Acknowledgements

First of all, I express my gratitude to my supervisor, Professor Giacomo Zanella. Without his patience in providing me guidance throughout every step of the process, I would never have gained the many insights that led to the completion of this thesis. I would also like to extend my gratitude to Professor Antonio Lijoi, who together with Professor Zanella provided me with selfless and inspiring mentorship during my participation to the Visiting Student Initiative at Bocconi University. Taking part in the many activities organized by the Department of Decision Sciences has been a once-in-a-lifetime learning experience, that made me feel part of a community driven by their genuine appreciation and dedication to science.

I profoundly thank my parents and grandparents for their unconditional love and support, that made this journey possible. I also wish to express gratitude to the many amazing friends that made these last two years some of the most enjoyable and lighthearted in my life, while always being there to help and support each other.

Thank you,
Giosuè

Contents

Chapter 1

Introduction

In many application domains of statistics, datasets in which features outnumber the observations have become ubiquitous. For this reason, recent advancements in the broad field of statistics have focused on this particular situation by devising new methods that are robust to a large number of features. The field of *high-dimensional statistics* encompasses research efforts aimed at solving problems arising in this setting. Classical methods often fail in high-dimensional settings, and the problem is circumvented by applying techniques that filter out irrelevant features. This is the case of linear models fit by least-squares, as the problem of defining uniquely the regression coefficients and the related standard errors becomes ill-posed when the number of predictors is larger than the sample size (Bühlmann and Van De Geer, 2011). Fitting linear models in this setting requires *variable selection*, which can be defined as the process of identifying a subset of relevant predictors to be included in the model (Tadesse and Vannucci, 2021).

Foundational techniques in the domain of frequentist statistics have been established, often by selecting a priori a subset of variables according to an objective criterion (see the review by Miller, 2002) or by adjusting the loss function with a penalty term that depends on the dimensionality of the model. Prominent examples in this sense include the Lasso (Tibshirani, 1996), Ridge regression (Hoerl and Kennard, 1970), and other penalized likelihood methods (Fan and Li, 2001). Bayesian statistics has addressed the problem by embedding in the hierarchical model latent variables that determine the inclusion of each predictor in the regression model. The approach is particularly attractive, as it provides a quantification of the uncertainty related to the choice of including each variable by means of posterior inclusion probabilities. The preference for parsimonious models can then be expressed by specifying a prior over the variables that determine the sparsity of the model. Foundational approaches in this setting have been established by Mitchell and Beauchamp (1988) with the usage spike-and-slab priors, that have later been developed in their continuous version (George and McCulloch, 1997). Other lines of research focused on shrinkage priors, that implicitly introduce a penalty term in a similar fashion to penalized regression methods, and it has been shown that both Lasso (Park and Casella, 2008) and Ridge can be interpreted as pertaining to this class of models. A recent effort aimed at bridging the gaps between the frequentist and the Bayesian approaches while borrowing strengths from both is the spike-and-slab Lasso by Ročková and George (2018).

While the Bayesian approach presents attractive features, the computational hurdle required to compute the posterior in such large parametric spaces necessitates efficient computational methods. Foundational work on stochastic search variable selection by George and McCulloch (1993) deployed Gibbs sampling techniques to search for promising models, while the reversible jump MCMC approach in Green (1995) involved the usage of a Metropolis-Hastings algorithm. A typical challenge for the Gibbs sampler in this setting arises when the features are strongly correlated, as this significantly harms the ability of the algorithm to reach convergence. In high-dimensional settings, the problem is made even worse by the large number of parameters, and strong computational advantages can be obtained by ensuring that the sampler focuses on *promising* variables with high inclusion probability. The tempered Gibbs sampler (TGS) developed by Zanella and Roberts (2019) ensures an informed choice of the variables to update and mitigates the correlation between variables by considering a *flattened* version of the posterior distribution. A variation of the method, termed weighted TGS (wTGS), enables the algorithm to focus on the variables with higher posterior inclusion probability while ensuring irreducibility and, in turn, convergence to the posterior.

The promising features of the algorithm motivated the research questions that the thesis addresses, tackling aspects related to the practical implementation of the tempered Gibbs samplers considering them in the context of Bayesian variable selection. Furthermore, simulations in a new challenging setting are analyzed, enabling a direct comparison with state-of-the-art methods. New directions for future research are proposed and introduced, paving the way for future developments of the algorithms.

The thesis is structured as follows. Chapter 2 provides a wide-encompassing literature review of Bayesian inference and computational methods, introducing the most important tools used in Bayesian variable selection as well as foundational results. The review proceeds with Chapter 3, that studies model selection focusing on the Bayesian variable selection problem, with a particular emphasis on the computational challenges related to high-dimensional settings. The attention is then directed to the Gibbs sampler and how it can be leveraged in the context of Bayesian variable selection, underlining its criticalities and how they can be addressed by deploying the tempered Gibbs sampling techniques. Simulations highlighting the characteristics of the algorithms and providing a comparison on several metrics are then presented in Chapter 4, and Chapter 5 focuses on future directions of work that can be waged for evolving wTGS into an algorithm that deploys continuous-time Markov chains. Finally, basic results from numerical linear algebra are deployed in the Appendix A to propose alternative implementations of the methods.

Chapter 2

Bayesian inference

Throughout the following chapters, probabilistic models and simulation techniques are implemented through the Bayesian framework. This perspective is concerned with the very definition of probability. While classical interpretations of probability define it as the limiting frequency of random and repeatable events, the Bayesian approach postulates a *subjective* interpretation, and probability provides a quantification of uncertainty. Unknown quantities, such as parameters indexing a statistical model for the data, are considered random variables. Consequently, a joint probability distribution is constructed for the data and the parameters. This can naturally be expressed with the specification of a probability distribution for the parameters, called *prior distribution*, and a conditional distribution of the data given the parameters, called *likelihood*. It should be noted that only the observed data is actually considered, as the likelihood is a function of the parameters, and the uncertainty in the parameters is expressed by a probability distribution. By contrast, a *frequentist* interpretation would consider the parameters as fixed and build confidence intervals that depend entirely on the distribution of possible datasets.

2.1 Bayesian parametric models

For the following chapters, the analysis will focus on *parametric* models. For the sake of completeness, a formal probabilistic definition of such objects is found in Schervish (1995).

Definition 2.1.1. Let $(S, \mathcal{A}, \mu)$ be a probability space, let $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ and $(\Omega, \mathcal{B}(\Omega))$ be measurable spaces equipped with a Borel σ-algebra. Let $X : S \to \mathcal{X}$ and $\Theta : S \to \Omega$ be measurable. Then, Θ is a parameter and Ω a parameter space. The conditional distribution for X given Θ is called the *parametric family of distributions of X*. It is denoted by

$$\mathcal{M} = \{P_\theta : \forall A \in \mathcal{B}(\mathcal{X}), P_\theta(A) = \mathrm{P}(X \in A|\Theta = \theta), \text{ for } \theta \in \Omega\}.$$

For P_θ being a probability measure on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ that is absolutely continuous with respect to a measure ν on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, the Radon-Nykodim derivative is denoted

$$\frac{dP_\theta}{d\nu}(x) = p(x|\theta),$$

and it is commonly referred to as likelihood function. Next, the main ingredients of Bayesian inference are introduced. The prior distribution of Θ is the probability measure μ_Θ over $(\Omega, \mathcal{B}(\Omega))$ that is induced by Θ from μ. The marginal distribution of X is denoted as μ_X, and it follows that, using Fubini-Tonelli, we can express it for some $A \in \mathcal{B}(\mathcal{X})$ as

$$\mu_X(A) = \int_\Omega \int_A p(x|\theta) d\nu(x) d\mu_\Theta(\theta) = \int_A \int_\Omega p(x|\theta) d\mu_\Theta(\theta) d\nu(x),$$

that shows μ_X to be absolutely continuous with respect to ν, with the expression of the *marginal density* of X being

$$p(x) = \int_\Omega p(x|\theta) d\mu_\Theta(\theta).$$

Let $\mathcal{X} \times \Omega$ denote the sample space, then the joint distribution on this space is determined by the prior distribution and the statistical model $\{P_\theta : \theta \in \Omega\}$. Assuming absolute continuity of the prior distribution and the model, then a set $B \subseteq \mathcal{X} \times \Omega$ has probability

$$\mathrm{P}((X, \Theta) \in B) = \int \int \mathbb{1}_B(x, \theta) p(x|\theta) p(\theta) dx d\theta,$$

for $p(\theta)$ being the prior density.

One of the most attractive features of Bayesian inference is its principled way to update probability distributions as new data comes available. Information is encoded through the conditional distribution of Θ given the data $X = x$, and we denote it as $\mu_{\Theta|X}$. This distribution is called *posterior distribution*, and its computation follows from the *Bayes' theorem* under the assumption of P_θ being dominated by some measure ν.

Theorem 1 (Bayes' theorem)**.** Let X have a parametric family $\mathcal{P}_0$ of distributions with parameter space Ω. Suppose $P_\theta \ll \nu$ for all $\theta \in \Omega$ with conditional density $p(x|\theta)$. Let μ_Θ be the prior of Θ, and $\mu_{\Theta|X}$ be the posterior distribution of Θ given X.
Then, $\mu_{\Theta|X} \ll \mu_\Theta$ and the Radon-Nykodim derivative is

$$\frac{\mu_{\Theta|X}}{\mu_\Theta}(\theta|x) = \frac{p(x|\theta)}{\int_\Omega p(x|\theta) d\mu_\Theta(\theta)}$$

for every x such that the dominator is neither 0 nor infinite. For every other x, the posterior can be defined arbitrarily.

A proof can be found in Schervish (1995).

Besides being used to make direct inferences about Θ, the posterior distribution is deployed for the task of prediction of future observations. Let $X_1 = x_1, \ldots, X_n = x_n$ be the data at our disposal, and let them be conditionally independent given Θ. Then, the *predictive density* of future observations $X_{n+1}, \ldots, X_{n+k}$ is

$$p(x_{n+1}, \ldots, x_{n+k}|x_1, \ldots, x_n) = \int_\Omega \prod_{i=1}^{k} p(x_{n+i}|\theta) d\mu_{\Theta|X_1,\ldots,X_n}(\theta|x_1, \ldots, x_n).$$

2.1.1 Exchangeability

In the precedent expression, the result relied on an assumption of conditional independence. This assumption, replacing the existence of a "fixed and unknown θ" and the stronger frequentist assumption of independent data, implies a notion of symmetry in the data generating process known as *exchangeability*. It should first be noted that, as suggested by Bernardo and Smith (1994), no learning from experience takes place within a sequence of independent data. It is apparent that in such a case, for any $1 \leq m < n$

$$p(x_{m+1}, \ldots, x_n|x_1, \ldots, x_{m-1}) = p(x_{m+1}, \ldots, x_n).$$

When this is not the case, some form of dependence must be assumed. When the labels specifying the order of the data are uninformative, and therefore the joint distribution of the data is invariant to permutations, the data are exchangeable.

Definition 2.1.2 (Exchangeability)**.** A finite set $X_1, \ldots, X_n$ of random variables is said to be exchangeable under a probability measure P if the joint distribution satisfies

$$P(X_1, \ldots, X_n) = \mathrm{P}(X_{\pi(1)}, \ldots, X_{\pi(n)})$$

for every permutation π defined on the set $\{1, \ldots, n\}$. An infinite sequence is *infinitely exchangeable* if every finite subsequence is exchangeable.

From a practical standpoint, the information obtained from any data point is no more relevant than any other, regardless of the position in the sequence of observations (Bernardo and Smith, 1994). The notion of exchangeability comes in handy in providing a theoretical justification for the usage of the main tools of Bayesian inference. This result is De Finetti's representation theorem, and the version presented below can be found in Schervish (1995).

Theorem 2 (De Finetti's representation theorem for Bernoulli models)**.** An infinite sequence $\{X_n\}_{n=1}^{\infty}$ of Bernoulli random variables is exchangeable if and only if there is a random variable Θ with distribution Q and taking values in $[0, 1]$ such that the joint probability function of every finite subsequence $X_1 = x_1, \dots, X_n = x_n$ is

$$P(x_1, \dots, x_n) = \int_0^1 \prod_{i=1}^{n} \theta^{x_i}(1-\theta)^{1-x_i} dQ(\theta).$$

Furthermore, if the sequence is exchangeable, then the distribution Q is unique and

$$\frac{1}{n}\sum_{i=1}^{n} X_i \to \Theta \quad \text{almost surely.}$$

The learning process is represented by updates of the prior, via Bayes' theorem, into the posterior distribution. Operationally, this update is used to define the form of the *posterior predictive distribution* (Bernardo and Smith, 1994).

Corollary 1. For $X_1, X_2, \dots$ defined as in Theorem 2, the posterior predictive distribution has form

$$P(x_{m+1}, \dots, x_n | x_1, \dots, x_m) = \int_0^1 \prod_{i=m+1}^{n} \theta^{x_i}(1-\theta)^{1-x_i} dQ(\theta | x_1, \dots, x_m), \quad 1 \leq m < n$$

where

$$dQ(\theta | x_1, \dots, x_m) = \frac{\prod_{i=1}^{m} \theta^{x_i}(1-\theta)^{1-x_i} dQ(\theta)}{\int_0^1 \prod_{i=1}^{m} \theta^{x_i}(1-\theta)^{1-x_i} dQ(\theta)}$$

The representation theorem defined above can be extended over the case of Bernoulli-distributed data to any infinitely exchangeable sequence of random variables in $\mathbb{R}$ with probability measure P. The result is the most general form of De Finetti's representation theorem, stated below following Bernardo and Smith (1994).

Theorem 3 (General representation theorem)**.** Let $X_1, X_2 \dots$ be an infinitely exchangeable sequence of random variables, taking values in $\mathbb{R}$ and with probability measure P. Then, there exists a probability measure Q over the space $\mathcal{F}$ of distribution functions in $\mathbb{R}$, such that the joint probability function of every finite subsequence $X_1 = x_1, \dots, X_n = x_n$ is

$$P(x_1, \dots, x_n) = \int_{\mathcal{F}} \prod_{i=1}^{n} F(x_i) dQ(F)$$

where, for F_n being the empirical distribution function defined by $x_1, \dots, x_n$,

$$Q(F) = \lim_{n \to \infty} P(F_n).$$

A proof is provided in Chow et al. (1988).

2.2 Bayesian computation

For $\mu_{\Theta|X}$ being the posterior distribution, a statistician might be interested in computing the posterior mean for some functional h, denoted as $\mu_{\Theta|X}(h)$. Resorting to an unnormalized version of the posterior distribution, that is

$$\mu^u_{\Theta|X}(\theta|x) \equiv p(x|\theta)\mu_\Theta(\theta) \propto \mu_{\Theta|X}(\theta|x),$$

we can compute the posterior mean by solving the integral

$$\mu_{\Theta|X}(h) = \mathbb{E}_{\Theta|X}[h(\Theta)] = \frac{\int_\Omega h(\theta) d\mu^u_{\Theta|X}(\theta|x)}{\int_\Omega d\mu^u_{\Theta|X}(\theta|x)}$$

An analytical solution is typically unfeasible in situations where Ω is high-dimensional. The problem of computing integrals from probability distributions with untractable analytic expression is typically solved by numerical methods. A sample $X_1, \dots, X_N$ from the posterior is used to compute the *Monte Carlo estimator* of the integral.

$$\hat{\mu}_{\Theta|X}(h) = \frac{1}{N}\sum_{n=1}^{N} h(X_n)$$

Deterministic and stochastic algorithms have been devised to simulate such samples, and what follows is a brief overview of two of the most popular techniques: importance sampling (IS) and Markov chain Monte Carlo (MCMC).

2.2.1 Importance sampling

When direct simulation from the posterior is unfeasible, importance sampling allows using samples from another distribution to approximate an integral with respect to the posterior. A comprehensive review can be found in Ch.8 of Chopin et al. (2020). Let G be a distribution we can easily sample from, and such that $\mu_{\Theta|X} \ll G$. For $f(\cdot)$ being the posterior density and $g(\cdot)$ the density of G, termed *proposal density*, the method deploys the identity:

$$\mu_{\Theta|X}(h) = \int_{\Omega} h(x)f(x)dx = \int_{\Omega} h(x)\frac{f(x)}{g(x)}g(x)dx.$$

For $w(x) = f(x)/g(x)$ being the IS weights, the Monte Carlo estimator is then obtained by taking

$$\hat{\mu}_{\Theta|X}(h) = \frac{1}{N}\sum_{n=1}^{N} w(X_n)h(X_n), \quad X_n \sim G. \tag{2.1}$$

A criticality of IS is its poor scalability with respect to the dimensionality of the target. The issue becomes apparent when one targets the joint distribution $f(\cdot)$ of a random vector $\boldsymbol{X} \in \mathbb{R}^d$ with components that are independent both under f and under the proposal q. Then, the IS (normalized) weight for a sample is $w(\boldsymbol{X}) = \prod_{i=1}^{d} w(X_i)$ and its variance under the proposal is

$$\mathrm{Var}_q(w(\boldsymbol{X})) = \mathbb{E}_q[w(\boldsymbol{X})^2] - 1 = \prod_{i=1}^{d}(1 + \mathrm{Var}_q(w_i(X_i)).$$

Clearly, variance grows exponentially with respect to d unless it decreases rapidly as d grows. This problem is typically referred to as the *curse of dimensionality* of IS.

It is often the case that evaluating the density point-wise is unfeasible, and it is only possible up to a normalizing constant. In this case, a density $f^u(\cdot) \propto f(\cdot)$ is used to compute the unnormalized IS weight $w^u(x) = \frac{f^u(x)}{g(x)}$. The method is termed *self-normalized importance sampling* (SIS), and the Monte Carlo estimator is then:

$$\hat{\mu}_{\Theta|X}(h) = \frac{\sum_{n=1}^{N} w^u(X_n) h(X_n)}{\sum_{n=1}^{N} w^u(X_n)}, \quad X_i \sim G$$

2.2.2 Markov Chain Monte Carlo

One of the most widely used classes of algorithms for posterior computation in Bayesian inference is MCMC. A review of these techniques in general state spaces can be found in Tierney (1994), Robert and Casella (2004), and Roberts and Rosenthal (2004). Results will be presented for general state spaces, but, as infinite state spaces are beyond the scope of this review, some results are explicitly stated for finite state spaces. A reference for MCMC in finite state spaces is Levin and Peres (2017).

MCMC algorithms obtain a sample from the posterior distribution by simulating a Markov chain with transition kernel $P(x, dy)$, for $x, y \in \Omega$. Sufficient conditions must be established to guarantee that the chain converges asymptotically to the posterior of interest. If simulation from such a kernel is possible, an estimate of $\mu_{\Theta|X}(f)$ is then obtained from the sample $X_0, \ldots, X_N \sim \mu_{\Theta|X}(\cdot)$ by computing the *ergodic average*, corresponding to the Monte Carlo estimator in 2.1. The posterior target will henceforth be referred to as π to improve readability.

The first property such a chain must possess is referred to as *invariance* or *stationarity* with respect to the posterior π, meaning that if a sample X_0 is drawn from π, every other X_t obtained from a π-stationary Markov chain will be distributed as π. This means that the chain satisfies

$$\int_{x \in \Omega} \pi(dx) P(x, dy) = \pi(dy). \tag{2.2}$$

In practice, the distribution of the starting point is not necessarily π, and stationarity in itself provides no guarantees of asymptotic convergence, as this simple example in Roberts and Rosenthal (2004) shows:

Example 1. Let $\mathcal{X} = \{1, 2, 3\}$ and π be uniform on $\mathcal{X}$. A Markov chain (X_n) with transition matrix

$$P = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

is π-stationary, but the distribution of X_n does not converge to π.

The reason convergence does not occur in this case is the reducibility of the chain. Let $P^n(x, A)$ be the transition kernel at the n^{th}-step, namely

$$P^n(x, A) = \mu(X_n \in A | X_0 = x).$$

For finite state spaces, a chain is *irreducible* if there exists an integer t such that $P^t(x, y)$ for any $x, y \in \Omega$. A problem can be encountered when the state space Ω is uncountable, as in the continuous state space case $\Omega = \mathbb{R}$, since the condition is impossible (Roberts and Rosenthal, 2004). For this reason, the weaker notion of ϕ-irreducibility is introduced. Together with *aperiodicity*, defined negatively below, the two notions form the necessary conditions for the asymptotic convergence theorem to be stated.

Definition 2.2.1 (ϕ-irreducibility)**.** A Markov chain is ϕ-irreducible if there exists a non-zero σ-finite measure ϕ on Ω such that, for any $A \subseteq \Omega$ with $\phi(A) > 0$, there exists an $n = n(x, A) > 0$ such that $P^n(x, A) > 0$.

Definition 2.2.2 (Aperiodicity)**.** A π-stationary Markov chain on Ω is *periodic* if there exist a $d \geq 2$ such that a collection of disjoint subsets $\Omega_1, \dots, \Omega_d \subseteq \Omega$ (with $\pi(\Omega_1) > 0$) satisfies $P(x, \Omega_{i+1}) = 1$ for all $x \in \Omega_i$ with $1 \leq i \leq d-1$, and $P(x, \Omega_1) = 1$ for $x \in \Omega_d$. A chain is aperiodic if it is not periodic.

Asymptotic convergence

Distance from stationarity is measured in terms of the *total variation* distance, which represents the maximum difference between the probabilities assigned by two probability measures to an event $A \in \mathcal{B}(\Omega)$.

Definition 2.2.3 (Total variation distance)**.** The total variation distance between two probability measures π_1 and π_2 on the σ-algebra $\mathcal{B}(\Omega)$ is:

$$||\pi_1 - \pi_2||_{TV} = \sup_A |\pi_1(A) - \pi_2(A)|$$

Theorem 4 (Asymptotic convergence theorem)**.** Let $P(\cdot, \cdot)$ be a π-invariant Markov kernel on Ω. If it is ϕ-irreducible and aperiodic, then for π-a. e. $x \in \Omega$,

$$\lim_{n \to \infty} ||P^n(x, \cdot) - \pi(\cdot)||_{TV} = 0$$

Notably, $\lim_{n\to\infty} P^n(x, A) = \pi(A)$ for all measurable $A \subseteq \Omega$.

It is possible to define a bound on the convergence rate to the limiting distribution. This property is referred to as *uniform ergodicity*, and conditions ensuring it are established below following Tierney (1994) and Roberts and Rosenthal (2004).

Definition 2.2.4 (Uniform ergodicity)**.** A π-stationary Markov chain is uniformly ergodic if

$$||P^n(x, \cdot) - \pi(\cdot)||_{TV} \leq M\rho^n, \quad n = 1, 2, 3, \dots$$

for some $\rho < 1$ and $M < \infty$.

Definition 2.2.5. A set $C \in \mathcal{B}(\Omega)$ is small if there exists a positive integer m, a constant $\beta > 0$, and a probability measure ν on $\mathcal{B}(\Omega)$ such that

$$P^m(x, \cdot) \geq \beta\nu(\cdot) \quad \text{for all } x \in C,$$

defined as the minorization condition $M(m, \beta, C, \nu)$ for transition kernel P.

Theorem 5. A Markov chain is uniformly ergodic if and only if the state space Ω is small.

A proof can be found in Nummelin (1984). It should be noted that if the state space is finite, all irreducible and aperiodic Markov chains are uniformly ergodic (Roberts and Rosenthal, 2004).

Metropolis-Hastings algorithm

In practical applications, the focus of the problem reduces to finding a kernel satisfying 2.2. A simpler way to impose this condition is by letting the chain be *reversible*, and stationarity is then implied.

Definition 2.2.6. A Markov Chain on Ω is π-reversible, with π being a probability distribution on Ω, if

$$\pi(dx)P(x, dy) = \pi(dy)P(y, dx), \quad x, y \in \Omega$$

Proposition 1. If a Markov chain is π-reversible, then π is stationary for the chain.

It is therefore sufficient to construct a Markov chain that satisfies reversibility. This result is deployed in the Metropolis-Hastings algorithm. Let π be the target density in the sampling procedure, and let π_u be its unnormalized version. Let $Q(x, dy)$ be an arbitrary transition kernel with unnormalized density $q(x, y)$, such that $Q(x, dy) \propto$

$q(x, y)dy$. The Metropolis-Hastings algorithm generates a new *proposal* $X_t = y$ at each step t conditionally on $X_{t-1} = x$ drawn at the previous iteration, sampling from $Q(x, dy)$. The proposal is then accepted with an *acceptance probability* $\alpha(x, y)$, that in the original Metropolis-Hastings algorithm is

$$\alpha(x, y) = \min \left[1, \frac{\pi_u(y)q(y, x)}{\pi_u(x)q(x, y))} \right]$$

The resulting Markov kernel can be formally expressed as

$$P_{\text{MH}}(x, dy) = Q(x, dy)\alpha(x, y) + r(x)\delta_x(dy)$$

where $r(x) = \int_\Omega (1 - \alpha(x, y))Q(x, dy)$

Proposition 2. The kernel $P_{\text{MH}}(x, dy)$ produces a Markov chain that is π-reversible.

This algorithm is particularly attractive because it lifts the need to find a kernel satisfying reversibility, enforcing it through the correction by an acceptance step. It is therefore sufficient to compute the density point-wise at each iteration.

Gibbs sampler

Let π_u be the unnormalized k-dimensional density of a random vector $\boldsymbol{X} = (X_1, \dots, X_k)$, where $\boldsymbol{X} \in \mathcal{X}$. The Gibbs sampler simulates in turn from the conditional distributions of the components of $\boldsymbol{X}$, conditionally on all the other components. Such distributions are termed *full conditionals*.

For a set $S_{x,i,A} = \{y \in \mathcal{X}; y_j = x_j \text{ for } j \neq i, \text{ and } y_i \in A\}$, the kernel for the full conditional of X_i is defined as

$$P_i(x, S_{x,i,A}) = \frac{\int_A \pi_u(x_1, \dots, x_{i-1}, t, x_{i+1}, \dots, x_k)dt}{\int_{-\infty}^{\infty} \pi_u(x_1, \dots, x_{i-1}, t, x_{i+1}, \dots, x_k)dt}.$$

Variations of the Gibbs sampler are obtained by picking the components in different orders, or more formally by combining the kernels in a different manner. Two of the most popular schemes are:

- **Random-scan Gibbs sampler**: at each step an update is performed on the variable whose index has been sampled from a uniform distribution on $\{1, \dots, k\}$. The kernel is then obtained by a *mixture* of the kernels for the full conditionals

$$P = \frac{1}{k} \sum_{i=1}^{k} P_i. \tag{2.3}$$

- **Deterministic-scan Gibbs sampler**: coordinates are updated sequentially in a deterministic order. The kernel is then obtained by *cycle* of the kernels for the full conditionals

$$P = P_1 P_2 \dots P_k. \tag{2.4}$$

Tierney (1994) argues that while a single kernel is typically not irreducible, a combination that encompasses the entirety of $\mathcal{X}$ can produce an irreducible kernel. The manuscript provides theorems showing the validity of kernel cycles and mixtures, of which Gibbs samplers are particular instances.

Theorem 6. Let P_1 and P_2 be two π-invariant kernels, and P_1 be uniformly ergodic. Then, for $0 < \alpha < 1$, the mixture kernel $\alpha P_1 + (1 - \alpha) P_2$ is uniformly ergodic.

Theorem 7. Let P_1 and P_2 be two π-invariant kernels, and let P_1 satisfy the minorization condition $M(1, \beta, \Omega, \nu)$ for some β and ν. Then, the cycle kernels $P_1 P_2$ and $P_2 P_1$ are uniformly ergodic.

The minorization condition with $m = 1$ is always satisfied by Metropolis-Hastings kernels with an independent proposal, therefore it is sufficient to insert such a kernel in any mixture or cycle of π-invariant kernels to gain uniform ergodicity.

When closed-form expressions of the full conditionals are unavailable, one can resort to

Metropolis-within-Gibbs samplers. Simulations are drawn from kernels that are stationary with respect to the full conditionals, and invariance is guaranteed by an acceptance step.

2.2.3 Asymptotic variance

A key quantity in assessing convergence of a Monte Carlo estimate is *asymptotic variance*, and a definition from Deligiannidis and Lee (2018) is presented below.

Definition 2.2.7. For a generic method MC that produced a sample $X_1 = x_1, \dots, X_n = x_n \in \mathcal{X}$ to approximate a function $h \in L^2(\mathcal{X}, \mu)$ by an ergodic average $\hat{h}_N$, the asymptotic variance of that ergodic average is

$$\text{var}(h, MC) := \lim_{N\to\infty} N\text{var}\left\{\frac{1}{N}\sum_{n=1}^{N} h(x_n)\right\}, \qquad x_n \sim \pi$$

This quantity is fundamental in establishing Central Limit Theorems for several algorithms. A notable result in this sense is Theorem 5 in Tierney (1994). The result assumes uniform ergodicity of the sample produced by a stationary chain, but it follows from section 3.3 of Roberts and Rosenthal (2004) that this condition is always satisfied if we reduce the focus to aperiodic and irreducible Markov chains in finite state spaces.

Theorem 8. Let $X_1, \dots, X_N \in \mathcal{X}$ be a sample from a Markov chain MC that is π-stationary, and suppose $h \in L^2(\mathcal{X}, \pi)$ is real valued and that $\mathcal{X}$ is finite. Then, there exist a real number $\text{var}(h, MC)$ such that the distribution of $\sqrt{N}(\hat{h}_N - \pi(h))$ converges weakly to $N(0, \text{var}(h, MC))$ for any initial distribution.

Chan and Geyer (1994) established that the variance for stationary Markov chains started in stationarity ($X_1 \sim \pi$ for a target π) is

$$N\text{Var}(\hat{h}_N) = \text{Var}(h(X_1)) + 2\sum_{k=1}^{N-1}\frac{N-k}{N}\text{Cov}(h(X_1), h(X_k)).$$

Then, the asymptotic variance is equal to

$$\lim_{N\to\infty} N\mathrm{Var}(\hat{h}_N) = \mathrm{Var}(h(X_1)) + 2\sum_{k=1}^{\infty} \mathrm{Cov}(h(X_1), h(X_k)),$$

that clearly shows how the quality of a MCMC ergodic average depends on the degree of autocorrelation between the samples drawn from the Markov chain.
Asymptotic variance can also be determined for SIS (Deligiannidis and Lee (2018)), and for an estimate $\hat{h}_N$ with IS weights w it is

$$\mathrm{var}(h, SIS) = \mathbb{E}_h[\hat{h}_N w].$$

Another common quantity used to evaluate the quality of the sample used in a Monte Carlo estimate is the *effective sample size* (ESS). As it shall be seen, this quantity can be defined as a transformation of the asymptotic variance. Let $\mathrm{Var}(\tilde{h}_n) = \mathrm{Var}_\pi(h(X))/n$ be the variance of a Monte Carlo estimator for a sample drawn directly from the target π. Then, ESS for Markov chains can be defined as

$$ESS(\hat{h}_N) = N\,\frac{\mathrm{Var}(\tilde{h}_N)}{\mathrm{Var}(\hat{h}_N)} = \frac{N}{1 + 2\sum_{k=1}^{\infty}\mathrm{Corr}(h(X_1), h(X_k))}$$

The quantity is similarly defined for the set of weights $w^{1:N}$ of a SIS estimate, see Chopin et al. (2020).

$$ESS(w^{1:N}) := \frac{(\sum_{n=1}^{N} w(X_n))^2}{\sum_{n=1}^{N} w(X_n)^2}$$

An attractive feature of this quantity is its interpretability that, as the name suggests, is related to the true size of uncorrelated samples among those that were simulated. Indeed, for the SIS we have that $ESS(w^{1:N}) \in [1, N]$, and if k weights are one and the rest are zero, then $ESS(w^{1:N}) = k$. The quantity is also particularly important when we are interested in estimating the variance of the ergodic average for MCMC (Robert and Casella, 2004). The standard variance estimator $N^{-2}\sum_{n=1}^{N}(h(X_n) - \hat{h}_N)$ cannot be used, since the correlation of the samples would result in an underestimation of the variance of $\hat{h}_N$. Reliability of the variance estimate can be improved by considering the estimator

$$\frac{1}{N \times ESS(\hat{h}_N)} \sum_{n=1}^{N} (h(X_n) - \hat{h}_N)^2. \tag{2.5}$$

Chapter 3

Bayesian variable selection

When approaching an inference problem, choosing the right statistical model is of crucial importance. The problem of model selection is approachable from a probabilistic standpoint through the theoretical tools of Bayesian inference. An extensive review on this topic is provided in Chipman et al. (2001).

3.1 General framework of Bayesian model selection

Let $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_K\}$ be a collection of parametric models for the data, denoted as Y. Assuming identifiability up to a parameter θ_k, the conditional density of the data for a given model will be $p(Y|\theta_k, \mathcal{M}_k)$. In the Bayesian approach, a prior distribution is assigned to each model in $\mathcal{M}$ and to the value of θ_k. Combining the distributions in a hierarchical model induces a joint distribution, that allows for model selection based on the conditional distribution of the model given the data. The resulting model in hierarchical form can be represented as

$$\begin{aligned} Y|\theta_k, \mathcal{M}_k &\sim p(Y|\theta_k, \mathcal{M}_k), \\ \theta_k|\mathcal{M}_k &\sim p(\theta_k|\mathcal{M}_k), \\ \mathcal{M}_k &\sim p(\mathcal{M}_k). \end{aligned} \tag{3.1}$$

The problem of model selection reduces to finding those with higher posterior probability, with the posterior easily obtainable, with Bayes' theorem, as

$$p(\mathcal{M}_k|Y) = \frac{p(Y|\mathcal{M}_k)p(\mathcal{M}_k)}{\sum_k p(Y|\mathcal{M}_k)p(\mathcal{M}_k)}, \tag{3.2}$$

with

$$p(Y|\mathcal{M}_k) = \int p(Y|\theta_k, \mathcal{M}_k)p(\theta_k|\mathcal{M}_k)d\theta_k \tag{3.3}$$

being the marginal likelihood of $\mathcal{M}_k$. Pairwise comparison of models are based on posterior odds, that are the product between Bayes factors and prior odds:

$$\frac{p(\mathcal{M}_1|Y)}{p(\mathcal{M}_2|Y)} = \frac{p(Y|\mathcal{M}_1)}{p(Y|\mathcal{M}_2)} \times \frac{p(\mathcal{M}_1)}{p(\mathcal{M}_2)}$$

Challenges in this approach are the specification of the prior distributions in 3.1 and the computation of the posterior in 3.2.

It should be noted that besides taking the mode of the posterior distribution, one could also resort to a *model averaging* approach. Assuming the aim is to make inferences about some quantity of interest ζ, performing Bayesian model averaging would amount to computing the posterior of ζ as

$$p(\zeta|Y) = \sum_{k=1}^{K} \mathbb{E}[\zeta|\mathcal{M}_k, Y]p(\mathcal{M}_k|Y),$$

and the posterior mean would be defined as

$$\mathbb{E}[\zeta|Y] = \sum_{k=1}^{K} \mathbb{E}[\zeta|\mathcal{M}_k, Y]p(\mathcal{M}_k|Y).$$

Prior specification

The simplest prior specification for $p(\mathcal{M}_k)$ is that of uniform priors for the different models:

$$p(\mathcal{M}_k) = \frac{1}{K}$$

Under such a choice, $p(\mathcal{M}_k|Y) \propto p(Y|\mathcal{M}_k)$ and pairwise comparisons reduce to a comparison of Bayes factors. This choice has some significant drawbacks, as it does not account for the "size" of each model $\mathcal{M}_k$, and for the similarity between models. The choice of parameter priors $p(\theta_k|\mathcal{M}_k)$ is typically aimed at reducing computational cost of the marginal (3.3). If the model belongs to an exponential family of distributions, conjugate priors are a common choice.

Posterior computation

Computing the integral in (3.3) and, consequently, the denominator in (3.2) can be challenging, especially in high-dimensional settings. MCMC techniques can be used to simulate directly from the posterior, and they allow for a search in the space of models.

Let η denote the couple $(\theta_k, \mathcal{M}_k)$, such that each η specifies a density $p(Y|\eta)$. A MCMC approach would build a Markov chain by simulating a sequence $\eta^{(1)}, \eta^{(2)}$ from a transition kernel $P(\eta^{(j-1)}, d\eta^{(j)})$ having as stationary distribution the posterior $p(\eta|Y)$. Examples of practical implementations of these techniques are Gibbs Samplers (GS) and Metropolis-Hastings (MH) algorithms. In the former, considering the case of $\eta \in \mathbb{R}^p$, samples are obtained simulating iteratively from the full conditional $p(\eta_i|\eta_{-i}, Y)$ for $i = 1, \dots, p$. MH algorithms sample a candidate η from an arbitrary transition kernel, specified by the proposal density $q(\eta^{(j)}|\eta^{(j-1)})$, and then proceed with the acceptance step.

Computations are facilitated when closed-form expressions of the marginal density $p(Y|\mathcal{M}_k)$ are available. In alternative, easily computable approximations can be used, such as the Laplace approximation, detailed in Tierney and Kadane (1986). Let $p(Y|\mathcal{M}_k) =$

$\int e^{h(\theta_k)} d\theta_k$ with $h(\theta_k) \equiv \log p(Y|\theta_k, \mathcal{M}_k)p(\theta_k|\mathcal{M}_k)$ being a smooth and positive function, and let $\hat{\theta}_k$ be the MAP estimator of θ_k. Computing the Taylor expansion and evaluating it at $\hat{\theta}_k$ yields $h(\theta_k) \approx h(\hat{\theta}_k) + \frac{1}{2}(\theta_k - \hat{\theta}_k)' H(\hat{\theta}_k)(\theta_k - \hat{\theta}_k)$, where $H(\hat{\theta}_k)$ is the Hessian of h. Plugging it in the expression of $p(Y|\mathcal{M}_k)$ and solving the Gaussian integral gives

$$p(Y|\mathcal{M}_k) \approx (2\pi)^{d_k/2} |A(\hat{\theta}_k)|^{1/2} p(Y|\hat{\theta}_k, \mathcal{M}_k) p(\hat{\theta}_k|\mathcal{M}_k)$$

Where $A(\hat{\theta}_k)$ is $-H(\hat{\theta}_k)^{-1}$ and d_k is the dimension of θ_k. When obtaining the posterior mode $\hat{\theta}_k$ is costly, it is also possible to expand at the maximum likelihood estimate $\hat{\theta}_k^{MLE}$ and substitute $A(\hat{\theta}_k)$ with Fisher's information matrix. Schwarz (1978) showed that a further approximation can be obtained considering $A = nA^{(unit)}$, taking the log, and ignoring the terms that are asymptotically constant. What is left is the Bayesian Information Criterion (BIC) approximation, defined as

$$\log p(Y|\mathcal{M}_k) \approx \log p(Y|\hat{\theta}_k, \mathcal{M}_k) - (d_k/2) \log n.$$

3.1.1 Bayesian variable selection for the normal linear model

The analytical tractability of linear models makes them particularly feasible for the analysis of Bayesian variable selection. Let Y be the variable of interested, referred to as *response variable*, and suppose it is explainable by a linear combination of a set of *predictors* $X_1, \ldots, X_p$, observed in n samples. Let X denote the design matrix, that is the $n \times p$ matrix representation of such samples. The problem of model selection then reduces to variable selection, that is finding an optimal subset of the p many explanatory variables. This setting is typical in high-dimensional regression problems, where p is particularly large and we are interested in making inferences about a target variable Y with a parsimonious model.

Normal linear model

Suppose that the response variable Y is continuous, and that the error is normally distributed. Then, the model is typically referred to as *normal linear model*, and it takes the form

$$Y|X, \beta, \sigma \sim N_n(X\beta, \sigma^2 I) \tag{3.4}$$

where $\beta \in \mathbb{R}^p$ and $\sigma \in \mathbb{R}_+$. An extensive review of this model is provided in Gelman et al. (1995).

Being in a Bayesian setting, a joint distribution is induced by placing a prior on the parameters β and σ. It is common practice to assume X as non-random (as in a controlled experiment). In any case, it is immediate to see that, even if X were given a prior $p(X|\psi)$, an assumption of independence between the priors of $\{\beta, \sigma\}$ and ψ would be enough to factor the posterior as $p(\psi, \beta, \sigma|X, Y) = p(\psi|X)p(\beta, \sigma|X, Y)$, justifying an analysis of $p(\beta, \sigma|X, Y)$ alone (Gelman et al., 1995). For this reason, conditioning on X is omitted in the following notation.

The exponential form of the model allows for a conjugate normal prior on $\beta|\sigma$, whereas for σ^2 a standard noninformative prior is assigned as in Fernández et al. (2001), being $p(\sigma^2) \propto 1/\sigma^2$. The prior is improper, and it could be interpreted as an Inverse-Gamma prior with the two parameters equally close to zero. The priors then take the form

$$\begin{aligned} \beta|\sigma^2 &\sim N(\beta_0, \sigma^2\Sigma) \\ \sigma^2 &\sim p(\sigma^2) \propto 1/\sigma^2 \end{aligned}$$

It is common to set $\beta_0 = 0$, representing a neutral opinion about the sign of the effect of each predictor, and to choose either $\Sigma = c(X'X)^{-1}$ with $c \in \mathbb{R}_+$ or $\Sigma = c\mathbb{I}_n$. In the latter case, it is easy to see that the log posterior takes the form of a penalized sum of squares, analogous to the loss function for ridge regression. The former formulation is typically referred to as g-priors and it is motivated in Zellner (1986). Essentially, the parameter c regulates the width of the region of plausible values for β. A smaller c translates into higher confidence that the effect of the predictors is close to zero. Simulations performed in Fernández et al. (2001) in the case where $p(\gamma) = 2^{-p}$ suggest that a reasonable choice (in terms of predictive performance) is $c = \max\{p^2, n\}$. It is possible to formulate the variable selection problem by introducing a vector indexing the variables we wish to include in the model. Such vector is

$$\gamma = (\gamma_1, \dots, \gamma_p) \in \{0,1\}^p$$

Denoting by $|\gamma| = \sum_{i=1}^p \gamma_i$ the number of active predictors, we use X_γ and β_γ to indicate the design matrix and the coefficients vector containing only the included variables. Prior independence is assumed for components γ, therefore the prior distribution takes the form of a product of Bernoulli random variables:

$$p(\gamma) = \prod_{i=1}^{p} w_i^{\gamma_i}(1-w_i)^{(1-\gamma_i)} \tag{3.5}$$

where a *prior inclusion probability* can be specified for each predictor, or reduce the model by setting $w_i \equiv w \ \forall i$. In that case, w represents the expected proportion of included variables. The hierarchical model under this new formulation becomes

$$\begin{aligned} Y|\beta,\sigma,\gamma,w &\sim N_n(X_\gamma\beta_\gamma, \sigma^2 I) \\ \beta|\sigma,\gamma,w &\sim N(0, \sigma^2\Sigma_\gamma) \\ \sigma^2 &\sim p(\sigma^2) \propto 1/\sigma^2 \\ \gamma_i|w &\sim \text{Bern}(w) \qquad\qquad i = 1,\dots,p \end{aligned} \tag{3.6}$$

where Σ_γ is either $c(X_\gamma' X_\gamma)^{-1}$ or $\mathbb{I}_\gamma$. The model in 3.6 with g-priors will be the one under consideration for the following chapters, but it is only one of the many possibilities present in the literature regarding Bayesian variable selection.

3.1.2 Gibbs sampling for Bayesian variable selection

Performing variable selection in this setting is possible by making inference on the posterior $p(\gamma|Y)$. The most desirable case, as it is the fastest in computational terms, is when a closed-form expression of the posterior is available. This is achievable when a closed-form expression for $p(Y|\gamma)$ is obtainable, that is the likelihood marginalizing over β and σ, since

$$p(\gamma|Y) \propto p(Y|\gamma)p(\gamma).$$

When this is not possible, one can resort to MCMC methods that simulate a sequence of γ whose transition kernel converges asymptotically to the posterior as in 4. An estimate of the posterior inclusion probabilities (PIPs) can then be obtained. When the full conditional is available, it is possible to simulate a sequence

$$\gamma^{(1)}, \gamma^{(2)}, \dots \tag{3.7}$$

directly by running a Gibbs sampler on the full conditional $p(\gamma_i|\gamma_{-i}, Y)$ of each component. When the component-wise conditionals of γ are unavailable, the sequence 3.7 is obtained as a subsequence of the Markov chain

$$\beta^{(1)}, \sigma^{(1)}, \gamma^{(1)}, \beta^{(2)}, \sigma^{(2)}, \gamma^{(2)}, \dots \tag{3.8}$$

Simulating this sequence with a Gibbs sampler requires an iterative simulation from the full conditionals of all the parameters, that are

$$\begin{gathered} p(\beta|\sigma^2, \gamma, Y) \\ p(\sigma^2|\beta, Y) \\ p(\gamma_i|\gamma_{-i}, \beta)\ ,\ i = 1, \dots, p \end{gathered} \tag{3.9}$$

Computation of the full conditionals

For the model introduced in the previous section, it is possible to derive the full conditionals of each component of γ, sampling 3.7 directly. As suggested in the supplement of Zanella and Roberts (2019), we can easily derive $p(\gamma_i = 1|\gamma_{-i}, Y)$ by computing the ratio $r = \frac{p(\gamma_i=1|\gamma_{-i},Y)}{p(\gamma_i=0|\gamma_{-i},Y)}$ and then recovering the conditional using $p(\gamma_i = 1|\gamma_{-i}, Y) = r/(1+r)$. Standard computations of the marginal likelihood for Bayesian linear regression models with g-priors yield (Chipman et al., 2001)

$$p(Y|\gamma) \propto (1+c)^{-|\gamma|/2}(S(\gamma))^{-n/2} \tag{3.10}$$

where $S(\gamma) = Y'Y - \frac{c}{1+c}Y'X_\gamma(X'X)^{-1}X'_\gamma Y$. We can then use 3.10 to get

$$r = \frac{p(\gamma_i = 1|\gamma_{-i})}{p(\gamma_i = 0|\gamma_{-i})}\frac{p(Y|\gamma_{-i}, \gamma_i = 1)}{p(Y|\gamma_{-i}, \gamma_i = 0)} = \frac{w}{1-w}\frac{1}{(1+c)^{1/2}}\left(\frac{S(\gamma^0)}{S(\gamma^1)}\right)^{n/2}$$

where γ^0 represents the vector with γ_{-i} and $\gamma = 0$, and γ^1 contains γ_{-i} and $\gamma = 1$. Computational efficiency can be improved by following some tricks suggested in Smith and Kohn (1996), and further improvements from Zanella and Roberts (2019).

First, to avoid unnecessary computations, we can obtain recursive updates for the $S(\gamma)$ term. At each iteration, either $S(\gamma^0)$ or $S(\gamma^1)$ has already been computed at the previous step. Considering an iteration in which $\gamma = \gamma^0$ before the sampling step, it is only necessary to obtain $S(\gamma^1)$. The method proceeds as follows. Define $F_{\gamma^0} = (X'_{\gamma^0}X_{\gamma^0})^{-1}$. Being symmetric positive definite, it admits a unique Cholesky factorization $F_{\gamma^0} = LL'$, for L being a lower triangular matrix. These operations are typically not particularly expensive, as in most applications with real data the amount of active regressors $|\gamma|$ is small (an alternative implementation when the dimensionality of γ makes the two operations unfeasible is presented subsequently).

Let $I = \{j : \gamma^0_j = 1\}$ denote the set of indices of variables included in γ^0, then for

$v = X'Y$ we consider $v_{\gamma^0} = (v_j)_{j \in I}$. Let $A = X'X$, whose elements included in the γ^0 configuration are denoted as $a_i = (A_{ji})_{j \in I}$. Then, computation of $S(\gamma^1)$ for a given $S(\gamma^0)$ can efficiently be obtained by

$$S(\gamma^1) = S(\gamma^0) - \frac{c}{1+c} d_i (v'_{\gamma^0} F_{\gamma^0} a_i - v_i)^2, \tag{3.11}$$

where $d_i = (A_{ii} - a_i' F_{\gamma^0} a_i)^{-1}$. The result can be obtained deploying the Woodbury matrix identity and Schur complements (see Appendix B of Beal (2003)), and additional calculations can be found in the supplement of Zanella and Roberts (2019). Introducing the Cholesky decomposition of F defined above, computations can further be reduced by using

$$d_i = \sum_{j \in I} \left(\sum_{h \in I} A_{ih} L_{hj} \right) = \sum_{j \in I} (BL)^2_{ij}, \tag{3.12}$$

for B being the $p \times |\gamma|$ matrix of columns of A being included in γ. This amounts to summing the squared ℓ^2 norms of the rows of BL.

Next, the case where the iteration starts with $\gamma = \gamma^1$ and we want to retrieve $S(\gamma^0)$ is considered. It should be noted that quick updates of $F_{\gamma^1} = (X'_{\gamma^1} X_{\gamma^1})^{-1}$ are possible at each iteration, deploying results from linear algebra regarding the updates of matrices like F_{γ^1} when rows of X_{γ^1} are removed. Following Hager (1989), consider the case when the row is added in the last position. Such a case can easily be retrieved by permuting rows of F_{γ^1} before the inital step. When removing the variable j, let

$$\begin{aligned} F_{\gamma^1} &= \begin{bmatrix} X'_{\gamma^0} X_{\gamma^0} & X'_{\gamma^0} x_j \\ x_j' X_{\gamma^0} & x_j' x_j \end{bmatrix}^{-1} \\ &= \begin{bmatrix} F_{\gamma^1[-j,-j]} & F_{\gamma^1[\,\cdot\,,-j]} \\ F'_{\gamma^1[\,\cdot\,,-j]} & F_{\gamma^1[\,j,\,j]} \end{bmatrix}. \end{aligned}$$

Then,

$$F_{\gamma^0} = F_{\gamma^1[-j,-j]} - F^{-1}_{\gamma^1[\,j,\,j]}(F'_{\gamma^1[\,\cdot\,,-j]}F_{\gamma^1[\,\cdot\,,-j]}).$$

In the case where $|\gamma|$ is large and the initial computation of F_γ and its Cholesky factor is expensive, it is possible to reduce computations by using the Cholesky decomposition of $X'_\gamma X_\gamma$ directly and never computing the inverse. As the steps defined above are mainly concerned with computing quadratic forms, they can be performed efficiently by skipping the matrix inversion step. Considering for example $v'_\gamma F_\gamma v_\gamma$, the result can be obtained by taking the Cholesky factor C such that $CC' = X'_\gamma X_\gamma$, and then solving the following triangular systems of equations defined in Algorithm 1 (Rue and Held, 2005).

Algorithm 1 Solving quadratic form with no inverse

```
Compute $CC' = X'_\gamma X_\gamma$
Solve $C \cdot v_\gamma = \boldsymbol{z}$ with forward substitution
Solve $C' \cdot \boldsymbol{z} = \boldsymbol{w}$ with backward substitution
return $v'_\gamma \cdot \boldsymbol{w}$
```

Cholesky factors are then updated at each step following the deletion of columns in $X'_\gamma X_\gamma$, and computations to make the operation efficient are reported in Appendix A.1. This alternative implementation for computing the full conditionals, based on the code of Zanella and Roberts (2019), can be found in Appendix A.2.

Point estimation of regression coefficients

Suppose a Gibbs sampler was run to produce a chain as in 3.7. In a practical problem of regression, it might be insightful to compute a point estimate of the regression coefficients. Two ways by which the task can be achieved are presented in Smith and Kohn (1996).

The first method is based on an estimate of the posterior mode of γ. The support of the posterior distribution $p(\gamma|Y)$ has size 2^p and it is hence difficult to find the mode by direct enumeration, but an estimate can be obtained exploiting the fact that the Gibbs sampler iterations lie in a region of high probability (Smith and Kohn, 1996). Hence, the value of

$\gamma^{(t)}, t = 1, \dots, T$ maximizing $p(\gamma|Y)$ is taken as the estimate of the posterior mode, and it can be denoted as $\hat{\gamma}_M$. Considering the likelihood as in 3.10 and the prior as in 3.5, computing such a quantity is negligible once the samples are obtained. The regression coefficients estimate $\hat{\beta}$ can then be obtained using least squares and considering only the variables included in $\hat{\gamma}_M$, that would essentially work as a plug-in estimate. Therefore,

$$\mathbb{E}(\beta|Y, \hat{\gamma}_M, \sigma^2) = \frac{c\,\hat{\beta}}{1+c}, \tag{3.13}$$

where $\hat{\beta} = (X'_{\hat{\gamma}_M} X_{\hat{\gamma}_M})^{-1} X'_{\hat{\gamma}_M} Y$ and c being the hyperparameter defined in Section 3.1.1.

The second method considers an estimate of the posterior mean of β directly, by performing an averaging of the conditional posterior mean $\mathbb{E}(\beta|Y, \gamma^{(t)}, \sigma^2)$ over the $\gamma^{(t)}, t = 1, \dots, T$ samples. The estimate is then

$$\hat{\beta} = T^{-1} \sum_{t=1}^{T} \mathbb{E}(\beta|Y, \gamma^{(t)}, \sigma^2), \tag{3.14}$$

and the expression can be computed for each iteration since Zellner's g-priors allow for a quick evaluation of the posterior mean, that is

$$\mathbb{E}(\beta|Y, \gamma^{(t)}, \sigma^2) = \frac{c\,\hat{\beta}}{1+c}, \tag{3.15}$$

for $\hat{\beta} = (X'_{\gamma^{(t)}} X_{\gamma^{(t)}})^{-1} X'_{\gamma^{(t)}} Y$ and c defined as above.

Another method, described in Narisetty and He (2014), is *median probability thresholding*. Considering the PIPs for each γ, the model includes the variables having PIP above a threshold. A common choice for the threshold is to fix it at 0.5, but it is also possible to tune the parameter using some criterion such as the BIC.

3.2 Tempered Gibbs Sampler for variable selection

3.2.1 Limitations of standard Gibbs sampling

As seen in Section 2.2.2, the Gibbs sampler enjoys attractive properties and enough flexibility to make it the preferable choice for many sampling tasks. Nonetheless, the convergence of the algorithm can be impeded or dramatically slowed down if the posterior distribution has some particular attributes. One of such characteristics is a strong correlation between variables. As can be seen from the plot in Figure 3.1, full conditionals for highly correlated variables are often very concentrated, and this harms the ability of the chain to perform big jumps that speed up exploration and, in turn, convergence to stationarity. Another critical issue is when the support of the posterior can be partitioned in non-connected areas (Robert and Casella, 2004). The first scenario is typical in high-dimensional BVS, as it is often the case that some features are highly correlated. The problem is typically referred to as *multicollinearity*.

The problem is circumvented in Zanella and Roberts (2019) by introducing a new sampling scheme combining Gibbs Sampler and importance sampling, termed Tempered Gibbs Sampling (TGS) scheme. Considering a general setting where we want to sample a $\boldsymbol{x} = (x_1, \dots, x_d) \in \mathcal{X}$ with density $f(\boldsymbol{x})$, the algorithm makes use of a modified version of the full conditionals $\{g(x_i|\boldsymbol{x}_{-i})\}_{i=1}^{d}$. Each modified full conditional $g(x_i|\boldsymbol{x}_{-i})$ is absolutely continuous with respect to the original full conditional $f(x_i|\boldsymbol{x}_{-i})$, and there is no requirement for them to admit a global distribution $g(\boldsymbol{x})$. The algorithm resembles a self-normalized IS within each GS step, and it is presented in Algorithm 2.

An asymptotic convergence result providing theoretical justification for the use of the TGS scheme, provided by the authors, is presented in Proposition 3.

Proposition 3. fZ is a probability density function on $\mathcal{X}$, and the Markov chain induced by steps 1 and 2 of the TGS scheme is fZ-reversible. Assuming $Z(\boldsymbol{x})$ to be bounded away

Algorithm 2 TGS

```
1: Sample a coordinate $i \in \{1, \dots, d\}$ with probability
       $$p_i = \frac{g(x_i|\boldsymbol{x}_{-i})}{f(x_i|\boldsymbol{x}_{-i})} \quad \text{for } i = 1, \dots, d.$$
2: Sample $x_i \sim g(x_i|\boldsymbol{x}_{-i})$.
3: Assign to the new $x_i$ a weight $w(x) = Z^{-1}(x)$, where $Z(\boldsymbol{x}) = d^{-1} \sum_{i=1}^{d} p_i(\boldsymbol{x})$.
```

from zero and bounded above on compact sets, and the TGS chain to be fZ-irreducible, then

$$\hat{h}_T^{TGS} = \frac{\sum_{t=1}^{T} w(\boldsymbol{x}_t) h(\boldsymbol{x}_t)}{\sum_{t=1}^{T} w(\boldsymbol{x}_t)} \xrightarrow{a.s.} \mathbb{E}_f[h], \qquad \text{as } n \to \infty,$$

for every $h \in L^1(\mathcal{X}, f)$.

An attractive feature of the scheme is the possibility to specify as modified full conditionals a *tempered version* of the original full conditional, when this is available in closed form:

$$g(x_i|\boldsymbol{x}_{-i}) = f^{(\beta)}(x_i|\boldsymbol{x}_{-i}) = \frac{f(x_i|\boldsymbol{x}_{-i})^{\beta}}{\int_{\mathcal{X}} f(x_i|\boldsymbol{x}_{-i})^{\beta} dx_i}, \qquad \beta \in (0, 1) \tag{3.16}$$

Another choice proposed by the authors is a *symmetric mixture* of the original and the tempered version.

$$g(x_i|\boldsymbol{x}_{-i}) = \frac{1}{2} f(x_i|\boldsymbol{x}_{-i}) + \frac{1}{2} f^{(\beta)}(x_i|\boldsymbol{x}_{-i}) \tag{3.17}$$

This latter formulation is particularly preferred, as the tempered version is prone to move away from the region of high probability under the target, resulting in a higher variance of the IS weights. In both cases, considering a tempered conditional circumvents the problem with highly correlated variables that traditional Gibbs sampling techniques may encounter. It does so by:

- Making informed choices for variable updates, as opposed to random-scan GS techniques.

- Enabling bigger jumps between samples, resulting in a wide-ranging exploration of the sample space.

The example plotted in Figure 3.1 displays a sample from a standard random-scan GS and one from TGS using a tempered proposal as in 3.16 in two dimensions. The random variables are distributed as a bivariate normal, each with zero mean and unit variance, and the plot displays samples for levels of correlation ρ equal to 0.5, 0.99, and 0.998. Both samplers are ran for $T = 200$ iterations, and they are started at $(Y_1^{(0)}, Y_2^{(0)}) = (2.5, 2.5)$.Simulations for the tempered proposals have been generated using rejection sampling, and the tempering coefficient is $\beta = 1 - \rho$. The R code to reproduce the results can be found in Appendix A.3.

Robustness to high-dimensionality

As seen in section 2.2.1, importance sampling might scale very poorly with respect to the dimensionality of the target. TGS mitigates this problem, and theoretical guarantees for the growth of the asymptotic variance are provided in Zanella and Roberts (2019).

Lemma 1. Let $h \in L^1(\mathcal{X}, f)$ and $\bar{h}(\boldsymbol{x}_{-i}) - \mathbb{E}_f[h]$. If $\mathrm{var}(h, TGS) < \infty$, then

$$\mathrm{var}(h, TGS) = \mathbb{E}_f[\bar{h}^2 w] \left(1 + 2\sum_{t=1}^{\infty} \rho_t\right),$$

with ρ_t being the lag-t autocorrelation of $(w(\boldsymbol{x}_n)h(\boldsymbol{x}_n))_{n=1}^{\infty}$ for a TGS chain started in stationarity.

A useful interpretation of this Lemma comes from noticing that the first term is the asymptotic variance of an SIS using fZ as the proposal, while the second is that of a Markov chain. Robustness to high dimensionality can then be studied by analyzing the variance of the importance weights associated with the SIS component.

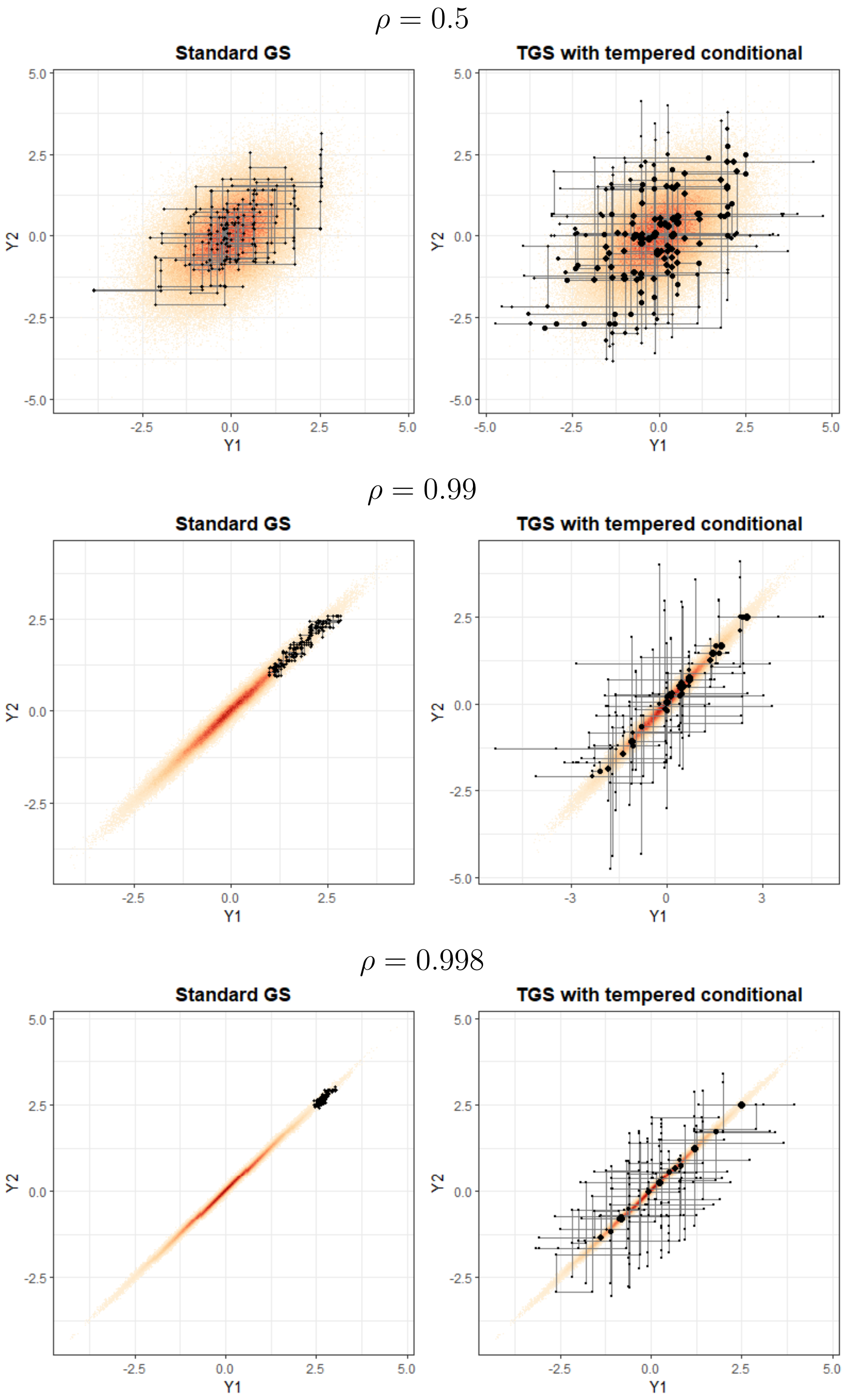

Figure 3.1: Black dots are samples simulated from a random-scan Gibbs Sampler (left plots) and a TGS (right plots). The size of dots in the left plots is proportional to the importance weights $w(x)$ assigned at each sample.

Proposition 4. Let $b = \sup_{i,\boldsymbol{x}} \frac{f(x_i|\boldsymbol{x}_{-i})}{g(x_i|\boldsymbol{x}_{-i})}$. For $W = w(\boldsymbol{X})$ and $\boldsymbol{X} \sim fZ$,

$$Var(W) \leq b - 1 \qquad \textit{and} \qquad \mathrm{var}(h, SIS) \leq b\mathrm{var}_f(h),$$

where $\mathrm{var}_f(h)$ is the standard Monte Carlo variance.

Therefore, as long as b is low, the SIS contribution to the variance of TGS is kept under control. It is the case for example when a symmetric mixture is used, and since b is at most 2, $Var(W) \leq 1$.

3.2.2 TGS for BVS

One can find a natural application of the TGS scheme in the problem of Bayesian Variables Selection outlined in Section 3.1.1. Considering the normal linear model specified in 3.6, it is possible to perform iterated TGS steps on a modified full conditional of the inclusion parameters γ, that for simplicity can be set to be the uniform distribution over the two states $\{0, 1\}$ (that is equivalent to setting $\beta = 0$ in the formulation 3.16). In this case, the algorithm reduces to the following steps:

Algorithm 3 TGS for BVS

```
For $t = 1, \dots, T$:
1: Sample $i$ from $\{1, \dots, p\}$ with probability
       $$p_i(\gamma) = \frac{1}{2p(\gamma_i|\gamma_{-i}, Y)} \quad \text{for } i = 1, \dots, d.$$
2: Set $\gamma_i \leftarrow 1 - \gamma_i$
3: Compute a weight $w(\gamma) = Z(\gamma)^{-1}$ for the new state $\gamma$, where $Z(\gamma) = \frac{1}{p}\sum_{i=1}^{p} p_i(\gamma)$
```

wTGS for BVS

It can be proven that TGS samples each index $i \in \{1, \dots, p\}$ with the same frequency (see Zanella and Roberts (2019)), and clearly in high-dimensional problems having variables with low posterior inclusion probability this might be very inefficient. The authors devised a version of TGS aimed at solving this problem, by weighting each p_i in step 1 of Algorithm 3 with a quantity $\eta_i(\gamma_{-i})$ proportional to the posterior inclusion probability. In BVS, this quantity was chosen to be $\eta_i(\gamma_{-i}) = p(\gamma_i = 1|\gamma_{-i}, Y) + \frac{k}{p}$, with $k < p$ being a fixed parameter. The algorithm, termed weighted Tempered Gibbs Sampler (wTGS), proceeds as follows.

Algorithm 4 wTGS for BVS

```
For $t = 1, \dots, T$:
1: Sample $i$ from $\{1, \dots, p\}$ with probability
       $$p_i(\gamma) = \frac{p(\gamma_i = 1|\gamma_{-i}, Y) + k/p}{2p(\gamma_i|\gamma_{-i}, Y)} \quad \text{for } i = 1, \dots, d.$$
2: Set $\gamma_i \leftarrow 1 - \gamma_i$
3: Compute a weight $w(\gamma) = Z(\gamma)^{-1}$ for the new state $\gamma$, where $Z(\gamma) \propto \sum_{i=1}^{p} p_i(\gamma)$
```

Computational complexity

Computations required at each iteration derive mostly from the scheme used to update the full conditionals, defined in 3.1.2. Assuming that $X'X$ and $Y'X$ are precomputing before running the algorithm, the cost is dominated by the computation of the $(d_i)_{i=1}^{p}$ terms in 3.12, that require $\mathcal{O}(p|\gamma|)$ operations at each iteration. If instead computing or storing $X'X$ is prohibitive and one wants to update the entries of $X_\gamma' X_\gamma$ and B (see 3.12) sequentially at each iteration, the cost is n and is performed at most p times, that compounded with the previous term results in a $\mathcal{O}(np + p|\gamma|)$ cost. In comparison, the standard Gibbs sampler runs at a cost that is lower of order $|\gamma|/p$.

Following the framework in Zanella and Roberts (2019), computational complexity is defined as the product between cost per iteration and the number of iterations required to obtain MC estimator with effective sample size of order 1. While the former has been established following the implementation to update the full conditionals, the latter can be determined considering the *relaxation time* of the Markov chains deployed in each algorithm. It can be shown that the relaxation times of the three algorithms are not significantly impacted by different correlation structures, and Table 3.1 reviews computational complexity results for the three algorithms under mild assumptions (see Zanella and Roberts, 2019 for further details). The results are defined for $s = \mathbb{E}_f(|\gamma|)$.

	GS	TGS	wTGS
Case 1	$\mathcal{O}(ps^2)$	$\mathcal{O}(p^2s)$	$\mathcal{O}(ps^2)$
Case 2	$\mathcal{O}(nps)$	$\mathcal{O}(np^2)$	$\mathcal{O}(nps)$

Table 3.1: Comparison of the computational complexity of GS, TGS, and wTGS. Case 1 refers to the case of pre-computed $X'X$ and $X'Y$, whereas in case 2 the entries referring to the element that is being updated have to be computed at each iteration.

Rao-Blackwellisation

In many statistical applications, it is possible to reduce the variance of an estimator $\delta(X)$ by conditioning on another random variable Y, using the inequality

$$\mathrm{var}(\mathbb{E}[\delta(X)|Y]) \leq \mathrm{var}(\delta(X)).$$

Following Robert and Casella (2004), if $\delta = \frac{1}{T}\sum_{t=1}^{T} h(X_t)$ is an estimator of $\mathbb{E}_f[h(X)]$, the estimator $\delta^*(Y) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\tilde{f}}[\delta(X)|Y_t]$ obtained by simulating X from the joint $\tilde{f}(x,y)$ dominates δ in terms of variance. This technique is referred to as *Rao-Blackwellisation*, and it is applicable to compute lower-variance estimates of Posterior Inclusion Probabilities (PIPs) $\{p(\gamma_i = 1|Y)\}_{i=1}^p$.

While a standard Monte Carlo estimator would compute the PIPs by averaging inclusions of each γ_i of a Markov Chain chain ran for T as $T^{-1}\sum_{t=1}^T \mathbb{1}(\gamma_i^{(t)} = 1)$, a *Rao-Blackwellised* estimator would be

$$\hat{p}(\gamma_i = 1|Y) = \frac{\sum_{t=1}^T w(\gamma^{(t)})p(\gamma_i^{(t)}|Y, \gamma_{-i} = \gamma_{-i}^{(t)})}{\sum_{t=1}^T w(\gamma^{(t)})} \tag{3.18}$$

where $w(\gamma^{(t)}) = Z(\gamma^{(t)})^{-1}$ are the importance weights. TGS and wTGS compute the quantities needed in 3.18 at each step, therefore the estimator can be computed recursively at each iteration with no extra costs.

Chapter 4

Simulation studies

4.1 Simulated data

The full potential of the algorithms presented in the previous section can be illustrated considering some simulations in particularly nontrivial situations. Two such situations were described in Zanella and Roberts (2019), both considering a data matrix X that is a zero-mean multivariate Gaussian with $\Sigma_{ii} = 1$, and they are the following:

- *Scenario 1*: two variables with a strong correlation, such that $\Sigma_{12} = \Sigma_{21} = 0.99$, and $\Sigma_{ij} = 0$ for all the other $i \neq j$. The coefficients are then $\beta^* = (1, 0, \dots, 0)$.

- *Scenario 2*: two batches of 3 highly correlated variables, such that $\Sigma_{ij} = 0.9$ if $i, j \in \{1, 2, 3\}$ or $i, j \in \{4, 5, 6\}$, and $\Sigma_{ij} = 0$ otherwise. The coefficients are $\beta^* = (3, 3, -2, 3, 3, -2, 0, \dots, 0)$.

Simulations are ran also on a sample of uncorrelated data (*scenario 3*), where X has covariance $\Sigma_{ij} = 0$ for all $i \neq j$, and $\beta^* = (2, -3, 2, 2, -3, 3, -2, 3, -2, 3, 0, \dots, 0)$. The response vector is then generated as $y \sim \mathcal{N}(X\beta_0^*, \sigma^2)$, with $\sigma^2 = 1$ and $\beta_0^* = \mathrm{SNR}\sqrt{\frac{\sigma^2 \log(p)}{n}}\beta^*$, where SNR is a parameter set at SNR $= 2$. Another nontrivial example considered for the simulations is taken from Ročková and George (2018), where it was used to test the efficacy of the spike-and-slab LASSO, another state-of-the-art BVS method. The simulation (*scenario 4*) considers a data matrix X generated from a multivariate Gaussian with mean $\mathbf{0}_p$ and block-diagonal covariance matrix $\Sigma = \mathrm{bdiag}(\tilde{\Sigma}, \dots, \tilde{\Sigma})$, where each $\tilde{\Sigma} = (\tilde{\sigma}_{ij})_{i,j}^{5}0$, with $\tilde{\sigma}_{ij} = 0.9$, $i \neq j$ and $\tilde{\sigma}_{ii} = 1$. The vector of coefficients β_0 has 6 nonzero entries, one for each of the first 6 blocks $\tilde{\Sigma}$. These coefficients are $\frac{1}{\sqrt{3}}\{-2.5, -2, -1.5, 1.5, 2, 2.5\}$, assigned at positions $\{1, 51, 101, 151, 201, 251\}$. The authors consider a setting where $n = 100$ and $p = 1000$, and responses are generated as $y \sim \mathcal{N}(X\beta_0^*, \sigma^2)$. This setting is particularly challenging for variable selection tasks, as the high correlation in each block makes it difficult to identify the true components of the model.

4.1.1 Convergence of the chains

The first simulation considers *scenario 1*, having two strongly correlated variables. Data were simulated for $n = 100$ and $p = 1000$, and the three algorithms were run for 3 repetitions with the same data. It can clearly be seen from Figure 4.1 that the performance in terms of convergence of the three algorithms differs dramatically. A similar plot is present in Zanella and Roberts (2019). While wTGS is consistent in its estimates of the PIPs across different repetitions, GS seems to get stuck in one of the two strongly correlated variables, and TGS converges very slowly. This should be attributed to the very large p, since every variable in GS and TGS is updated with uniform probability over $\{1, \dots, p\}$, inducing fewer updates on the two variables with high PIP as compared to wTGS. The argument is confirmed by the trace plots in Figure 4.2, displaying clearly the frequency of the updates for each of the two variables in 1,000 illustrative iterations of the chains.

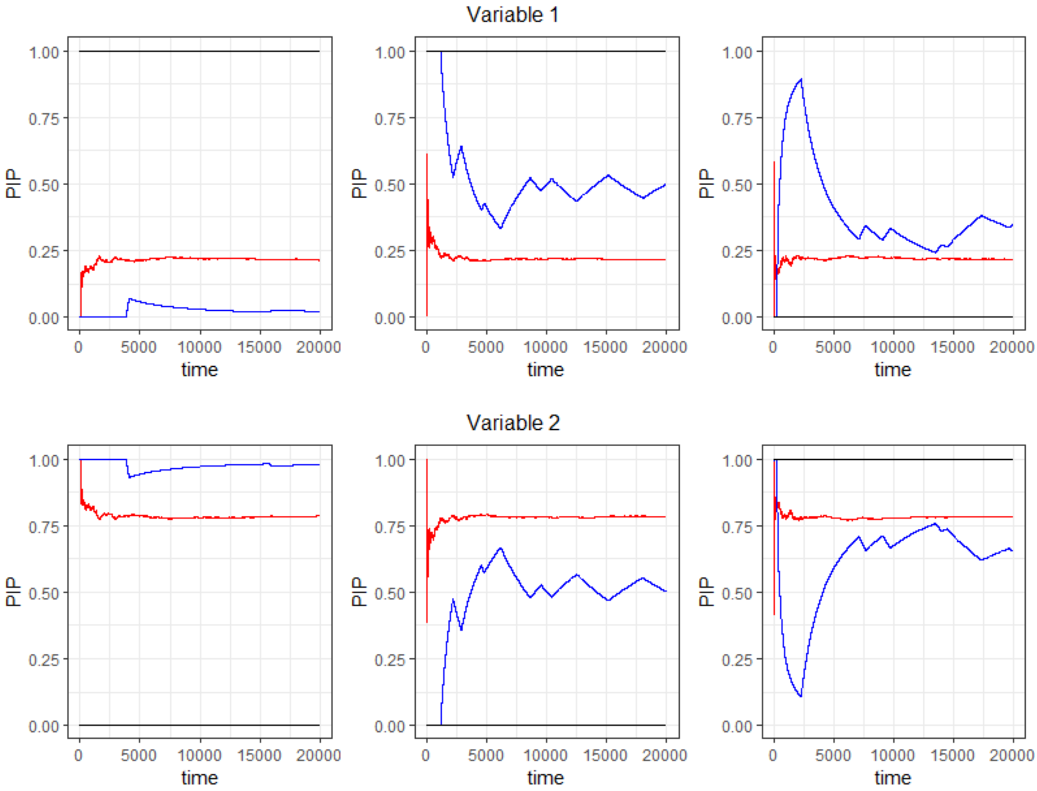

Figure 4.1: Estimated posterior inclusion probabilities (PIPs) across 20,000 iterations of a GS (black line), TGS (blue line), and wTGS (red line). Each of the three columns represent a run of the algorithms, keeping the same data across runs.

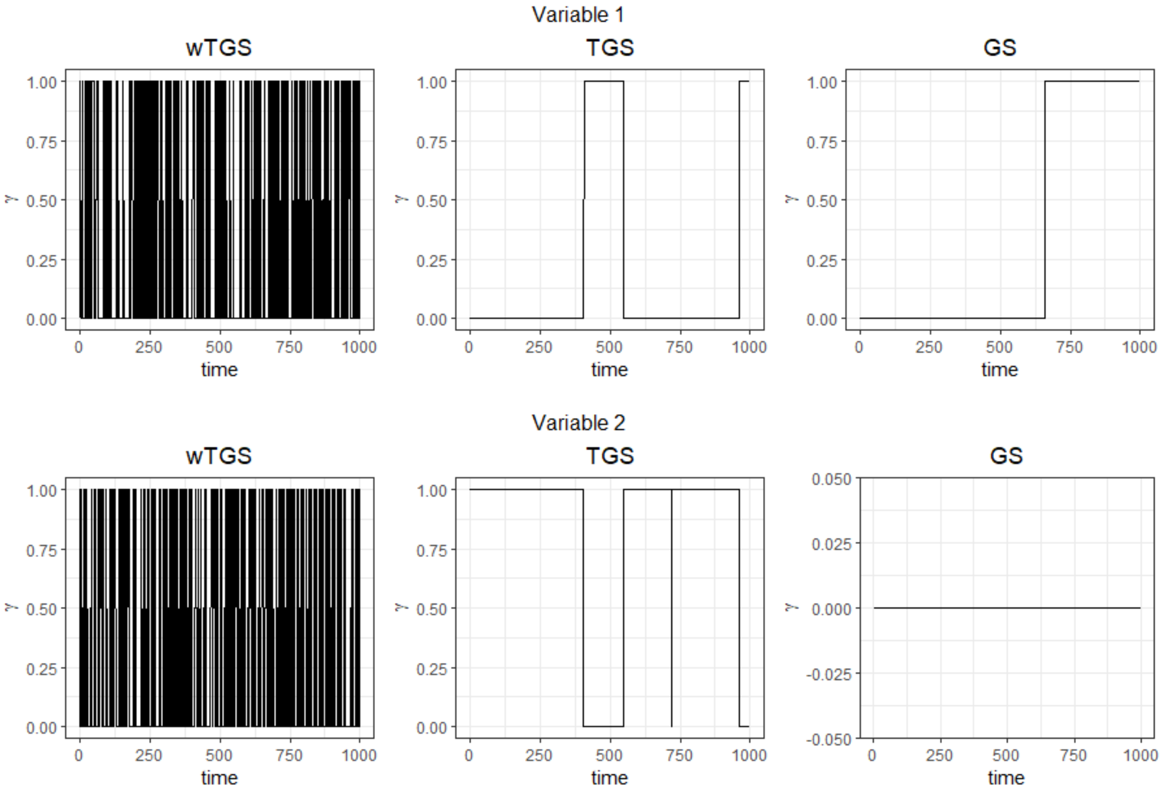

Figure 4.2: Trace plots for two strongly correlated binary variables γ_1 and γ_2, when $p = 1000$.

4.1.2 Full comparison

Following a setting similar to Ročková and George (2018) and Bai et al. (2021), an experiment was conducted by averaging a set of quantities of interest across 50 repetitions of the algorithms. Such quantities are the mean squared error (MSE) with respect to the true coefficients, that is $\frac{1}{p}||\beta_0^* - \hat{\beta}||_2^2$, the false discovery rate (FDR), false nondiscovery rate (FNR), and Matthews correlation coefficient (MCC), defined as

$$FDR = \frac{FP}{TN + FP}, \quad FNR = \frac{FN}{TP + FN} \tag{4.1}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.2}$$

for TP, FP, TN, FN being true and false positives and true and false negatives, defined for the inclusion variables γ with respect to the true data generating scheme. The MCC coefficient is particularly interpretable, as it ranges from -1 to 1 indicating lower to higher degrees of correct classification. For the MSE, the point estimates $\hat{\beta}$ correspond to the conditional expectation $\mathbb{E}[\beta|\hat{\gamma}_M, Y, \sigma^2] = \frac{c}{1+c}\tilde{\beta}_{\hat{\gamma}_M}$, where $\hat{\gamma}_M$ is the maximum a posteriori (MAP) and $\tilde{\beta}_{\hat{\gamma}_M}$ is obtained by least squares considering only the variables in $\hat{\gamma}_M$. Other quantities recorded at each run are estimated model size (DIM), the percentage of times the MAP $\hat{\gamma}_M$ corresponded to the true model (TRUE), the average variance across the Monte Carlo estimates of the PIPs (MCVAR), and execution time (TIME).

Table 4.1 displays estimates of such quantities based on 50 repetitions of each algorithm. Each chain was ran for 25,000 iterations, with a burn in of 5,000 iterations. While the data matrix X and the true coefficients β_0^* were kept fix, the response vector Y used by the three algorithms was generated anew at each run, in order to account for the variability induced by the error term in Y. The table provides numerous insights about the algorithms. For the first two scenarios considered in Zanella and Roberts (2019), TGS and wTGS outperform the standard GS in every metric, with an execution time that is between two and three times larger. In the uncorrelated case, the three methods perform equivalently in all the metrics assessing convergence to the true model, but estimates obtained with TGS and wTGS present a significantly lower variance by virtue of the Rao-Blackwellized estimators. A particularly promising improvement is observed in the fourth scenario, with several blocks of correlated variables among which only one is actually included in the true model. The four metrics assessing the goodness of the MAP estimate (MSE, FDR, FNR, DIM) show considerable improvements when TGS and wTGS are used, and a noteworthy result is the relative number of times the MAP corresponded to the true model (TRUE). While GS is unable to identify the original components of γ in the 25,000 iterations for which the algorithm was ran, the tempered Gibbs samplers do so roughly one-fifth of the time. This makes them directly comparable to other state-of-the-art techniques like spike-and-slab Lasso, that in its different variations reached the true model between 20 and 23 percent of the times (Ročková and George, 2018).

		MSE	FDR	FNR	MCC	DIM	TRUE	MCVAR	TIME
scenario 1	GS	4.5755	5e-04	0.46	0.5337	1.06	0.52	1.0000	1.0000
	TGS	4.8135	4e-04	0.38	0.6079	1.06	0.58	0.3099	2.0473
	wTGS	4.8135	4e-04	0.38	0.6079	1.06	0.58	0.2392	2.1570
scenario 2	GS	47.0544	0	0.0433	0.9738	5.78	0.74	1.0000	1.0000
	TGS	47.0516	0	0.0333	0.9810	5.82	0.78	0.4922	2.6939
	wTGS	47.0516	0	0.0333	0.9810	5.82	0.78	0.4867	2.6142
scenario 3	GS	47.4723	1e-04	0	0.9953	10.1	0.9	1.0000	1.0000
	TGS	47.4723	1e-04	0	0.9953	10.1	0.9	0.6024	3.3683
	wTGS	47.4723	1e-04	0	0.9953	10.1	0.9	0.6042	3.2826
scenario 4	GS	9.4871	0.0042	0.6933	0.3002	6.04	0.00	1.0000	1.0000
	TGS	4.9712	0.0023	0.3900	0.6086	5.98	0.20	0.4260	2.7669
	wTGS	4.9421	0.0023	0.3800	0.6187	5.98	0.22	0.3551	2.7484

Table 4.1: Comparison of quantities of interest averaged over 50 repetitions of each algorithm. MCVAR and TIME are displayed relatively to GS.

Chapter 5

Future developments

While quicker rates of convergence have been empirically observed for the TGS and wTGS algorithms in comparison to a standard Gibbs sampler, the main limitation of the algorithms lies in computational cost. Operations related to the requirement of recomputing the full conditionals for every element of γ at each step are what increase the cost of each iteration of TGS and wTGS by a factor of $p/|\gamma|$ if compared to a standard GS, as seen in section 3.2.2. In the setting under consideration, that is for high-dimensional problems where $p \gg |\gamma|$, the computational burden implied by the algorithms can dramatically impact their efficiency.

5.1 Blocked wTGS

A possible solution would be to induce sparsity by a cheap method, that rules out quickly a subset of variables having inclusion probability close to zero, and to identify a set of *promising* variables. Denoting such a set as S and its dimensionality as m, where $m < p$, it would then be possible to run a Markov chain resembling the one induced by wTGS, but updating only the full conditionals of the variables γ_i in the set S of promising variables. The requirement to compute full conditionals for all of the p elements of γ at each iteration would then be lifted, resulting in a reduced computational complexity by a factor of m/p compared to wTGS. In order to ensure convergence to the posterior $f(\gamma) = p(\gamma|Y)$ of the new algorithm, simulations from the new Markov chain would need to be sampled in an alternate way from the one induced by the original wTGS of Algorithm 4 targeting all variables. The resulting Markov kernel would then resemble a composition of kernels as seen in Section 2.2.2, with the composition either being a cycle as in 2.4 or a mixture as in 2.3. While the complete wTGS would have a computational complexity of $\mathcal{O}(nps)$ (see Section 3.2.2), the kernel focusing on the reduced target would have a cost that is reduced by a factor of m/p, translating in a $\mathcal{O}(nms)$ cost. The complexity of the composition of kernels would still be dominated by the cost of the component targeting the entire set of variables, but devising a clever updating strategy would boost the speed of convergence for the variables in the set S.

Let $P_A(\gamma, \gamma')$ denote the transition matrix of the discrete-time Markov chain for the wTGS procedure (with $k = 0$ for ease of notation). As presented in the supplement to Zanella and Roberts (2019), entries are $P_A(\gamma, \gamma') = 0$ when $\gamma_{-i} \neq \gamma'_{-i}$, whereas when $\gamma_{-i} = \gamma'_{-i}$,

$$P_A(\gamma, \gamma') = \frac{p_i(\gamma)}{\sum_{j=1}^{p} p_j(\gamma)} g(\gamma'_i|\gamma'_{-i}) = \frac{p(\gamma_i = 1|\gamma_{-i})}{4\sum_{j=1}^{p} \frac{p(\gamma_j=1|\gamma_{-j},Y)}{2p(\gamma_j=1|\gamma_{-j},Y)} p(\gamma_i|\gamma_{-i}, Y)}. \tag{5.1}$$

The transition kernel is fZ-invariant, as it can be easily proven to be fZ-reversible.

Let $P_B(\gamma, \gamma')$ denote the transition matrix of the Markov chain for the subroutine focusing on the variables in the set S. From Theorem 6, it is sufficient to devise an algorithm satisfying invariance with respect to fZ for the mixture of kernels $P = \alpha P_A + (1-\alpha)P_B$ to convergence asymptotically to fZ, as the irreducibility of $P_A(\gamma, \gamma')$ would provide a sufficient condition. Future lines of research on the topic could be focused on devising a routine such that the discrete-time Markov chains satisfies fZ-reversibility. Another possibility would be to consider a continuous-time version of the Markov chain.

This version has been considered for TGS and wTGS in Section 4.5.1 of Zanella and Roberts (2019), and the jump matrix for the continuous-time Markov chain of the latter algorithm is defined as $Q_A(\gamma, \gamma') = Z(\gamma)P_A(\gamma, \gamma')$. Let $s = \mathbb{E}_f[|\gamma|]$, then

$$Q_A(\gamma, \gamma') = \frac{1}{s} \sum_{i=1}^{p} \frac{p(\gamma_i = 1|\gamma_{-i}, Y)}{p(\gamma_i|\gamma_{-i}, Y)} \mathbb{1}(\gamma_{-i} = \gamma'_{-i}) \qquad \gamma' \neq \gamma;\ \gamma', \gamma \in \{0,1\}^p, \tag{5.2}$$

and the diagonal entries would be $Q_A(\gamma, \gamma) = -\sum_{\gamma' \neq \gamma} Q_A(\gamma, \gamma')$.

5.1.1 Continuous-time Markov chains for variable selection

The idea of using continuous-time Markov chain Monte Carlo samplers for variable selection problems for Bayesian linear regression problems dates back to Stephens (2000). His approach is to use a *birth and death* MCMC, but while he outlined a possible general approach for continuous-time MCMC, he did not specialize it to BVS, giving only a general idea of the problem setting. The version reported below connects such a setting presented in Stephens (2000) with the general approach as it was reported in Robert and Casella (2004).

For p possible variables each associated to a coefficient $\beta_i \in \mathbb{R}, i = 1, \ldots, p$, a model containing s many variables can be represented by a set of s points $\{(i_1, \beta_{i_1}), \ldots, (i_s, \beta_{i_s})\}$ in $\{1, \ldots, p\} \times \mathbb{R}$ where each $i_j \in \{1, \ldots, p\}, j = 1, \ldots, s$ and $i_j \neq i_h,\ \forall\ j, h$. For short, let θ_s indicate such a set of points. The prior for i being in the model ($\gamma_i = 1$ in the usual notation) is defined as $p(i)$ and it is independent for all i, while the prior for β_i conditionally on i being included is $p(\beta_i)$, independent for all i. The joint distribution is then defined as

$$\pi(s, \theta_s) = \begin{cases} 0, & \text{if } i_a = i_b \text{ for some } a, b, \\ p(i_1)p(\beta_{i_1}) \ldots p(i_s)p(\beta_{i_s}), & \text{otherwise.} \end{cases} \tag{5.3}$$

A birth in the process corresponds to the inclusion in the model of a new point (i, β_i), while death to an exclusion from the model.

Following results from Green (1995), switching between configurations of different dimensionality is possible by imposing a *dimension matching* condition, that can be outlined as follows. Let $s < \ell$, then the dimensions of the two configurations θ_s and θ_ℓ are matched by sampling a random vector $u_{s,\ell}$ of length m_1 from a known density $\phi_{s,\ell}(u_{s,\ell})$, independent of θ_s. The configuration θ_s and the simulated $u_{s,\ell}$ are then mapped to a new state θ_ℓ by a function $T_{s\ell}(\theta_s, u_{s\ell})$. If the reverse move were to be performed, it would similarly be possible to obtain θ_s by a deterministic function $T_{\ell s}(\theta_\ell, u_{\ell s})$, with $u_{\ell s}$ being a vector of length m_2 sampled from $\phi_{\ell,s}(u_{\ell,s})$. The condition for dimension matching is then satisfied if the mapping between $(\theta_\ell, u_{\ell s})$ and $(\theta_s, u_{s\ell})$ is bijective, and the lengths of $u_{\ell s}$ and $u_{s\ell}$ satisfy $\ell + m_2 = s + m_1$.

The time spent by the continuous-time Markov chain at configuration θ_s is $V \sim \text{Exp}(q_s)$, with q_s depending on θ_s, and then a switch is made according to the transition kernel. Transition rates include a term corresponding to the determinant of the Jacobian matrix of $T_{s\ell}(\theta_s, u_{s\ell})$, arising from the change of variables by the function $T_{s\ell}$ (Fan and Sisson, 2011). The procedure is described in Algorithm 5.

Algorithm 5 Continuous-time MCMC

```
For $\theta^{(t)} = (s, \theta_s)$:
1: For each $\ell \in \{1, \dots, p\}$, draw $u_{s,\ell} \sim \phi_{s,\ell}$ satisfying the dimension matching condition.
2: Compute transition rates
       $$q_{s\ell} = \frac{p(y|\theta_\ell)\pi(\ell, \theta_\ell)}{p(y|\theta_s)\pi(s, \theta_s)} \frac{\phi_{\ell,s}(u_{\ell,s})}{\phi_{s,\ell}(u_{s,\ell})} \left| \frac{\partial T_{s\ell}(\theta_s, u_{s\ell})}{\partial(\theta_s, u_{s\ell})} \right|$$
3: Compute
       $$q_s = q_{s1} + \cdots + q_{sp}.$$
4: Generate arrival time $t + V$, for $V \sim \text{Exp}(q_s)$.
5: Select the move to $j \in \{1, \dots, p\}$ with probability $q_{sj}/q_j$.
6: Set time at $t = t + V$, set $\theta^{(t)} = (j, \theta_j)$
```

Moves are always accepted, but configurations with low posterior probability die sooner in the process than ones with higher probability, as the transition rates are directly influenced by such quantities. Since $\mathbb{E}[V] = 1/q_s$, a larger $p(y|\theta_\ell)\pi(\ell, \theta_\ell)$ reduces the time until a jump to configuration θ_ℓ, making such configuration more likely to be visited in the path of the chain.

Ensuring that the Markov process is invariant to a distribution proportional to the posterior amounts to verifying that the *local detailed balance* is satisfied (Cappé et al., 2003), that is

$$p(y|\theta_s)\pi(s, \theta_s)q_{s\ell}(\theta_s, \theta_\ell) = p(y|\theta_\ell)\pi(\ell, \theta_\ell)q_{\ell s}(\theta_\ell, \theta_s). \tag{5.4}$$

5.1.2 Future research

Devising a method to perform wTGS with a Markov chain in continuous time would amount to defining a transition rate that is a mixture between a chain targeting the whole set of variables as in 5.2 and one targeting only those in S. Denoting the latter as Q_B, the new jump matrix would then have non-diagonal entries that are

$$Q(\gamma, \gamma') = Q_A(\gamma, \gamma') + Q_B(\gamma, \gamma')$$

and diagonal $Q(\gamma, \gamma) = \sum_{\gamma' \neq \gamma} Q(\gamma, \gamma')$. Future lines of work could focus on devising a method to sample from the continuous-time Markov chain just described while making sure that the composition satisfies the local detailed balance condition. This would ensure f-invariance of the chain, and in turn, convergence to stationarity.

Chapter 6

Conclusion

The thesis has shown the advantages of the computational methods related to tempered Gibbs sampling techniques proposed in Zanella and Roberts (2019), providing a comprehensive review of the theoretical results that are foundational to the context of Bayesian variable selection. Particular attention was directed to the practical and methodological aspects of the task while justifying the usage of the algorithms under consideration. An alternative implementation based on some basic results from numerical linear algebra was presented in Section 3.1.2, with the potential of easing the implementation of the method when both the dimensionality of the features and the number of relevant ones are large. The study on simulated data in Section 4.1.2 has highlighted the practical advantage of the algorithm over traditional Gibbs sampling techniques in high-dimensional settings, showing a significantly higher ability to accurately identify the set of predictors in a Bayesian linear regression model. The performance in such a task was shown to be comparable to that of other state-of-the-art methods like the spike-and-slab Lasso by Ročková and George (2018). Finally, directions for possible extensions of the model that would leverage properties of continuous-time Markov chains were proposed in Section 5.1. Such techniques are particularly worthy of exploration, as they could reduce the computational cost of the scheme if implemented as generally outlined in the section.

Bibliography

Edward Anderson, Zhaojun Bai, Christian Bischof, L Susan Blackford, James Demmel, Jack Dongarra, Jeremy Du Croz, Anne Greenbaum, Sven Hammarling, Alan McKenney, et al. *LAPACK Users' guide.* SIAM, 1999.

Ray Bai, Veronika Ročková, and Edward I. George. Spike-and-slab meets LASSO: A review of the spike-and-slab LASSO. In *Handbook of Bayesian Variable Selection*, pages 81–108. Chapman and Hall/CRC, dec 2021. doi: 10.1201/9781003089018-4. URL `https://doi.org/10.1201%2F9781003089018-4`.

Matthew James Beal. *Variational algorithms for approximate Bayesian inference.* University of London, University College London (United Kingdom), 2003.

José M Bernardo and Adrian FM Smith. *Bayesian theory*, volume 405 of *Wiley Series in Probability and Statistics.* John Wiley & Sons, 1994.

Peter Bühlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications.* Springer Science & Business Media, 2011.

Olivier Cappé, Christian P Robert, and Tobias Rydén. Reversible jump, birth-and-death and more general continuous time markov chain monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(3):679–700, 2003.

Kung Sik Chan and Charles J Geyer. Discussion: Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4):1747–1758, 1994.

Hugh Chipman, Edward I George, Robert E McCulloch, Merlise Clyde, Dean P Foster, and Robert A Stine. The practical implementation of bayesian model selection. *Lecture Notes-Monograph Series*, pages 65–134, 2001.

Nicolas Chopin, Omiros Papaspiliopoulos, et al. *An introduction to sequential Monte Carlo*. Springer, 2020.

Y.S. Chow, Y.S. Chow, and H. Teicher. *Probability Theory: Independence, Interchangeability, Martingales*. Springer texts in statistics. World Publishing Company, 1988. ISBN 9783540966951. URL `https://books.google.it/books?id=FOeePwAACAAJ`.

George Deligiannidis and Anthony Lee. Which ergodic averages have finite asymptotic variance? *The Annals of Applied Probability*, 28(4):2309–2334, 2018.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

Yanan Fan and Scott A Sisson. Reversible jump mcmc. *Handbook of Markov Chain Monte Carlo*, pages 67–92, 2011.

Carmen Fernández, Eduardo Ley, and Mark F.J. Steel. Benchmark priors for bayesian model averaging. *Journal of Econometrics*, 100(2):381–427, 2001. ISSN 0304-4076. doi: https://doi.org/10.1016/S0304-4076(00)00076-2. URL `https://www.sciencedirect.com/science/article/pii/S0304407600000762`.

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

Edward I George and Robert E McCulloch. Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

Edward I George and Robert E McCulloch. Approaches for bayesian variable selection. *Statistica sinica*, pages 339–373, 1997.

Peter J Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

William W Hager. Updating the inverse of a matrix. *SIAM review*, 31(2):221–239, 1989.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

Alan Miller. *Subset selection in regression.* chapman and hall/CRC, 2002.

Toby J Mitchell and John J Beauchamp. Bayesian variable selection in linear regression. *Journal of the american statistical association*, 83(404):1023–1032, 1988.

Naveen Naidu Narisetty and Xuming He. Bayesian variable selection with shrinking and diffusing priors. *The Annals of Statistics*, 42(2):789–817, 2014.

Esa Nummelin. *General Irreducible Markov Chains and Non-Negative Operators.* Cambridge Tracts in Mathematics. Cambridge University Press, 1984. doi: 10.1017/CBO9780511526237.007.

Trevor Park and George Casella. The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

Christian P Robert and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 2004.

Gareth O Roberts and Jeffrey S Rosenthal. General state space markov chains and mcmc algorithms. *Probability surveys*, 1:20–71, 2004.

Veronika Ročková and Edward I. George. The spike-and-slab lasso. *Journal of the American Statistical Association*, 113(521):431–444, 2018. doi: 10.1080/01621459.2016.1260469. URL `https://doi.org/10.1080/01621459.2016.1260469`.

Havard Rue and Leonhard Held. *Gaussian Markov random fields: theory and applications*. Chapman and Hall/CRC, 2005.

Mark J Schervish. *Theory of statistics*. Springer Series in Statistics. Springer-Verlag, 1995.

Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

Michael Smith and Robert Kohn. Nonparametric regression using bayesian variable selection. *Journal of Econometrics*, 75(2):317–343, 1996. URL `https://EconPapers.repec.org/RePEc:eee:econom:v:75:y:1996:i:2:p:317-343`.

Matthew Stephens. Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods. *Annals of statistics*, pages 40–74, 2000.

Gilbert W Stewart. *Matrix algorithms: volume 1: basic decompositions*. SIAM, 1998.

Mahlet G Tadesse and Marina Vannucci. *Handbook of Bayesian variable selection*. Chapman & Hall, CRC Handbooks of Modern Statistical Methods, 2021.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Luke Tierney. Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728, 1994.

Luke Tierney and Joseph B Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the american statistical association*, 81(393):82–86, 1986.

Giacomo Zanella and Gareth Roberts. Scalable importance tempering and bayesian variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81(3):489–517, 2019. doi: 10.1111/rssb.12316. URL `https://doi.org/10.1111%2Frssb.12316`.

Arnold Zellner. On assessing prior distributions and bayesian regression analysis with g prior distributions. *Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti. Studies in Bayesian Econometrics and Statistics*, pages 233–243, 1986.

Appendix A

Alternative implementations

A.1 Recursive Cholesky updates

Consider a positive semidefinite matrix of the type $A = X'X$. Let C be the lower triangular matrix satisfying $CC' = A$, namely the Cholesky factor. If a column at the j^th position is removed from X, it is possible to obtain updates of the Cholesky factors as follows. It should first be noted that in the case where j is the last column, the new Cholesky factor $\tilde{C}$ can be obtained by simply removing the last row and the last column.

$$\begin{aligned} X'X &= \begin{bmatrix} X_{1:j-1} & x_j & X_{j+1:n} \end{bmatrix}' \begin{bmatrix} X_{1:j-1} & x_j & X_{j+1:n} \end{bmatrix} \\ &= \begin{bmatrix} X'_{1:j-1}X_{1:j-1} & X'_{1:j-1}x_j & X'_{j+1:n}X_{1:j-1} \\ x'_jX_{1:j-1} & x'_jx_j & x'_jX_{j+1:n} \\ X'_{j+1:n}X_{1:j-1} & X_{1:j-1}x_j & X'_{j+1:n}X_{j+1:n} \end{bmatrix} \\ &= \begin{bmatrix} C_{11} & & \\ c_{12} & c_{22} & \\ C_{31} & c_{32} & C_{33} \end{bmatrix}' \begin{bmatrix} C_{11} & & \\ c_{12} & c_{22} & \\ C_{31} & c_{32} & C_{33} \end{bmatrix} \end{aligned}$$

Deleting column j we get

$$A = \begin{bmatrix} \tilde{C}_{11} & \\ \tilde{C}_{21} & \tilde{C}_{22} \end{bmatrix}' \begin{bmatrix} \tilde{C}_{11} & \\ \tilde{C}_{21} & \tilde{C}_{22} \end{bmatrix}$$

where

$$\begin{aligned} \tilde{C}_{11} &= C_{11} \\ \tilde{C}_{21} &= C_{31} \\ \tilde{C}_{22}\tilde{C}_{22}' &= C_{33}C_{33}' + c_{32}c_{32}', \end{aligned}$$

and the last operation is a rank-one update that can be computed efficiently. Derivations for the computation of recursive updates follow Anderson et al. (1999), and alternative implementations can be obtain by applying Givens rotations to C after having removed the j^{th} row and column. What follows is the R code for the operations described above, and the rank-one update is based on the code by Stewart (1998). The same result is attainable by using the function `choldrop` of the `mgcv` library. The latter procedure is faster, as it is implemented in C.

```
cholupdate <- function(S,k){
  S <- t(S)
  K <- dim(S)[1]
  if (K==k){
    R <- S[1:(k-1),1:(k-1)]
    return(t(R))
  }
  x <- S[(k+1):K,k]
  L <- S[(k+1):K,(k+1):K]
  n <- length(x)
  if (K==k+1){
    r <- sqrt(L^2 + x^2)
    L <- r
  }else{
  for (i in 1:n){
      r <- sqrt(L[i, i]^2 + x[i]^2)
      c <- r / L[i, i]
      s <- x[i] / L[i,i]
      L[i, i] <- r
      if (i < n){
        L[(i+1):n, i] <- (L[(i+1):n, i] + s * x[(i+1):n]) / c
        x[(i+1):n] <- c * x[(i+1):n] - s * L[(i+1):n, i]
      }
    }
  }
  R <- zeros(K-1,K-1)
  R[1:(k-1),1:(k-1)] <- S[1:(k-1),1:(k-1)]
  R[k:(K-1),1:(k-1)] <- S[(k+1):K,1:(k-1)]
  R[k:(K-1),k:(K-1)] <- L
  return(t(R))
}
```

A.2 Full conditional with no inversion

The following code, based on the one provided in the supplement to Zanella and Roberts (2019), computes the full conditionals in 3.1.2 without the need to invert the matrix $X'_\gamma X_\gamma$, using instead Cholesky decomposition and recursive updates of Cholesky factors. The inputs of the function full_cond are gamma, indicating whether each variable is included or excluded, and hyper_par, containing the matrices $X'X$, $Y'X$, and $Y'Y$, and the parameters c and h defined as in 3.6. The function returns a vector of full conditional probabilities for each γ_i, $i = 1, \dots, p$. The ancillary function ch_noinv replicates the procedure described in Algorithm 1, taking as input an upper triangular Cholesky factor and the vector or matrix V. The output is $V'(X'_\gamma X_\gamma)^{-1}V$. The function ch_noinv_bilinear takes an upper triangular Cholesky as before, two vectors or matrices W and V, and returns $W(X'_\gamma X_\gamma)^{-1}V$.

Comparisons of the different routines to obtain full conditionals are reported in Table A.1, with data simulated for $n = 500$ and a dependency structure equal to that of *scenario 1* in Section 4. The schemes were tested for different choices of p and of $|\gamma|$, with the latter being set by defining a variable w such that $p(\gamma_i = 1) = w, i = 1, \dots, p$.

		$w = 0.1$		$w = 0.05$		$w = 0.01$	
		elapsed	relative	elapsed	relative	elapsed	relative
$p = 500$	No inversion, package	37.94	1.000	10.80	1.000	1.38	1.290
	No inversion, no package	225.67	5.948	56.72	5.252	2.55	2.383
	Inversion	44.48	1.172	10.89	1.008	1.07	1.000
$p = 250$	No inversion, package	6.61	1.025	2.05	1.079	0.67	1.175
	No inversion, no package	37.95	5.884	6.81	3.584	1.05	1.842
	Inversion	6.45	1.000	1.90	1.000	0.57	1.000
$p = 100$	No inversion, package	2.17	1.212	0.88	1.189	0.20	1.053
	No inversion, no package	8.25	4.609	1.75	2.365	0.19	1.000
	Inversion	1.79	1.000	0.74	1.000	0.23	1.211

Table A.1: Time comparison over 2,000 runs for computing the full conditionals. "No inversion, no package" rows are computed with the code reported below, "No inversion, no package" rows are computed using the `mgcv` package, "Inversion" rows are computed with the original version of the code.

Functions for computing full conditionals

```
full_cond<-function(gamma,hyper_par=NULL){
  fc<-rep(NA,length(gamma))
  included<-which(gamma==1)
  p_gamma<-sum(gamma)
  n<-hyper_par$n
  p<-length(gamma)
  h<-hyper_par$prior_p_incl
  c<-hyper_par$c
  yty<-hyper_par$yty
  ytX<-hyper_par$ytX
  XtX<-hyper_par$XtX
  if(p_gamma==0){
    S_old<-yty
    for (j in 1:p){
      S_new<-yty-c/(1+c)*ytX[j]^2/XtX[j,j]
      post_ratio<-h/(1-h)*(S_old/S_new)^(n/2)/(sqrt(1+c))
      fc[j]<-1/(1+post_ratio)
    }
  }else{
    ytX_gamma<-matrix(ytX[included],nrow = p_gamma,ncol = 1)
    U = chol(XtX[included,included])
    ytXFXty<-as.numeric(ch_noinv(U,ytX_gamma))
    S_old<-yty-c/(1+c)*ytXFXty
    P = ch_noinv00(U,XtX[included,])
    d_vec<-1/(diag(XtX)-P$B+10e-10)
    ytXFXtxj_vec<-t(ytX_gamma)%*%P$W
    S_new_vec<-yty-c/(1+c)*(ytXFXty+d_vec*(ytXFXtxj_vec-ytX)^2)
    post_ratio_vec<-h/(1-h)*(S_old/S_new_vec)^(n/2)/(sqrt(1+c))
    fc<-1/(1+post_ratio_vec)
    for (j in included){
      if(p_gamma==1){
        S_new<-yty
      }else{
        pos_rem<-which(included==j)
        included_new<-included[-pos_rem]
        ytX_gamma_new<-matrix(ytX[included_new],nrow = 1,ncol = p_gamma-1)
        U_star<- cholupdate(U,pos_rem)
        ytXFXty_new<-ch_noinv(U_star,t(ytX_gamma_new))
        S_new<-yty-c/(1+c)*ytXFXty_new
      }
      post_ratio<-(1-h)/h*(S_old/S_new)^(n/2)*(sqrt(1+c))
      fc[j]<-1/(1+post_ratio)
    }
  }
  output<-list(fc=fc)
  return(output)
}

ch_noinv<-function(U,v){
  if (length(U)==1){
    output<-drop(U)^(-2)*(t(v)%*%v)
    return(output)
  }
  z = forwardsolve(t(U),v)
  w = backsolve(U,z)
  output<-t(v)%*%w
  return(output)
}

ch_noinv_bilinear<-function(U,w,v){
  if (length(U)==1){
    output<-drop(U)^(-2)*(w%*%v)
    return(output)
  }
  z = forwardsolve(t(U),v)
  w = backsolve(U,z)
  output<-w%*%w
  return(output)
}
```

A.3 TGS for a bivariate normal distribution

Let the random vector $(Y_1, Y_2) \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, where $\boldsymbol{\mu} = [0 \quad 0]$ and $\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$. The following R functions enable to sample from the joint distribution of (Y_1, Y_2) using a random-scan GS and a TGS with tempered proposal. In order to generate proposals from a tempered version of the full conditionals, a rejection sampling approach is used, with a uniform distribution covering most of the domain of the tempered full conditional, that is rescaled by an (arbitrary) factor, here set to β, such that the whole distribution lies below the uniform. An alternative (and more efficient) version could be implemented by computing analytically the normalization constant related to the tempered full conditional. In `TGS_normal`, the tempering coefficient β is automatically set to be $1 - \rho$.

```
dnorm_beta <- function(beta,x,mu=0,sd=1){
  return(dnorm(x,mu,sd)^beta)
}

rejection_sampling <- function(beta,r,mu=0,sd=1){
  x <- runif(1,-r,r)
  u <- runif(1,0,dunif(x,-r,r))
  if (u<(dnorm_beta(beta,x,mu,sd)*beta)){
    return(x)
  }else{
    rejection_sampling(beta,r,mu,sd)
  }
}

pfull_cond_normal <- function(i,x,mu,Sigma){
  j <- -i+3
  mean <- mu[i] + Sigma[i,j]/Sigma[j,j]*(x-mu[j])
  var <- Sigma[i,i] - Sigma[i,j]^2/Sigma[j,j]
  return(c(mean,sqrt(var)))
}
```

```
rfull_cond_normal <- function(i,x,mu,Sigma){
  j <- -i+3
  mean <- mu[i] + Sigma[i,j]/Sigma[j,j]*(x-mu[j])
  var <- Sigma[i,i] - Sigma[i,j]^2/Sigma[j,j]
  return(rnorm(1,mean,sqrt(var)))
}

GS_normal <- function(mu,Sigma,first_fixed=NA,T=100,burn_in=10){
  if ((NA %in% first_fixed)==TRUE){
    y<-rnorm(2)
  }else{
    y<-first_fixed
  }
  sample<-data.frame(y1=rep(NA,T-burn_in),y2=rep(NA,T-burn_in))
  sample[1,c(1,2)]<-y
  for (t in 2:T){
    u <- runif(1)
    if (u<1/2){
      i <- 1
    }else{
      i <- 2
    }
    j <- -i+3
    y[i] <- rfull_cond_normal(i,y[j],mu,Sigma)
    if (t>burn_in){
      sample[t,c(1,2)]<-y
    }
  }
  return(sample)
}

TGS_normal <- function(mu,Sigma,first_fixed=NA,T=100,burn_in=10){
  if ((NA %in% first_fixed)==TRUE){
    y<-rnorm(2)
  }else{
    y<-first_fixed
  }
  sample<-data.frame(y1=rep(NA,T-burn_in),y2=rep(NA,T-burn_in),
    weight=rep(NA,T-burn_in))
  par1 <- pfull_cond_normal(1,y[2],mu,Sigma)
  g1<-dnorm_beta(1-Sigma[1,2],y[1],par1[1],par1[2])
  par2 <-pfull_cond_normal(2,y[1],mu,Sigma)
  g2<-dnorm_beta(1-Sigma[1,2],y[2],par2[1],par2[2])
  p <- c(g1/dnorm(y[1],par1[1],par1[2]),g2/
    dnorm(y[2],par2[1],par2[2]))
  print(c(par1,g1,par2,g2))
  w <- 2/sum(p)
  sample[1,c(1,2)] <- y
  sample[1,3] <- w
  for (t in 2:T){
    u <- runif(1)
    if (u<p[1]/sum(p)){
      i <- 1
    }else{
      i <- 2
    }
    j <- -i+3
    par <- pfull_cond_normal(i,y[j],mu,Sigma)
    y[i] <- rejection_sampling(1-Sigma[1,2],50,par[1],par[2])
    parj <- pfull_cond_normal(j,y[i],mu,Sigma)
    p[i] <- dnorm_beta(1-Sigma[1,2],y[i],par[1],par[2])/
        dnorm(y[i],par[1],par[2])
    p[j] <- dnorm_beta(1-Sigma[1,2],y[j],parj[1],parj[2])/
        dnorm(y[j],parj[1],parj[2])
    w <- 2/sum(p)
    if (t>burn_in){
      sample[t,c(1,2)] <- y
      sample[t,3] <- w
    }
  }
  sample["weight"] <- sample["weight"]/sum(sample["weight"])
  return(sample)
}
```