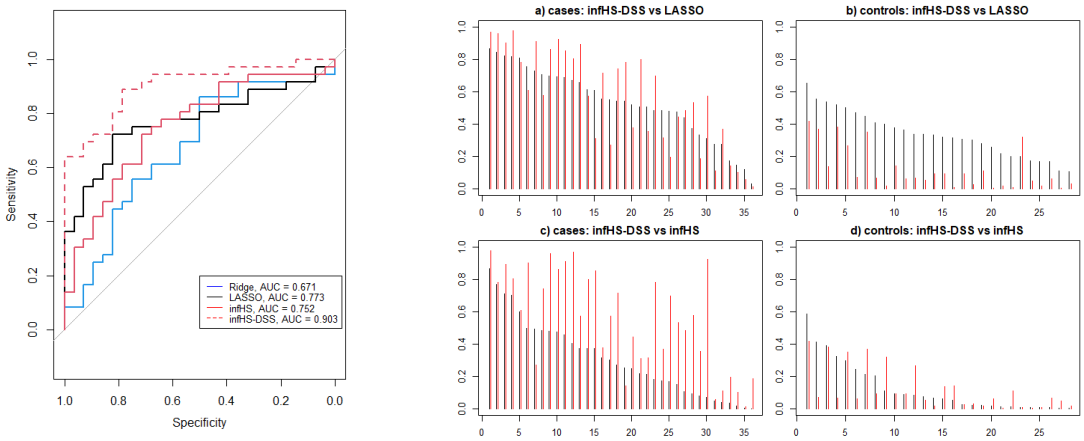


Figure 4.7.2: Results of the LOOCV. Left panel: ROC curves for Ridge regression (blue), LASSO (black), the informative Horseshoe regression (red) and the informative Horseshoe with DSS variable selection procedure (dotted-red); right panel: *a* and *b*. predicted probabilities for cases ($y_i = 1$) and controls ($y_i = 0$) with infHS-DSS (red) and LASSO (black) in decreasing order of LASSO prediction; *c* and *d*. predicted probabilities for cases and controls with infHS-DSS (red) and ordinary infHS (black) in decreasing order of infHS prediction.

increase of the variational lower bound is less than $\epsilon = 0.001$ or if the maximum number of iteration $B = 1000$ is reached.

The left panel of Figure 4.7.2 shows the ROC curves for the considered competitors, where the predicted probabilities for infHS are computed as $p_i = \Phi\left(\mathbf{x}_i^\intercal \boldsymbol{\beta}\right)$. InfHS-DSS gives the best forecasting results with AUC = 0.903, greatly improving compared to its ordinary version with AUC = 0.752; the LASSO performs better than infHS with AUC = 0.773 and provides the sparsest model, with an average of 8 variables included as opposed to the 49 estimated by infHS-DSS. Among these 49 probes, 37 of them are selected in more than 70% of the folds, whereas only 10 appear in all. The ordinary Ridge does not compete in terms of predictive performance (AUC = 0.671). Note that GRridge co-data method by Van de Wiel et al. (2016) was previously reported to achieve an AUC = 0.77 (Verlaat et al., 2018).

The right panel of Figure 4.7.2 shows the predicted probabilities of infHS-DSS model against the LASSO and the ordinary infHS. When compared to the LASSO (figures *a* and *b*, values in decreasing order of LASSO prediction), infHS-DSS gives similar predictions for the cases ($y_i = 1$), with some of the observations achieving higher predictive scores and other assuming lower ones; on the other hand, it provides significantly lower scores for the controls ($y_i = 0$), reducing the false positive

rate and improving the overall performance. The impact of the DSS procedure on the prediction can be assessed by analyzing figures *c* and *d*, which compares infHS-DSS to infHS model (values in decreasing order of infHS prediction). The former completely overwhelms the latter, as it gives higher probabilities for almost all cases and reduces the predicted probabilities for the highest scores of the controls. To sum up these results we compute the mean absolute difference between the true labels and the predicted probabilities. The results are shown in Table 4.7.1, where infHS-DSS provides the best performance.

	Ridge	LASSO	infHS	infHS-DSS
$\sum_{i=1}^{64} \frac{\lvert y_i - p_i \rvert}{64}$	0.466	0.410	0.444	**0.299**

Table 4.7.1: Mean absolute difference between true labels ($y_i = 0$ or $y_i = 1$) and the predicted probabilities ($p_i = \Phi\left(\mathbf{x}_i^\intercal \boldsymbol{\beta}\right)$).

We summarize the co-data information in table 4.7.2, which shows the distribution of the overall co-data sources and the distribution of the $p_{sel} = 37$ most selected probes. The co-data related to the the probes' means in cancer cells show the strongest difference in distribution between the original and the selected features, suggesting that this source is the most informative in terms of prior information. No evident differences are found in the other co-data sources.

The computational time is around 5 minutes for the estimation of infHS. The method is tested on a x64 Windows 11 operating system and the local variances are evaluated in parallel with 4 cores.

4.8 Discussion

We introduced a novel Bayesian regression approach for high-dimensional data able to learn from auxiliary prior information, i.e. co-data. We showed that both prediction and variable selection benefit from the inclusion of co-data, when these are actually informative. The model allows for both continuous and binary outcome, as well as both continuous and discrete co-data sources. In particular, we provided a flexible method that estimates multiple co-data coefficients jointly, contrary to the previous method of Van Nee et al. (2021) that models each source separately.

We discussed a full Bayesian approach, where we developed a Gibbs sampler to update each parameter iteratively by sampling from their full-conditional distributions. To do so, we introduced a novel rejection sampling method to sample the local variances, which showed a non-closed form target density. Eventually,

Co-data	Selected probes distribution ($p_{sel} = 37$)	Original distribution ($p = 11251$)
external p-value	mean: 0.0016 sd: 0.0014 range: $(2.4 \cdot 10^{-6}, 0.005)$	mean: 0.0019 sd: 0.0015 range: $(9.9 \cdot 10^{-11}, 0.005\}$
probes sd	mean: 0.035 sd: 0.022 range: $(0.014, 0.129)$	mean: 0.038 sd: 0.019 range: $(0.009, 0.320)$
genomic region	Distant: 35.1% Island: 43.2% N shelf: 5.4% N shore 10.8% S shelf: 0% S shore: 5.4%	Distant: 31.7% Island: 37.3% N shelf: 4.6% N shore: 12.4% S shelf: 4% S shore: 10%
degree of methylation	Hypo-methylated: 57% Hyper-methylated: 43%	Hypo-methylated: 56% Hyper-methylated: 44%
probes' mean in cancer cells	mean: 0.150 sd: 1.116 range: $(-1.28, 3.34)$	mean: -0.099 sd: 1.137 range: $(-4.89, 8.10)$

Table 4.7.2: Summary of the co-data estimates and the co-data distribution in the $p_{sel} = 37$ most selected probes and in the original population.

we proposed a Variational approximation to the joint posterior distribution and applied the CAVI algorithm to optimize the target density. This latter method is suited for high-dimensional problems, as it does not require the explicit inversion of the $p \times p$ posterior covariance matrix $\boldsymbol{\Sigma}^{\star}_{\beta}$. In particular, only its diagonal is required and we implemented the methods proposed in Münch et al. (2019) to efficiently achieve this. Beside this, another computational advantage is the parallel evaluation of the local variances. However, the evaluation of these parameters represents the computational bottleneck of the method: the acceptance probability of the rejection sampling is small for some settings of the parameters, whereas the numerical integration required in the Variational approximation is the most expensive step of the algorithm.

The Variational inference is less useful than the Gibbs sampler for posterior inference, as it provides posterior means point estimates and underestimates the posterior variances (MacKay et al., 2003; Wang and Titterington, 2005; Turner and Sahani, 2011; Giordano et al., 2017). This lack of accuracy, on the other

hand, does not necessarily affect the performance of the model (Blei and Jordan, 2006). Therefore, the Variational algorithm should be used for large p datasets, whereas the Gibbs sampling could be useful when the interest is in the posterior inference (credible intervals) and the number of covariates is moderate.

A possible limitation of infHS for some applications occurs when a categorical co-data source contains one very strong, relatively small co-data group. Under this circumstance, the sparsity assumption may not be realistic for this particular group of variables. For such applications, an interesting extension of infHS would be to allow a dense prior for a small group of variables, for which one expects a particularly relevant prior evidence.

Another extension to allow more flexibility would be the specification of the prior $\lambda_j \sim$ Half-$t(v)$, with $v > 1$. Here $v = 1$ since the purpose is to apply the Horseshoe prior. In Biswas et al. (2021) the authors develop coupling techniques for high-dimensional regression with this particular prior and argue that larger values of v can affect the statistical and computational performance of the model.

To conclude, we provided one of the fastest implementation of Horseshoe regression able to learn from auxiliary information. The R code for package `infHS` is available at https://github.com/cbusatto/infHS.

Appendix

Appendix 4.A Details of the Variational lower bound

Here we report the components for the evaluation of the lower bound $\mathcal{L}$ in (4.5) and (4.6). The results below rely on the property $\mathbb{E}_x\left[\mathbf{x}^\intercal \mathbf{A}\mathbf{x}\right] = \mathbb{E}_x(\mathbf{x})^\intercal \mathbf{A}\mathbb{E}_x(\mathbf{x}) + tr\left(\mathbf{A}\mathbf{V}\right)$, where $\mathbf{V}$ is the correlation matrix of generical random variable $\mathbf{x}$.

4.A.1 Linear regression

The variational lower bound $\mathcal{L}$ in (4.5) is computed with the following components:

$$
\begin{aligned}
\mathbb{E}_{q^\star}\left[\log p\left(\mathbf{y} \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\lambda}\right)\right] \propto & -\frac{n}{2}\mathbb{E}_{\sigma^2}\left[\log \sigma^2\right] - \frac{1}{2}\mathbb{E}_{\beta\cdot\sigma^2}\left[\frac{\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2}{\sigma^2}\right] \\
\mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\beta} \mid \sigma^2, \tau^2, \lambda_0^2, \boldsymbol{\lambda}\right)\right] \propto & -\frac{p+1}{2}\mathbb{E}_{\sigma^2}\left[\log \sigma^2\right] - \frac{p+1}{2}\mathbb{E}_{\tau^2}\left[\log \tau^2\right] - \frac{1}{2}\mathbb{E}_{\lambda_0^2}\left[\log \lambda_0^2\right] - \\
& \sum_{j=1}^{p}\mathbb{E}_{\lambda}\left[\log \lambda_j\right] - \frac{1}{2}\sum_{j=0}^{p}\mathbb{E}_{\beta_0\cdot\beta\cdot\lambda_0^2\cdot\boldsymbol{\lambda}\cdot\tau^2\cdot\sigma^2}\left[\frac{\beta_j^2}{\sigma^2\tau^2\lambda_j^2}\right] \\
\mathbb{E}_{q^\star}\left[\log \pi\left(\lambda_0^2 \mid \psi_0\right)\right] \propto & -\frac{1}{2}\mathbb{E}_{\psi_0}\left[\log \psi_0\right] - \frac{3}{2}\mathbb{E}_{\lambda_0^2}\left[\log \lambda_0^2\right] - \mathbb{E}_{\lambda_0^2\cdot\psi_0}\left[\frac{1}{\psi_0\lambda_0^2}\right] \\
\mathbb{E}_{q^\star}\left[\log \pi\left(\psi_0\right)\right] \propto & -\frac{3}{2}\mathbb{E}_{\psi_0}\left[\log \psi_0\right] - \mathbb{E}_{\psi_0}\left[\psi_0^{-1}\right] \\
\mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\lambda} \mid \boldsymbol{\gamma}, \boldsymbol{\varphi}^2\right)\right] \propto & \sum_{j=1}^{p}\left(-\log k_j - \frac{1}{2}\mathbb{E}_{\varphi^2}\left[\log \varphi_j^2\right] - \frac{1}{2s_0^2}\mathbb{E}_{\lambda\cdot\gamma\cdot\varphi^2}\left[\frac{\left(\lambda_j - \mathbf{z}_j^\intercal\boldsymbol{\gamma}\right)^2}{\varphi_j^2}\right]\right) \\
\mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\varphi}^2\right)\right] \propto & -\frac{1}{2}\sum_{j=1}^{p}\left(3\mathbb{E}_{\varphi^2}\left[\log \varphi_j^2\right] + \mathbb{E}_{\varphi^2}\left[\varphi_j^{-2}\right]\right) \\
\mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\gamma} \mid \boldsymbol{\kappa}^2\right)\right] \propto & -\frac{1}{2}\sum_{d=1}^{D}\left(m_d\mathbb{E}_{\kappa^2}\left[\log \kappa_d^2\right] + \mathbb{E}_{\gamma\cdot\kappa^2}\left[\frac{\boldsymbol{\gamma}_d^\intercal\boldsymbol{\gamma}_d}{\kappa_d^2}\right]\right)
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\kappa}^2\right)\right] \propto & -\sum_{d=1}^{D}\left((a_d+1)\mathbb{E}_{\kappa^2}\left[\log \kappa_d^2\right]+b_d\mathbb{E}_{\kappa^2}\left[\kappa_d^{-2}\right]\right)\\
\mathbb{E}_{q^\star}\left[\log \pi\left(\tau^2 \mid \zeta\right)\right] \propto & -\frac{1}{2}\mathbb{E}_{\zeta}\left[\log \zeta\right]-\frac{3}{2}\mathbb{E}_{\tau^2}\left[\log \tau^2\right]-\mathbb{E}_{\tau^2\cdot\zeta}\left[\frac{1}{\zeta\tau^2}\right]\\
\mathbb{E}_{q^\star}\left[\log \pi\left(\zeta\right)\right] \propto & -\frac{3}{2}\mathbb{E}_{\zeta}\left[\log \zeta\right]-\mathbb{E}_{\zeta}\left[\zeta^{-1}\right]\\
\mathbb{E}_{q^\star}\left[\log \pi\left(\sigma^2\right)\right] \propto & -\left(v+1\right)\mathbb{E}_{\sigma^2}\left[\log \sigma^2\right]-q\mathbb{E}_{\sigma^2}\left[\sigma^{-2}\right]\\
\mathbb{E}_{q^\star}\left[\log q^\star\left(\boldsymbol{\beta}\right)\right] \propto & -\frac{p+1}{2}\mathbb{E}_{\sigma^2}\left[\log \sigma^2\right]-\frac{1}{2}\log\left|\boldsymbol{\Sigma}_{\beta}^{\star}\right|-\frac{p+1}{2}\\
\mathbb{E}_{q^\star}\left[\log q^\star\left(\lambda_0^2\right)\right] \propto & \log\left(a_0^\star\right)-2\mathbb{E}_{\lambda_0^2}\left[\log \lambda_0^2\right]-\mathbb{E}_{\lambda_0^2\cdot\psi_0}\left[\frac{1}{\psi_0\lambda_0^2}\right]-\frac{1}{2}\mathbb{E}_{\beta_0\cdot\lambda_0^2\cdot\tau^2\cdot\sigma^2}\left[\frac{\beta_0^2}{\sigma^2\tau^2\lambda_0^2}\right]\\
\mathbb{E}_{q^\star}\left[\log q^\star\left(\psi_0\right)\right] \propto & \log k_0^\star-2\mathbb{E}_{\psi_0}\left[\log \psi_0\right]-\mathbb{E}_{\psi_0}\left[\psi_0^{-1}\right]-\mathbb{E}_{\lambda_0^2\cdot\psi_0}\left[\frac{1}{\psi_0\lambda_0^2}\right]\\
\mathbb{E}_{q^\star}\left[\log q^\star\left(\boldsymbol{\lambda}\right)\right] \propto & \sum_{j=1}^{p}\Bigg(-\log s_j-\mathbb{E}_{\lambda}\left[\log \lambda_j\right]-\frac{1}{2}\mathbb{E}_{\beta\cdot\lambda\cdot\sigma^2\cdot\tau^2}\left[\frac{\beta_j^2}{\sigma^2\tau^2\lambda_j^2}\right]-\\
& \frac{1}{2s_0^2}\mathbb{E}_{\lambda\cdot\varphi^2}\left[\frac{\lambda_j^2}{\varphi_j^2}\right]+\frac{1}{s_0^2}\mathbf{z}_j^{\intercal}\mathbb{E}_{\lambda\cdot\gamma\cdot\varphi^2}\left[\frac{\boldsymbol{\gamma}\lambda_j}{\varphi_j^2}\right]\Bigg)\\
\mathbb{E}_{q^\star}\left[\log q^\star\left(\boldsymbol{\varphi}^2\right)\right] \propto & \sum_{j=1}^{p}\Bigg(\log d_j^\star-2\mathbb{E}_{\varphi^2}\left[\log \varphi_j^2\right]-\frac{1}{2}\mathbb{E}_{\varphi^2}\left[\varphi_j^{-2}\right]-\\
& \frac{1}{2s_0^2}\mathbb{E}_{\lambda\cdot\gamma\cdot\varphi^2}\left[\frac{\left(\lambda_j-\mathbf{z}_j^{\intercal}\boldsymbol{\gamma}\right)^2}{\varphi_j^2}\right]\Bigg)\\
\mathbb{E}_{q^\star}\left[\log q^\star\left(\boldsymbol{\gamma}\right)\right] \propto & -\frac{1}{2}\log\left|\boldsymbol{\Sigma}_{\gamma}^{\star}\right|-\frac{M}{2s_0^2}\\
\mathbb{E}_{q^\star}\left[\log q^\star\left(\kappa^2\right)\right] \propto & \sum_{d=1}^{D}\Bigg(\left(a_d+\frac{m_d}{2}\right)\log f_d^\star-b_d\mathbb{E}_{\kappa^2}\left[\kappa_d^{-2}\right]-\left(a_d+\frac{m_d}{2}+1\right)\mathbb{E}_{\kappa^2}\left[\log \kappa_d^2\right]-\\
& \frac{1}{2}\mathbb{E}_{\gamma\cdot\kappa^2}\left[\frac{\boldsymbol{\gamma}_d^{\intercal}\boldsymbol{\gamma}_d}{\kappa_d^2}\right]\Bigg)\\
\mathbb{E}_{q^\star}\left[\log q^\star\left(\tau^2\right)\right] \propto & \left(\frac{p}{2}+1\right)\log g^\star-\left(\frac{p}{2}+2\right)\mathbb{E}_{\tau^2}\left[\log \tau^2\right]-\mathbb{E}_{\tau^2\cdot\zeta}\left[\frac{1}{\zeta\tau^2}\right]-\\
& \frac{1}{2}\sum_{j=0}^{p}\mathbb{E}_{\beta_0\cdot\beta\cdot\lambda_0^2\cdot\lambda\cdot\tau^2\cdot\sigma^2}\left[\frac{\beta_j^2}{\sigma^2\tau^2\lambda_j^2}\right]\\
\mathbb{E}_{q^\star}\left[\log q^\star\left(\zeta\right)\right] \propto & \log h^\star-2\mathbb{E}_{\zeta}\left[\log \zeta\right]-\mathbb{E}_{\zeta}\left[\zeta^{-1}\right]-\mathbb{E}_{\tau^2\cdot\zeta}\left[\frac{1}{\zeta\tau^2}\right]\\
\mathbb{E}_{q^\star}\left[\log q^\star\left(\sigma^2\right)\right] \propto & \left(v+\frac{n+p+1}{2}\right)\log l^\star-q\mathbb{E}_{\sigma^2}\left[\sigma^{-2}\right]-\left(v+\frac{n+p+3}{2}\right)\mathbb{E}_{\sigma^2}\left[\log \sigma^2\right]-\\
& \frac{1}{2}\mathbb{E}_{\beta\cdot\sigma^2}\left[\frac{\left\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right\|_2^2}{\sigma^2}\right]-\frac{1}{2}\sum_{j=0}^{p}\mathbb{E}_{\beta_0\cdot\beta\cdot\lambda_0^2\cdot\lambda\cdot\sigma^2\cdot\tau^2}\left[\frac{\beta_j^2}{\sigma^2\tau^2\lambda_j^2}\right]
\end{aligned}
$$

4.A.2 Probit regression

The variational lower bound $\mathcal{L}$ in (4.5) is computed with the following components:

$$\mathbb{E}_{q^\star}\left[\log p\left(\mathbf{y} \mid \mathbf{w}\right)\right] \propto \sum_{i=1}^{n}\left(\mathbb{E}_{w}\left[y_i \log(\mathbb{I}_{(w_i>0)}\right] + \mathbb{E}_{w}\left[(1-y_i)\log(\mathbb{I}_{(w_i<0)}\right]\right) = 0$$

$$\mathbb{E}_{q^\star}\left[\log p\left(\mathbf{w} \mid \boldsymbol{\beta}\right)\right] \propto -\frac{1}{2}\sum_{i=1}^{n}\mathbb{E}_{w}\left[\|w_i - \mathbf{x}_i^\intercal\boldsymbol{\beta}\|_2^2\right]$$

$$\mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\beta} \mid \sigma^2, \tau^2, \lambda_0^2, \boldsymbol{\lambda}\right)\right] \propto -\frac{p+1}{2}\mathbb{E}_{\tau^2}\left[\log \tau^2\right] - \frac{1}{2}\mathbb{E}_{\lambda_0^2}\left[\log \lambda_0^2\right] - \sum_{j=1}^{p}\mathbb{E}_{\lambda}\left[\log \lambda_j\right] - \frac{1}{2}\sum_{j=0}^{p}\mathbb{E}_{\beta_0\cdot\beta\cdot\lambda_0^2\cdot\lambda\cdot\tau^2}\left[\frac{\beta_j^2}{\tau^2\lambda_j^2}\right]$$

$$\mathbb{E}_{q^\star}\left[\log \pi\left(\lambda_0^2 \mid \psi_0\right)\right] \propto -\frac{1}{2}\mathbb{E}_{\psi_0}\left[\log \psi_0\right] - \frac{3}{2}\mathbb{E}_{\lambda_0^2}\left[\log \lambda_0^2\right] - \mathbb{E}_{\lambda_0^2\cdot\psi_0}\left[\frac{1}{\psi_0\lambda_0^2}\right]$$

$$\mathbb{E}_{q^\star}\left[\log \pi\left(\psi_0\right)\right] \propto -\frac{3}{2}\mathbb{E}_{\psi_0}\left[\log \psi_0\right] - \mathbb{E}_{\psi_0}\left[\psi_0^{-1}\right]$$

$$\mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\lambda} \mid \boldsymbol{\gamma}, \boldsymbol{\varphi}^2\right)\right] \propto \sum_{j=1}^{p}\left(-\log k_j - \frac{1}{2}\mathbb{E}_{\varphi^2}\left[\log \varphi_j^2\right] - \frac{1}{2s_0^2}\mathbb{E}_{\lambda\cdot\gamma\cdot\varphi^2}\left[\frac{\left(\lambda_j - \mathbf{z}_j^\intercal\boldsymbol{\gamma}\right)^2}{\varphi_j^2}\right]\right)$$

$$\mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\varphi}^2\right)\right] \propto -\frac{1}{2}\sum_{j=1}^{p}\left(3\mathbb{E}_{\varphi^2}\left[\log \varphi_j^2\right] + \mathbb{E}_{\varphi^2}\left[\varphi_j^{-2}\right]\right)$$

$$\mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\gamma} \mid \boldsymbol{\kappa}^2\right)\right] \propto -\frac{1}{2}\sum_{d=1}^{D}\left(m_d\mathbb{E}_{\kappa}\left[\log \kappa_d^2\right] + \mathbb{E}_{\gamma\cdot\kappa}\left[\frac{\boldsymbol{\gamma}_d^\intercal\boldsymbol{\gamma}_d}{\kappa_d^2}\right]\right)$$

$$\mathbb{E}_{q^\star}\left[\log \pi\left(\boldsymbol{\kappa}^2\right)\right] \propto \sum_{d=1}^{D}\left(-(a_d+1)\mathbb{E}_{\kappa^2}\left[\log \kappa_d^2\right] - b_d\mathbb{E}_{\kappa^2}\left[\kappa_d^{-2}\right]\right)$$

$$\mathbb{E}_{q^\star}\left[\log \pi\left(\tau^2 \mid \zeta\right)\right] \propto -\frac{1}{2}\mathbb{E}_{\zeta}\left[\log \zeta\right] - \frac{3}{2}\mathbb{E}_{\tau^2}\left[\log \tau^2\right] - \mathbb{E}_{\tau^2\cdot\zeta}\left[\frac{1}{\zeta\tau^2}\right]$$

$$\mathbb{E}_{q^\star}\left[\log \pi\left(\zeta\right)\right] \propto -\frac{3}{2}\mathbb{E}_{\zeta}\left[\log \zeta\right] - \mathbb{E}_{\zeta}\left[\zeta^{-1}\right]$$

$$\mathbb{E}_{q^\star}\left[\log q^\star\left(\mathbf{w}\right)\right] \propto \sum_{i=1}^{n}\left(-\frac{1}{2}\mathbb{E}_{w}\left[\|w_i - \mathbf{x}_i^\intercal\boldsymbol{\beta}\|_2^2\right] - y_i\log\left(1-\Phi_i\right) - (1-y_i)\log\left(\Phi_i\right)\right)$$

$$\mathbb{E}_{q^\star}\left[\log q^\star\left(\boldsymbol{\beta}\right)\right] \propto -\frac{1}{2}\log|\boldsymbol{\Sigma}_\beta^\star| - \frac{p+1}{2}$$

$$\mathbb{E}_{q^\star}\left[\log q^\star\left(\lambda_0^2\right)\right] \propto \log\left(a_0^\star\right) - 2\mathbb{E}_{\lambda_0^2}\left[\log \lambda_0^2\right] - \mathbb{E}_{\lambda_0^2\cdot\psi_0}\left[\frac{1}{\psi_0\lambda_0^2}\right] - \frac{1}{2}\mathbb{E}_{\beta_0\cdot\tau^2\cdot\lambda_0^2}\left[\frac{\beta_0^2}{\tau^2\lambda_0^2}\right]$$

$$
\begin{aligned}
\mathbb{E}_{q^\star}\left[\log q^\star\left(\psi_0\right)\right] \propto\ & \log k_0^\star - 2\mathbb{E}_{\psi_0}\left[\log \psi_0\right] - \mathbb{E}_{\psi_0}\left[\psi_0^{-1}\right] - \mathbb{E}_{\lambda_0^2\cdot\psi_0}\left[\frac{1}{\psi_0\lambda_0^2}\right] \\
\mathbb{E}_{q^\star}\left[\log q^\star\left(\boldsymbol{\lambda}\right)\right] \propto\ & \sum_{j=1}^{p}\Bigg(- \log s_j - \mathbb{E}_{\lambda}\left[\log \lambda_j\right] - \frac{1}{2}\mathbb{E}_{\beta\cdot\tau^2\cdot\lambda}\left[\frac{\beta_j^2}{\tau^2\lambda_j^2}\right] - \frac{1}{2s_0^2}\mathbb{E}_{\lambda\cdot\varphi^2}\left[\frac{\lambda_j^2}{\varphi_j^2}\right] + \\
& \frac{1}{s_0^2}\mathbf{z}_j^\intercal\mathbb{E}_{\lambda\cdot\gamma\cdot\varphi^2}\left[\frac{\boldsymbol{\gamma}\lambda_j}{\varphi_j^2}\right]\Bigg) \\
\mathbb{E}_{q^\star}\left[\log q^\star\left(\boldsymbol{\varphi}^2\right)\right] \propto\ & \sum_{j=1}^{p}\left(\log d_j^\star - 2\mathbb{E}_{\varphi^2}\left[\log \varphi_j^2\right] - \frac{1}{2}\mathbb{E}_{\varphi^2}\left[\varphi_j^{-2}\right] - \frac{1}{2s_0^2}\mathbb{E}_{\lambda\cdot\gamma\cdot\varphi^2}\left[\frac{\left(\lambda_j - \mathbf{z}_j^\intercal\boldsymbol{\gamma}\right)^2}{\varphi_j^2}\right]\right) \\
\mathbb{E}_{q^\star}\left[\log q^\star\left(\boldsymbol{\gamma}\right)\right] \propto\ & -\frac{1}{2}\log\left|\boldsymbol{\Sigma}_\gamma^\star\right| - \frac{M}{2s_0^2} \\
\mathbb{E}_{q^\star}\left[\log q^\star\left(\kappa^2\right)\right] \propto\ & \sum_{d=1}^{D}\Bigg(\left(a_d + \frac{m_d}{2}\right)\log f_d^\star - (a_d + \frac{m_d}{2} + 1)\mathbb{E}_{\kappa^2}\left[\log \kappa_d^2\right] - b_d\mathbb{E}_{\kappa^2}\left[\kappa_d^{-2}\right] - \\
& \frac{1}{2}\mathbb{E}_{\gamma\cdot\kappa^2}\left[\frac{\boldsymbol{\gamma}_d^\intercal\boldsymbol{\gamma}_d}{\kappa_d^2}\right]\Bigg) \\
\mathbb{E}_{q^\star}\left[\log q^\star\left(\tau^2\right)\right] \propto\ & \left(\frac{p}{2} + 1\right)\log\left(g^\star\right) - \left(\frac{p}{2} + 2\right)\mathbb{E}_{\tau^2}\left[\log \tau^2\right] - \mathbb{E}_{\tau^2}\left[\frac{1}{\zeta\tau^2}\right] - \\
& \frac{1}{2}\sum_{j=0}^{p}\mathbb{E}_{\beta_0\cdot\beta\cdot\lambda_0^2\cdot\lambda\cdot\tau^2}\left[\frac{\beta_j^2}{\tau^2\lambda_j^2}\right] \\
\mathbb{E}_{q^\star}\left[\log q^\star\left(\zeta\right)\right] \propto\ & \log h^\star - 2\mathbb{E}_{\zeta}\left[\log \zeta\right] - \mathbb{E}_{\zeta}\left[\zeta^{-1}\right] - \mathbb{E}_{\zeta}\left[\frac{1}{\zeta\tau^2}\right],
\end{aligned}
$$

where $\Phi_i = \Phi\left(-\mathbf{x}_i^\intercal\mathbb{E}_\beta\left[\boldsymbol{\beta}\right]\right)$.

Appendix 4.B Computational aspects of the Variational algorithm

The Variational algorithm has the advantage of not requiring the explicit inversion of covariance matrix $\boldsymbol{\Sigma}_\beta^\star = \left(\mathbf{X}^\intercal\mathbf{X} + \boldsymbol{\Delta}\right)^{-1}$, where $\boldsymbol{\Delta} = \mathbb{E}_{\lambda\cdot\tau^2}\left[\tau^{-2}\Lambda^{-2}\right]$ is diagonal, which becomes computationally infeasible when the number of covariates p increases. The quantities needed are the diagonal entries of $\boldsymbol{\Sigma}_\beta^\star$, the posterior mean $\boldsymbol{\mu}_\beta^\star = \boldsymbol{\Sigma}_\beta^\star\left(\mathbf{X}^\intercal\mathbf{y}\right)$ and $tr\left(\mathbf{X}^\intercal\mathbf{X}\boldsymbol{\Sigma}_\beta^\star\right)$. By means of the Woodbury identity, the needed

quantities can be efficiently computed as

$$\begin{aligned}\operatorname{diag}\left(\boldsymbol{\Sigma}_{\beta}^{\star}\right) &= \operatorname{diag}\left(\boldsymbol{\Delta}^{-1}\right) - \operatorname{diag}\left(\boldsymbol{\Delta}^{-1}\mathbf{X}^{\intercal}\left(\mathbf{I}_n + \mathbf{X}\boldsymbol{\Delta}^{-1}\mathbf{X}^{\intercal}\right)^{-1}\mathbf{X}\boldsymbol{\Delta}^{-1}\right) \qquad (4.7)\\ &= \operatorname{diag}\left(\boldsymbol{\Delta}^{-1}\right) - \left[\left(\boldsymbol{\Delta}^{-1}\mathbf{X}^{\intercal}\right)\left(\mathbf{I}_n + \mathbf{X}\boldsymbol{\Delta}^{-1}\mathbf{X}^{\intercal}\right)^{-1} \circ \left(\boldsymbol{\Delta}^{-1}\mathbf{X}^{\intercal}\right)\right] \cdot \mathbf{1}_{n\times 1},\\ \boldsymbol{\mu}_{\beta}^{\star} &= \left(\boldsymbol{\Sigma}_{\beta}^{\star}\mathbf{X}^{\intercal}\right)\mathbf{y} \qquad (4.8)\\ &= \left[\left(\boldsymbol{\Delta}^{-1}\mathbf{X}^{\intercal}\right) - \left(\boldsymbol{\Delta}^{-1}\mathbf{X}^{\intercal}\right)\left(\mathbf{I}_n + \mathbf{X}\boldsymbol{\Delta}^{-1}\mathbf{X}^{\intercal}\right)^{-1}\left(\mathbf{X}\boldsymbol{\Delta}^{-1}\mathbf{X}^{\intercal}\right)\right]\mathbf{y},\\ tr\left(\mathbf{X}\boldsymbol{\Sigma}_{\beta}^{\star}\mathbf{X}^{\intercal}\right) &= \sum_{i=1}^{n}\mathbf{x}_i^{\intercal}\boldsymbol{\Sigma}_{\beta}^{\star}\mathbf{x}_i = \sum_{i=1}^{n}\mathbf{x}_i^{\intercal}\left(\boldsymbol{\Sigma}_{\beta}^{\star}\mathbf{x}_i\right), \qquad (4.9)\end{aligned}$$

where $\circ$ denotes the Hadamard product, $\mathbf{I}_n$ is the $n \times n$ identity matrix and $\mathbf{1}_{n\times 1}$ denotes the n-dimensional vector of ones. In (4.9) the quantity $\boldsymbol{\Sigma}_{\beta}^{\star}\mathbf{x}_i$ has already been computed when evaluating vector $\boldsymbol{\mu}_{\beta}^{\star}$ and all the matrix products in (4.7)-(4.9) can be evaluated with $O\left(n^2 p\right)$ operations, which is linear in p. Note that in the probit regression the last quantity (4.9) is not needed, thus the posterior mean can be efficiently computed as

$$\boldsymbol{\mu}_{\beta}^{\star} = \boldsymbol{\Delta}^{-1}\left(\mathbf{X}^{\intercal}\mathbb{E}_w\left[\mathbf{w}\right]\right) - \left(\boldsymbol{\Delta}^{-1}\mathbf{X}^{\intercal}\right)\left(\mathbf{I}_n + \mathbf{X}\boldsymbol{\Delta}^{-1}\mathbf{X}^{\intercal}\right)^{-1}\left(\mathbf{X}\boldsymbol{\Delta}^{-1}\right)\left(\mathbf{X}^{\intercal}\mathbb{E}_w\left[\mathbf{w}\right]\right),$$

which only involves matrix-vector products, further improving the computational efficiency of the algorithm. Finally, the determinant can be efficiently evaluated as

$$\left|\boldsymbol{\Sigma}_{\beta}^{\star}\right| = \left|\boldsymbol{\Delta}\right| / \left|\mathbf{I}_n + \mathbf{X}\boldsymbol{\Delta}^{-1}\mathbf{X}^{\intercal}\right|.$$

Appendix 4.C Simulation study: Gibbs sampler vs Variational inference

Here we give insights on how co-data information is simulated for the assessment of the Variational approximation accuracy with respect to the Gibbs sampler in Section 4.6. Moreover we provide the graphical representation of the AUC for variable selection and of the computational time for both methods.

The four degrees of co-data information are simulated as:

$G0)$ **no co-data** set-up: we include in the co-data matrix only the intercept, therefore $\mathbf{Z} = \mathbf{1}_p$;

$G1)$ **non-informative** set-up: a binary co-data source is included in the model by randomly selecting 30 regressors, therefore the co-data matrix $\mathbf{Z}$ is created from the binary vector $\mathbf{z} \in \{0,1\}^p$, where $z_j = 1$ if the j-th variable is selected, $z_j = 0$ otherwise;

$G2$) **informative** set-up: a binary co-data source is included in the model by randomly selecting 20 of the true non-zero regressors and 10 of the true zero regressors, therefore the co-data matrix $\mathbf{Z}$ is created from the binary vector $\mathbf{z} \in \{0,1\}^p$, where $z_j = 1$ if the j-th variable is selected, $z_j = 0$ otherwise;

$G3$) **perfect co-data information** set-up: the co-data matrix $\mathbf{Z}$ is created from the binary vector $\mathbf{z} \in \{0,1\}^p$, where $z_j = 1$ if $\beta_j^0 \neq 0$, $z_j = 0$ otherwise.

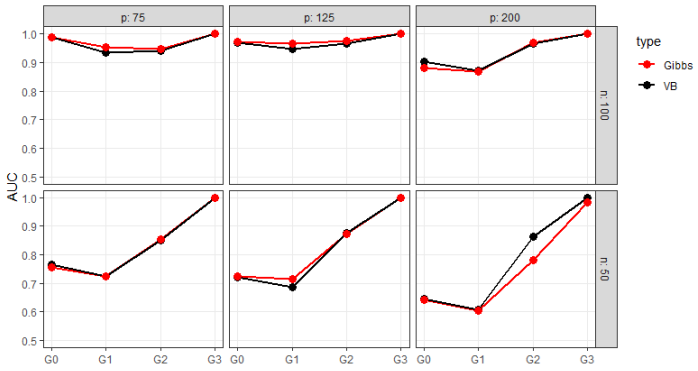


Figure 4.C.1: variable selection with the Gibbs sampler and the Variational algorithm; the AUC is evaluated over 10 replicates of the experiments.

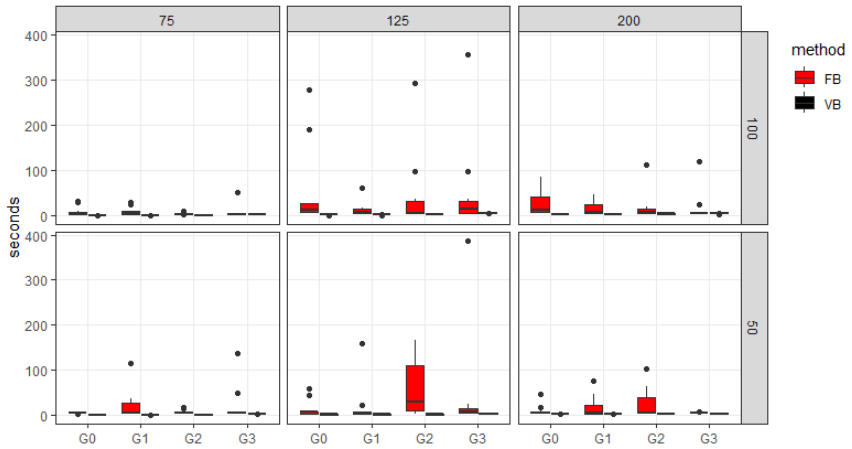


Figure 4.C.2: time in seconds for Gibbs sampler and Variational algorithm for different values of p and n.

Chapter 5

Conclusions and future extensions

This thesis introduces three novel Bayesian approaches for different high-dimensional applications. The main goal of the presented methods is to provide an accurate and efficient way for variable/model selection and prediction when the number of predictors is much higher than the number of observations available, a peculiar feature of modern data. Different applications of MCMC sampling algorithms and Variational approximations for both Bayesian generalized linear models and graphical models have been considered considered. The validity of the introduced methods has been assessed with simulation studies and applications to real datasets, leaving space for additional comments and future extensions. In particular:

- Chapter 2 deals with model selection in high-dimensional linear regression with Gaussian errors and is part of a larger ongoing project. The introduced methods rely on a Delta spike-and-slab prior (George and McCulloch, 1993, 1997). Multiple MCMC schemes for sampling model indicators from the target distribution are discussed, with the final introduction of an adaptive multiple-try algorithm suited for particularly high-dimensional problems. This approach reveals promising results in both simulation studies and real application analysis, with a much improved computational efficiency when compared to other existing full-Bayesian competitors for variable selection. The main drawback is the accuracy when low information is available, i.e. when the number of observation is particularly small. In this case further tuning of the proposal distribution is required, with a possible future application of informed trans-dimensional jumps between models, following Gagnon (2021). At the moment, the main focus is on the analysis of the convergence properties of the described adaptive scheme and the extension of the sampling methods to the case of binary data, which presents a more difficult task in terms of both accuracy of the estimates and computational efficiency. For the latter problem, the updating methods based on the thinQR

can be extended to the update of the inverse matrix $\mathbf{L} = \mathbf{R}^{-1}$, avoiding the explicit inversion of the $p \times p$ covariance matrix when sampling regressor parameter $\boldsymbol{\beta}$. Another consideration to be made concern the important choice of the prior distribution of σ^2, as the analyses show that the results are sensitive with respect to it. To this aim, our project involves the introduction of an auxiliary variable in order to avoid such choice and let the data guide the prior specification. Of course, other prior distributions can be assumed for the residual variance in order to overcome this issue, such as the positive Cauchy distribution that only involves the specification of the scale parameter. A final extension of these approaches could be the inclusion of co-data information by assuming a specific prior inclusion probability for each predictor and regressing these hyperparameters on the co-data following the methodology of Chapter 4, with the goal of improving the overall performances.

- Chapter 3 presents a full-Bayesian approach, called multiple Graphical Horseshoe (mGHS), for the joint analysis of multiple correlated networks, with the goal of improving estimation of similar precision matrices by borrowing strength and sharing sparsity patterns across groups. The proposed method is a direct extension of the graphical Horseshoe of Li et al. (2019) and follows the methodology introduced in Peterson et al. (2020) for multiple precision matrices estimation. The method relies on a novel multivariate shrinkage prior based on the Horseshoe prior (Carvalho et al., 2010) and provides improved edge selection and interpretation of the results as demonstrated through intensive simulation studies and the application to the bike-sharing dataset. The method shows good computational efficiency and scales well with respect to the number of variables, providing one of the fastest full-Bayesian approaches for the estimation of multiple precision matrices. Graphs structure up to a few hundreds of nodes are estimated within few hours. For higher dimensions, a Variational approximation to the joint posterior distribution needs to be implemented. Finally, a novel idea for edge selection based on model cuts (Zigler et al., 2013; Plummer, 2015) is discussed. This represents a basic idea for the estimation of the optimal threshold and is open for further extensions, with additional studies of the convergence properties required.

- Chapter 4 addresses the problem of variable selection and prediction in high-dimensional regression problems when prior information on the covariates is available. Co-data learning guides the estimation process by favoring those predictors that are more informative a-priori. The introduced method relies on an informative version of the Horseshoe regression (Carvalho et al., 2010)

for both continuous and binary outcomes. The model provides a general framework for the flexible inclusion of multiple co-data sources by regressing the hyper-variances on the co-data variables following Van Nee et al. (2021), with the final goal of improving the global performances. A novelty of the method is the joint estimation of the different co-data effects, contrary to the existing method that only admit one (discrete) co-data source and/or estimate the effects separately for each co-data. Both Gibbs sampler and Variational approximation are implemented, with the latter being suited for high-dimensional regression, as the analysis of variable selection and computational efficiency reveals good results when p increases. Future extensions could deal with the inclusion of a dense prior for co-data sources with particularly relevant small groups, forcing the model to learn from such strong prior information.

Bibliography

Ahrens, J. H. and Dieter, U. (1982). Generating Gamma variates by a modified rejection technique. *Commun. ACM*, 25(1):47–54.

Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679.

Andrieu, C. and Moulines, E. (2006). On the ergodicity properties of some adaptive MCMC algorithms. *Ann. Appl. Probab.*, 16(3):1462–1505.

Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4):343–373.

Bai, R., Moran, G. E., Antonelli, J. L., Chen, Y., and Boland, M. R. (2022). Spike-and-slab group Lassos for grouped regression and sparse generalized additive models. *Journal of the American Statistical Association*, 117(537):184–197.

Barbieri, M. M. and Berger, J. O. (2004). Optimal predictive model selection. *The Annals of Statistics*, 32(3):870–897.

Barbieri, M. M., Berger, J. O., George, E. I., and Ročková, V. (2021). The median probability model and correlated variables. *Bayesian Analysis*, 16(4):1085–1112.

Barnard, J., McCulloch, R., and Meng, X.-L. (2000). Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage. *Statistica Sinica*, 10(4):1281–1311.

Bayarri, M. J., Berger, J. O., and Liu, F. (2009). Modularization in Bayesian analysis, with emphasis on analysis of computer models. *Bayesian Analysis*, 4(1):119–150.

Beal, M. J. (2003). *Variational algorithms for approximate Bayesian inference*. University of London, University College London (United Kingdom).

Bellman, R. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton University Press.

Bernardi, M., Casarin, R., Maillet, B., and Petrella, L. (2016). Dynamic model averaging for Bayesian quantile regression.

Bhadra, A., Datta, J., Li, Y., Polson, N., and Willard, B. (2016). Prediction risk for global-local shrinkage regression.

Bhattacharya, A., Chakraborty, A., and Mallick, B. K. (2016). Fast sampling with Gaussian scale-mixture priors in high-dimensional regression. *Biometrika*, 103(4):985–991.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and Machine Learning*, volume 4. Springer.

Biswas, N., Bhattacharya, A., Jacob, P. E., and Johndrow, J. E. (2021). Coupling-based convergence assessment of some Gibbs samplers for high-dimensional Bayesian regression with shrinkage priors. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*.

Biswas, N., Mackey, L., and Meng, X.-L. (2022). Scalable spike-and-slab.

Björck, A. (2015). *Numerical methods in matrix computations*, volume 59 of *Texts in Applied Mathematics*. Springer, Cham.

Blangiardo, M., Hansell, A., and Richardson, S. (2011). A Bayesian model of time activity data to investigate health effect of air pollution in time series studies. *Atmospheric Environment - ATMOS ENVIRON*, 45:379–386.

Blei, D. M. and Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: a review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Boonstra, P. S., Taylor, J. M. G., and Mukherjee, B. (2013). Incorporating auxiliary information for improved prediction in high-dimensional datasets: an ensemble of shrinkage approaches. *Biostatistics (Oxford, England)*, 14(2):259–272.

Bottolo, L., Chadeau-Hyam, M., Hastie, D. I., Langley, S. R., Petretto, E., Tiret, L., Tregouet, D., and Richardson, S. (2011). ESS++: a C++ objected-oriented algorithm for Bayesian stochastic search model exploration. *Bioinformatics*, 27(4):587–588.

Bottolo, L. and Richardson, S. (2010). Evolutionary stochastic search for Bayesian model exploration. *Bayesian Anal.*, 5(3):583–618.

Breheny, P. and Huang, J. (2015). Group descent algorithms for nonconvex penalized linear and logistic regression models with grouped predictors. *Statistics and Computing*, 25(2):173–187.

Brown, P. J., Vannucci, M., and Fearn, T. (1998). Bayesian wavelength selection in multicomponent analysis. *Journal of Chemometrics*, 12(3):173–182.

Brown, P. J., Vannucci, M., and Fearn, T. (2002). Bayes model averaging with selection of regressors. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 64(3):519–536.

Bédard, M., Douc, R., and Moulines, E. (2012). Scaling analysis of multiple-try MCMC methods. *Stochastic Processes and their Applications*, 122(3):758–786.

Carlin, B. P. and Chib, S. (1995). Bayesian model choice via Markov Chain Monte Carlo methods. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(3):473–484.

Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010). The Horseshoe estimator for sparse signals. *Biometrika*, 97:465–480.

Casarin, R., Craiu, R., and Leisen, F. (2013). Interacting multiple-try algorithms with different proposal distributions. *Statistics and Computing*, 23:185–200.

Casella, G. (1985). An introduction to Empirical Bayes data analysis. *The American Statistician*, 39(2):83–87.

Casella, G. and George, E. (1992). Explaining the Gibbs sampler. *The American Statistician*, 46:167–174.

Castelletti, F., La Rocca, L., Peluso, S., Stingo, F. C., and Consonni, G. (2020). Bayesian learning of multiple directed networks from observational data. *Statistics in Medicine*, 39(30):4745–4766.

Chambers, J. M. (1971). Regression updating. *Journal of the American Statistical Association*, 66(336):744–748.

Chandra, N. K., Mueller, P., and Sarkar, A. (2022). Bayesian scalable precision factor analysis for massive sparse Gaussian graphical models.

Chang, H., Lee, C. J., Luo, Z. T., Sang, H., and Zhou, Q. (2022). Rapidly mixing Multiple-try Metropolis algorithms for model selection problems.

Chiang, A. P., Beck, J. S., Yen, H.-J., Tayeh, M. K., Scheetz, T. E., Swiderski, R. E., Nishimura, D. Y., Braun, T. A., Kim, K.-Y. A., Huang, J., Elbedour, K., Carmi, R., Slusarski, D. C., Casavant, T. L., Stone, E. M., and Sheffield,

V. C. (2006). Homozygosity mapping with SNP arrays identifies TRIM32, an E3 ubiquitin ligase, as a Bardet-Biedl syndrome gene (BBS11). *Proceedings of the National Academy of Sciences*, 103(16):6287–6292.

Chipman, H., George, E. I., and McCulloch, R. E. (2001). The practical implementation of Bayesian model selection. In *Model selection*, volume 38 of *IMS Lecture Notes Monogr. Ser.*, pages 65–134.

Clyde, M., Desimone, H., and Parmigiani, G. (1996). Prediction via orthogonalized model mixing. *Journal of the American Statistical Association*, 91(435):1197–1208.

Clyde, M. and George, E. I. (2004). Model uncertainty. *Statist. Sci.*, 19(1):81–94.

Craiu, R. and Lemieux, C. (2007). Acceleration of the multiple-try Metropolis algorithm using antithetic and stratified sampling. *Statistics and Computing*, 17:109–120.

Craiu, R. V., Gray, L., Łatuszyński, K., Madras, N., Roberts, G. O., and Rosenthal, J. S. (2015). Stability of adversarial Markov chains, with an application to adaptive MCMC algorithms. *The Annals of Applied Probability*, 25(6):3592–3623.

Danaher, P. J., Wang, P., and Witten, D. M. (2014). The joint Graphical Lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 76(2):373–397.

De los Campos, G., Naya, H., Gianola, D., Crossa, J., Legarra, A., Manfredi, E., Weigel, K. A., and Cotes, J. M. (2009). Predicting quantitative traits with regression models for dense molecular markers and pedigree. *Genetics*, 182:375–385.

Dellaportas, P., Forster, J. J., and Ntzoufras, I. (2002). On Bayesian model and variable selection using MCMC. *Statistics and Computing*, 12(1):27–36.

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

Dieter, U. (1981). Optimal acceptance-rejection envelopes for sampling from various distributions. *Mathematics of Computation*.

Douc, R., Guillin, A., Marin, J.-M., and Robert, C. P. (2007). Convergence of adaptive mixtures of importance sampling schemes. *The Annals of Statistics*, 35(1):420–448.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32(2):407–499.

Fan, J., Feng, Y., and Wu, Y. (2009). Network exploration via the adaptive Lasso and SCAD penalties. *The Annals of Applied Statistics*, 3(2):521–541.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Fan, Y. and Sisson, S. A. (2011). Reversible jump MCMC. *Handbook of Markov Chain Monte Carlo*, pages 67–92.

Feng, Z. and Li, J. (2015). An adaptive independence sampler MCMC algorithm for infinite dimensional Bayesian inferences. *arXiv: Numerical Analysis*.

Fontaine, S. and Bédard, M. (2022). An adaptive multiple-try Metropolis algorithm. *Bernoulli*, 28(3):1986–2011.

Forte, A., Garcia-Donato, G., and Steel, M. (2018). Methods and tools for Bayesian variable selection and model averaging in normal linear regression. *International Statistical Review*, 86(2):237–258.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the Graphical LASSO. *Biostatistics*, 9(3):432–441.

Gagnon, P. (2021). Informed reversible jump algorithms. *Electronic Journal of Statistics*, 15(2):3951–3995.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6:721–741.

George, E. I. and McCulloch, R. E. (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889.

George, E. I. and McCulloch, R. E. (1997). Approaches for Bayesian variable selection. *Statistica Sinica*, 7(2):339–373.

Geweke, J. (1993). Bayesian treatment of the independent Student-t linear model. *Journal of Applied Econometrics*, 8:S19–S40.

Geweke, J. (1996). Variable selection and model comparison in regression. In *Bayesian statistics, 5 (Alicante, 1994)*, Oxford Sci. Publ., pages 609–620. Oxford Univ. Press, New York.

Gilks, W. R., Roberts, G. O., and Sahu, S. K. (1998). Adaptive Markov Chain Monte Carlo through Regeneration. *Journal of the American Statistical Association*, 93(443):1045–1054.

Giordano, R., Broderick, T., and Jordan, M. (2017). Covariances, robustness, and Variational Bayes. *Journal of Machine Learning Research*, 19.

Golub, G. H. and Van Loan, C. F. (2013). *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, fourth edition.

Graham, R. L., Knuth, D. E., and Patashnik, O. (1994). *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Longman Publishing Co., Inc., USA, 2nd edition.

Green, P. J. (1995). Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732.

Griffin, J. E. and Brown, P. J. (2005). Alternative prior distributions for variable selection with very many more variables than observations. *University of Warwick. Centre for Research in Statistical Methodology*.

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242.

Hahn, P. R. and Carvalho, C. M. (2015). Decoupling shrinkage and selection in Bayesian linear models: A posterior summary perspective. *Journal of the American Statistical Association*, 110(509):435–448.

Hans, C. (2009). Model uncertainty and variable selection in Bayesian Lasso regression. *Statistics and Computing*, 20(2):221–229.

Hans, C. (2011). Elastic net regression modeling with the orthant normal prior. *Journal of the American Statistical Association*, 106(496):1383–1393.

Hans, C., Dobra, A., and West, M. (2007). Shotgun stochastic search for "large p" regression. *J. Amer. Statist. Assoc.*, 102(478):507–516.

Hastie, D. I. and Green, P. J. (2012). Model choice using reversible jump Markov chain Monte Carlo. *Statistica Neerlandica*, 66(3):309–338.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: Data Mining, inference and prediction*. Springer, 2 edition.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov Chains and their applications. *Biometrika*, 57(1):97–109.

Heinze, G., Wallisch, C., and Dunkler, D. (2018). Variable selection - a review and recommendations for the practicing statistician. *Biometrical Journal*, 60(3):431–449.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hsiang, T. C. (1975). A Bayesian view on ridge regression. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 24(4):267–268.

Hu, F. and Zidek, J. V. (2002). The weighted likelihood. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 30(3):347–371.

Ishwaran, H. and Rao, J. S. (2005). Spike and slab variable selection: Frequentist and Bayesian strategies. *The Annals of Statistics*, 33(2):730–773.

Ji, C. and Schmidler, S. C. (2013). Adaptive Markov Chain Monte Carlo for Bayesian variable selection. *Journal of Computational and Graphical Statistics*, 22(3):708–728.

Johndrow, J. E., Orenstein, P., and Bhattacharya, A. (2020). Scalable approximate MCMC algorithms for the Horseshoe prior. *Journal of Machine Learning Research*, 21:73:1–73:61.

Johnson, V. E. (2013). On numerical aspects of Bayesian model selection in high and ultrahigh-dimensional settings. *Bayesian analysis (Online)*, 8(4):741.

Johnson, V. E. and Rossell, D. (2012). Bayesian model selection in high-dimensional settings. *J. Amer. Statist. Assoc.*, 107(498):649–660.

Jordan, M., Ghahramani, Z., Jaakkola, T., and Saul, L. (1999). An introduction to Variational methods for graphical models. *Machine Learning*, 37:183–233.

Kpogbezan, G., van de Wiel, M., van Wieringen, W., and van der Vaart, A. (2019). Incorporating prior information and borrowing information in high-dimensional sparse regression using the horseshoe and Variational Bayes.

Kyung, M., Gill, J., Ghosh, M., and Casella, G. (2010). Penalized regression, standard errors, and Bayesian lassos. *Bayesian Analysis*, 5:369–411.

Lamnisos, D., Griffin, J. E., and Steel, M. F. J. (2009). Transdimensional sampling algorithms for Bayesian variable selection in classification problems with many more variables than observations. *Journal of Computational and Graphical Statistics*, 18(3):592–612.

Lappalainen, T., Sammeth, M., Friedländer, M., Hoen, P., Monlong, J., Rivas, M., Gonzàlez Porta, M., Kurbatova, N., Griebel, T., Ferreira, P., Barann, M., Wieland, T., Greger, L., Iterson, M., Almlöf, J., Ribeca, P., Pulyakhina, I., Esser, D., Giger, T., and Dermitzakis, E. (2013). Transcriptome and genome sequencing uncovers functional variation in humans. *Nature*, 501.

Leamer, E. E. (1978). Regression selection strategies and revealed priors. *J. Amer. Statist. Assoc.*, 73(363):580–587.

Lee, J., Hussain, S., Warnick, R., Vannucci, M., Menchaca, I., Seitz, A. R., Hu, X., Peters, M. A. K., and Guindani, M. (2023). A predictor-informed multi-subject Bayesian approach for dynamic functional connectivity.

Lee, S. Y. (2022). Gibbs sampler and Coordinate Ascent Variational Inference: A set-theoretical review. *Communications in Statistics - Theory and Methods*, 51(6):1549–1568.

Li, Q. and Lin, N. (2010). The Bayesian elastic net. *Bayesian Analysis*, 5:151–170.

Li, Y., Craig, B. A., and Bhadra, A. (2019). The Graphical Horseshoe estimator for inverse covariance matrices. *Journal of Computational and Graphical Statistics*, 28(3):747–757.

Liang, F., Truong, Y. K., and Wong, W. H. (2001). Automatic Bayesian model averaging for linear regression and applications in Bayesian curve fitting. *Statist. Sinica*, 11(4):1005–1029.

Lingjaerde, C., Fairfax, B. P., Richardson, S., and Ruffieux, H. (2022). Scalable multiple network inference with the joint graphical horseshoe. *arXiv:2206.11820*.

Liu, J. S., Liang, F., and Wong, W. H. (2000). The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134.

Liu, X. and Daniels, M. J. (2006). A new algorithm for simulating a correlation matrix based on parameter expansion and reparameterization. *Journal of Computational and Graphical Statistics*, 15:897–914.

Lunn, D., Best, N., Spiegelhalter, D., Graham, G., and Neuenschwander, B. (2009). Combining MCMC with 'sequential' PKPD modelling. *Journal of pharmacokinetics and pharmacodynamics*, 36:19–38.

MacKay, D. J., Mac Kay, D. J., et al. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Madigan, D., York, J., and Allard, D. (1995). Bayesian graphical models for discrete data. *International Statistical Review / Revue Internationale de Statistique*, 63(2):215–232.

Maire, F., Friel, N., Mira, A., and Raftery, A. E. (2019). Adaptive incremental mixture Markov Chain Monte Carlo. *Journal of Computational and Graphical Statistics*, 28(4):790–805.

Makalic, E. and Schmidt, D. F. (2016). A simple sampler for the Horseshoe estimator. *IEEE Signal Processing Letters*, 23(1):179–182.

Martino, L. (2018). A review of multiple-try MCMC algorithms for signal processing. *Digital Signal Processing*, 75:134–152.

Martino, L., Del Olmo, V. P., and Read, J. (2012). A multi-point Metropolis scheme with generic weight functions. *Statistics & Probability Letters*, 82(7):1445–1453.

Martino, L. and Louzada, F. (2017). Issues in the multiple try metropolis mixing. *Computational Statistics*.

McCandless, L. C., Douglas, I. J., Evans, S. J., and Smeeth, L. (2010). Cutting feedback in Bayesian regression adjustment for the propensity score. *The International Journal of Biostatistics*, 6(2).

Meuwissen, T. H. E., Hayes, B. J., and Goddard, M. E. (2001). Prediction of total genetic value using genome-wide dense marker maps. *Genetics*, 157(4):1819–1829.

Meyn, S. and Tweedie, R. (1993). *Markov Chains and Stochastic Stability*. Springer, London.

Mitchell, T. J. and Beauchamp, J. J. (1988). Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032.

Montesinos López, O. A., Montesinos López, A., and Crossa, J. (2022). *Bayesian Genomic Linear Regression*, pages 171–208. Springer International Publishing.

Muller, P., Parmigiani, G., and Rice, K. (2007). FDR and Bayesian multiple comparisons rules. *Bayesian Statistics 8*.

Münch, M., Peeters, C., van der Vaart, A., and van de Wiel, M. (2019). Adaptive group-regularized logistic elastic net regression. *Biostatistics (Oxford, England)*, 22.

Narisetty, N. N. (2020). Bayesian model selection for high-dimensional data. In *Principles and methods for data science*, volume 43 of *Handbook of Statist.*, pages 207–248. Elsevier/North-Holland, Amsterdam.

Narisetty, N. N. and He, X. (2014). Bayesian variable selection with shrinking and diffusing priors. *Ann. Statist.*, 42(2):789–817.

Neuenschwander, B., Roychoudhury, S., and Schmidli, H. (2016). On the use of co-data in clinical trials. *Statist. Biopharm. Res.*, 8(3):345–354.

Ni, Y., Baladandayuthapani, V., Vannucci, M., and Stingo, F. (2022). Bayesian graphical models for modern biological applications (with discussion). *Statistical Methods and Applications*, 31:197–225.

Nocedal, J. and Wright, S. J. (2006). *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition.

O'Hara, R. B. and Sillanpää, M. J. (2009). A review of bayesian variable selection methods: what, how and which. *Bayesian Anal.*, 4(1):85–117.

Pandolfi, S., Bartolucci, F., and Friel, N. (2010). A generalization of the multiple-try Metropolis algorithm for Bayesian estimation and model selection. *Journal of Machine Learning Research - Proceedings Track*, 9:581–588.

Pandolfi, S., Bartolucci, F., and Friel, N. (2014). A generalized multiple-try version of the reversible jump algorithm. *Computational Statistics & Data Analysis*, 72:298–314.

Papathomas, M., Dellaportas, P., and Vasdekis, V. G. S. (2011). A novel reversible jump algorithm for generalized linear models. *Biometrika*, 98(1):231–236.

Park, T. and Casella, G. (2008). The Bayesian Lasso. *Journal of the American Statistical Association*, 103(482):681–686.

Peterson, C., Stingo, F., and Vannucci, M. (2015). Bayesian inference of multiple Gaussian graphical models. *Journal of the American Statistical Association*, 110(509):159–174.

Peterson, C. B., Osborne, N., Stingo, F. C., Bourgeat, P., Doecke, J. D., and Vannucci, M. (2020). Bayesian modeling of multiple structural connectivity networks during the progression of alzheimer's disease. *Biometrics*, 76(4):1120–1132.

Petralias, A. and Dellaportas, P. (2013). An MCMC model search algorithm for regression problems. *Journal of Statistical Computation and Simulation*, 83(9):1722–1740.

Piironen, J. and Vehtari, A. (2017). Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11.

Plummer, M. (2015). Cuts in Bayesian graphical models. *Statistics and Computing*, page 37–43.

Polson, N. G. and Scott, J. G. (2011). Shrink globally, act locally: sparse Bayesian regularization and prediction. In *Bayesian Statistics 9*, pages 501–538. Oxford University Press.

Polson, N. G., Scott, J. G., and Windle, J. B. (2011). The Bayesian bridge. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76:713–733.

Polson, N. G., Scott, J. G., and Windle, J. B. (2013). Bayesian inference for logistic models using Pólya–Gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349.

Pourahmadi, M. (2011). Covariance estimation: the GLM and regularization perspectives. *Statistical Science*, 26(3):369–387.

Raftery, A. E., Madigan, D., and Hoeting, J. A. (1997). Bayesian model averaging for linear regression models. *J. Amer. Statist. Assoc.*, 92(437):179–191.

Raudys, S. and Jain, A. (1991). Small sample size effects in statistical pattern recognition: recommendations for practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):252–264.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. 2nd ed. Springer, New York.

Roberts, G. and Smith, A. (1994). Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms. *Stochastic Processes and their Applications*, 49(2):207–216.

Roberts, G. O. and Polson, N. G. (1994). On the geometric convergence of the Gibbs sampler. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(2):377–384.

Roberts, G. O. and Rosenthal, J. S. (2007). Coupling and ergodicity of adaptive Markov Chain Monte Carlo algorithms. *Journal of Applied Probability*, 44(2):458–475.

Ročková, V. and George, E. I. (2014). EMVS: the EM approach to Bayesian variable selection. *Journal of the American Statistical Association*, 109(506):828–846.

Ročková, V. and George, E. I. (2018). The spike-and-slab Lasso. *J. Amer. Statist. Assoc.*, 113(521):431–444.

Saeys, Y., Inza, I., and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517.

Salimans, T., Kingma, D. P., and Welling, M. (2015). Markov Chain Monte Carlo and Variational Inference: bridging the gap. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1218–1226.

Scheetz, T. E., Kim, K.-Y. A., Swiderski, R. E., Philp, A. R., Braun, T. A., Knudtson, K. L., Dorrance, A. M., DiBona, G. F., Huang, J., Casavant, T. L., Sheffield, V. C., and Stone, E. M. (2006). Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, 103(39):14429–14434.

Scott, J. G. and Berger, J. O. (2010). Bayes and empirical-Bayes multiplicity adjustment in the variable-selection problem. *Ann. Statist.*, 38(5):2587–2619.

Segura, J. (2021). Uniform (very) sharp bounds for ratios of Parabolic Cylinder functions. *Studies in Applied Mathematics*, 147.

Shaddox, E., Stingo, F., Peterson, C., Jacobson, S., Cruickshank-Quinn, C., Kechris, K., Bowler, R., and Vannucci, M. (2018). A Bayesian approach for learning gene networks underlying disease severity in COPD. *Statistics in Biosciences*, 10(1):59–85.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group Lasso. *J Comput Graph Stat*, 22(2):231–245.

Smith, M. and Kohn, R. (1996). Nonparametric regression using Bayesian variable selection. *Journal of Econometrics*, 75(2):317–343.

Stadlober, E. (1982). *Generating Student's T Variates by a Modified Rejection Method*, pages 349–360. Springer Netherlands.

Tadesse, M. G. and Vannucci, M. (2021). Handbook of bayesian variable selection.

Tai, F. and Pan, W. (2007). Incorporating prior knowledge of predictors into penalized classifiers with multiple penalty terms. *Bioinformatics*, 23(14):1775–1782.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Tipping, M. (2001). Sparse Bayesian learning and relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244.

Turner, R. E. and Sahani, M. (2011). Two problems with Variational expectation maximisation for time-series models. In *Bayesian Time series models*, pages 109–130. Cambridge University Press.

Van de Wiel, M. A., Lien, T. G., Verlaat, W., van Wieringen, W. N., and Wilting, S. M. (2016). Better prediction by use of co-data: adaptive group-regularized ridge regression. *Statistics in medicine*, 35(3):368–381.

Van de Wiel, M. A., Te Beest, D. E., and Münch, M. M. (2019). Learning from a lot: Empirical Bayes for high-dimensional model-based prediction. *Scandinavian Journal of Statistics*, 46(1):2–25.

Van der Pas, S., Kleijn, B. J., and van der Vaart, A. (2014). The Horseshoe estimator: Posterior concentration around nearly black vectors. *Electronic Journal of Statistics*, 8.

Van der Pas, S., Szabó, B., and van der Vaart, A. (2017). Uncertainty quantification for the horseshoe (with discussion). *Bayesian Analysis*, 12(4):1221–1274.

Van Erp, S., Oberski, D. L., and Mulder, J. (2019). Shrinkage priors for Bayesian penalized regression. *Journal of Mathematical Psychology*, 89:31–50.

Van Nee, M. M., Wessels, L. F., and van de Wiel, M. A. (2021). Flexible co-data learning for high-dimensional prediction. *Statistics in Medicine*, 40(26):5910–5925.

Velten, B. and Huber, W. (2019). Adaptive penalization in high-dimensional regression and classification with external covariates using Variational Bayes. *Biostatistics*, 22(2):348–364.

Verlaat, W., Snoek, B. C., Heideman, D. A., Wilting, S. M., Snijders, P. J., Novianti, P. W., van Splunter, A. P., Peeters, C. F., van Trommel, N. E., Massuger, L. F., et al. (2018). Identification and validation of a 3-gene methylation classifier for hpv-based cervical screening on self-samples. *Clinical Cancer Research*, 24(14):3456–3464.

Wang, B. and Titterington, D. M. (2005). Inadequacy of interval estimates corresponding to variational Bayesian approximations. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, pages 373–380. PMLR.

Wang, H. (2012). Bayesian Graphical LASSO models and efficient posterior computation. *Bayesian Analysis*, 7.

Wang, H. (2015). Scaling it up: Stochastic search structure learning in graphical models. *Bayesian Analysis*, 10(2):351–377.

Yang, J., Levi, E., Craiu, R. V., and Rosenthal, J. S. (2019). Adaptive componentwise multiple-try Metropolis sampling. *Journal of Computational and Graphical Statistics*, 28(2):276–289.

Yang, X., Gan, L., Narisetty, N., and Liang, F. (2021). Gembag: Group estimation of multiple Bayesian graphical models. *Journal of Machine Learning Research*, 22.

Yeo, I.-K. and Johnson, R. A. (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4):954–959.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B*, 68:49–67.

Zellner, A. (1986). On assessing prior distributions and Bayesian regression analysis with g-prior distributions. In *Bayesian inference and decision techniques*, volume 6 of *Stud. Bayesian Econometrics Statist.*, pages 233–243. North-Holland, Amsterdam.

Zhu, Y. and Foygel Barber, R. (2015). The log-shift penalty for adaptive estimation of multiple Gaussian Graphical models. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pages 1153–1161.

Zigler, C. (2016). The central role of Bayes theorem for joint estimation of causal effects and Propensity Scores. *The American Statistician*, 70:47–54.

Zigler, C., Watts, K., Yeh, R., Wang, Y., Coull, B., and Dominici, F. (2013). Model feedback in Bayesian Propensity Score estimation. *Biometrics*, 69.

Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the Elastic Net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(2):301–320.