Bayesian adaptive variable selection in linear models: A generalization of Zellner's informative g-prior

Mémoire

Djibril Ndiaye

Master thesis, Statistics
Maître ès sciences (M. Sc.)

Québec, Canada

© Djibril Ndiaye, 2021

Bayesian adaptive variable selection in linear models: A generalization of Zellner's informative g-prior

Mémoire

Djibril Ndiaye

Sous la direction de:

Khader Khadraoui

Abstract

Bayesian inference is about recovering the full conditional posterior distribution of the parameters of a statistical model. This exercise, however, can be challenging to undertake if the model specification is not available a priori, as is typically the case. This thesis proposes a new framework to select the subset of regressors that are the relevant features that explain a target variable in linear regression models. We generalize Zellner's g-prior with a random matrix, and we present a likelihood-based search algorithm, which uses Bayesian tools to compute the posterior distribution of the model parameters over all possible models generated, based on the maximum a posteriori (MAP). We use Markov chain Monte Carlo (MCMC) methods to gather samples of the model parameters and specify all distributions underlying these model parameters. We then use these simulations to derive a posterior distribution for the model parameters by introducing a new parameter that allows us to control how the selection of variables is done. Using simulated datasets, we show that our algorithm yields a higher frequency of choosing the correct variables and has a higher predictive power relative to other widely used variable selection models such as adaptive Lasso, Bayesian adaptive Lasso, and relative to well-known machine learning algorithms. Taken together, this framework and its promising performance under various model environments highlight that simulation tools and Bayesian inference methods can be efficiently combined to deal with well-known problems that have long loomed the variable selection literature.

Résumé

L'inférence bayésienne consiste à retrouver la distribution conditionnelle a posteriori complète des paramètres d'un modèle statistique. Cet exercice, cependant, peut être difficile à entreprendre si la spécification du modèle n'est pas disponible a priori, comme c'est généralement le cas. Cette thèse propose une nouvelle approche pour sélectionner le sous-ensemble de régresseurs qui sont les caractéristiques pertinentes qui expliquent une variable cible dans les modèles de régression linéaire. Nous généralisons le g-prior de Zellner avec une matrice aléatoire et nous présentons un algorithme de recherche basé sur la vraisemblance, qui utilise des outils bayésiens pour calculer la distribution a posteriori des paramètres du modèle sur tous les modèles possibles générés. La sélection du modèle se fera sur la base du maximum a posteriori (MAP). Nous utilisons les méthodes de Monte Carlo par chaînes de Markov pour échantillonner suivant les distributions a posteriori de ces paramètres du modèle. Nous utilisons ensuite ces simulations pour dériver une estimation a posteriori des paramètres du modèle en introduisant un autre paramètre qui nous permet de contrôler la manière dont la sélection de la variable est effectuée. À l'aide de données simulées, nous montrons que notre méthode donne une fréquence plus élevée de choix des variables importantes et a un pouvoir prédictif plus élevé par rapport à d'autres modèles de sélection de variables largement utilisés tels que le Lasso adaptatif, le Lasso adaptatif bayésien, et par rapport aux algorithmes d'apprentissage automatique bien connus. Pris ensemble, cette approche et ses performances prometteuses dans divers scénarios de données mettent en évidence le fait que les outils de simulation et les techniques d'inférence bayésienne puissent être efficacement combinés pour traiter des problèmes bien connus qui ont longtemps pesé sur la littérature de la sélection de variables (en particulier en grande dimension).

Contents

List of Figures

List of Tables

"Dans la nature innée des hommes se trouve le penchant vers la tyrannie et l'opression mutuelle." *** "Une nation s'affaiblit lorsque s'altère et se corrompt le sentiment religieux." *** "L'homme est fils de ses habitudes et de son milieu, et non fils de sa nature et de son mélange d'humeurs."

Ibn Khaldoun (1332 - 1406)

Acknowledgements

I would like to express my deep gratitude to Khader Khadraoui, my thesis advisor, for the countless advice he has given me. I thank him for listening to me, guiding me and motivating me throughout this journey. I also thank him for having trusted me with regard to my research question and for having always pushed me to choose a subject that has a real potential to contribute to the field. I also thank the committee members for the correction of the thesis and the constructive comments.

I would like to thank Université Laval and my supervisor Khader Khadraoui for the financial support that allowed me to focus on my learning and my development as a researcher. I would like to thank the company Intact for offering me an internship with them and for being able to develop my practical knowledge.

This memoir is dedicated to my friend Mor Niang, who unfortunately left us too early when he was in the prime of his life.

Introduction

Model specification is key to conducting inference on the parameters of a statistical model. The widely used linear regression models, for example, represent a set of solutions to the statistical problem of finding unknown distributions of variables using data, under the assumption that a linear combination of the explanatory variables $\boldsymbol{X}$ can be used to derive useful information about the target variable of interest $\boldsymbol{y}$. In this family of models, the Ordinary Least Squares (OLS) regression can be used to derive $E(\boldsymbol{y}|\boldsymbol{X})$ and similar expectations of interest. Bayesian inference with either conjugate prior distributions or non-informative prior distributions can also be conducted to recover the full conditional distribution of the target variable, namely $p(\boldsymbol{y}|\boldsymbol{X})$. For a review of Bayesian variable selection methods, we refer the reader to O'Hara and Sillanpaa (2009).

In practice, however, such Bayesian inference can be challenging to conduct since the model specification is often not available a priori. The unavailability of the model specification could stem from two sources: (i) an inability to separate the set of explanatory variables that have predictive power and should be included in the model from those that should be ignored, and (ii) conditional on knowing the explanatory variables that are relevant to estimate the target variable, an inability to know the true functional form governing the relationship between the target variable and the explanatory variables. How should we approach this model selection problem when we are interested in getting the full distribution of the target variable? What mechanism is needed to select the model structure that would best describe the observed data?

This thesis proposes a new framework to select the subset of regressors that are the relevant features that explain a target variable $\boldsymbol{y}$. We propose a new model that generalizes Zellner's g-prior[1] with a random matrix instead of a scalar. Mathematically, we consider situations in which we have data of the form $\{\boldsymbol{y}, X_1, \ldots, X_p\}$ which contains both useful features $U := \{X_{i_1}, \ldots, X_{i_k}\}$ and irrelevant features $L := \{X_{j_1}, \ldots, X_{j_l}\}$ that are not related to the data (at least not part of the generating process), such that $k + l = p$. The useful features U might be hard to identify and to separate from the unrelated features L because these useful features

[1] In this model, the prior variance of coefficients is $g\sigma^2(\boldsymbol{X'X})^{-1}$ where g is a positive constant and σ^2 is the variance parameter.

are each potentially correlated to some set of features that include variables from both U and L. The scope of the thesis is fourfold: (i) to propose methods to find a good partition $[\hat{U}, \hat{L}]$ of the whole set of regressors that best approximates the true partition of explanatory variables with respect to some objective function to be optimized, (ii) to compare the performance of this method to those of existing variable selection methods widely used in the literature, (iii) to show how this method would perform under complex but practical cases, and (iv) to apply this proposed method to secondary datasets and test/show its generalizability in deriving the posterior distribution of model parameters.

The first characterization of the proposed framework is that it deviates from traditional model optimization algorithms, such as models assuming normally distributed errors, which tend to suffer from the fact that they try to find the mode (which is often not a statistic of interest as much as the mean is) since they transform the variable selection problem into an optimization exercise. These optimization exercises, while they could be sufficient in of themselves, are very limited since they only yield a point estimate of only one statistic and no confidence intervals to rely on to conduct inference (in the case of most machine learning methods). Although traditional statistical methods (like standard linear regression) provide more results than the latter, they are limited by the strong assumptions they make about the features (such as orthogonality) which may fail under certain difficult situations.

Rather than approaching the variable selection problem as an optimization exercise, we present a search algorithm, which uses Bayesian tools to compute the posterior distribution of the model parameters over all possible models and then use them to approximate variables based on the maximum a posteriori (MAP is an estimate of an unknown quantity that equals the mode of the posterior distribution) or other relevant heuristics we choose. A key operational strength of this search algorithm is that it can allow us to calculate a variety of statistics other than the MAP, unlike the optimization algorithms.

Our Bayes variable selection methodology proceeds in the following sequence tying together the model selection and the data generating process (or data modeling): (i) we use Markov chain Monte Carlo (MCMC) methods to gather samples of the parameters of our model and specify all distributions underlying these model parameters, then (ii) we use these simulations to derive a posterior distribution for the model parameters by introducing a new parameter that allows us to control how the selection of variables is done.

In the first step of our selection method, we use MCMC methods to gather samples of the parameters of our model and specify all distributions underlying these model parameters. We consider the generated samples as joint distributions or marginal distributions. For the second step of our variable selection algorithm, we adopt a general framework that derives a posterior distribution based on a uniform prior distribution of a selection variable and on its likelihood. In particular, we consider a hierarchical model and add the selection variable γ

to the structure, with $\gamma \in \{0,1\}^p$ being a vector (of length p) of 0 and 1 indicating whether the corresponding regressor is included or excluded. After generating MCMC samples as a first step, there are various techniques that can be applied in the second step to make it more likely that we end up with something closer to the true distribution of the variables of interest. For example, we can adopt one of the most widely used selection techniques, which consists of including all variables such that the marginal proportion (which is interpreted as the estimated probability that a regressor is selected) is greater than $\frac{1}{2}$. This translates into the probability of belonging in U superior to that of belonging to L.

There is a challenge in this selection approach, namely that the Bayesian selection models yield different samples for different runs, which points to variability within each sample. To tackle this subtle inconsistency, we treat these found proportions as mean random variables coming from a common distribution which is assumed to be a mixture of two simple distributions. Since we are dealing here with a problem involving some unobserved mixtures components, we maximize the likelihood of a mixture of two densities for which we calculate the parameters using an Expectation Maximization (EM) algorithm [2]. We considered a mixture of two Beta distributions and used the Methods of Moments during the expectation step [3]. Maximum Likelihood Estimate (MLE) is another option that can be used as well. In any case, the objective is to find the two appropriate clusters ($\hat{U}$ and $\hat{L}$) that best reproduce the mixture distribution.

After laying out our likelihood-based variable selection algorithm, we turn to assessing its potential contribution to the current statistical literature along two dimensions in particular: its performance relative to other well-known variable selection algorithms and its generalizability. In particular, we focus on some challenging examples used in the empirical experimentation. We also compare our selection consistency results with those obtained by Leng et al. (2014) who study various variants of the least absolute shrinkage and selection operator (Lasso) invented by Tibshirani (1996). We also compare our methods to others, such as adaptive Lasso (aLasso) (Zou, 2006) and the Bayesian adaptive Lasso (BaLasso) (Leng et al., 2014), that have been developed to improve model selection in cases where there is a high correlation between variables (multicollinearity).

To this extent, we study the behavior of our method using simulated datasets. We start with simulated data characterized by full rank, i.e., where we have the number of observations n is greater than the number of regressors p ($p < n$). In basic regression, the way the regressors

[2] In general, EM is used to solve optimization problems with latent variables or problems with more unknown parameters than we wish to estimate. All those unknown components apart from the parameters of interest may be considered as missing data. This is an iterative method that oscillates between computing the expected value of the missing values in order to be comprehensive and finding the best parameters that optimize the objective function, conditional on these most recently computed expected values.

[3] As we show later, this method has a lot of value and is quite robust. However, it requires that we have enough data points in order to calculate the likelihoods. If we have a small number of variables, we can use the vanilla method or something else.

are weighted does not matter because the model parameters adjust accordingly. However, with the presence of regularization terms, the outcome of the inference itself is affected. With this simulated dataset, we first find that our likelihood-based variable selection algorithm consistently has the highest frequency of correct selections over 100 replications for a given data generating process, among all variable selection models and for different combinations of sample size n and variance σ^2 of the random error we add to the data.

We consider the following three additional examples: (i) a low dimension setting in which the correlation between the different variables changes depending on the number of correct variables specified, (ii) an extended version of the data generating process described above, where we consider higher dimensions, and (iii) an example in which we compare our model with a semiparametric method and a closely linked normal linear model. The results from these two extensions reinforce our results above that our method yields a higher frequency of choosing the correct variables than all other methods and a higher predictive power based on the mean squared error (MSE).

The superior performance of our algorithms is also confirmed by a comparison exercise done with machine learning models such as support vector regression, kernel ridge, elasticnet, gradient boosting regression, orthogonal matching pursuit and least angle regression. We use the popular python package "pycaret" to build the models, train them on half of the data, and evaluate them on the remaining portions. All models are created and tuned (optimized) using a 5-fold cross-validation. For each example, we generate n data points for training and n other data points for testing purposes with different standard deviation σ of the error terms.

After establishing the superior performance of this method over commonly used ones, we turn to applications to test its generalizability. We test the method on a real dataset. The dataset contains diabetes related information and consists of 9 variables on 403 subjects who were interviewed in a study to understand the prevalence of obesity, diabetes, and other cardiovascular risk factors for African Americans in Virginia (Schmidt et al., 1992). For this application, our model is able to pick relevant variables to derive the posterior distribution that yields good prediction performances.

The rest of this document is organized as follows. Chapter 1 introduces the basics of Markov Chain Monte Carlo (MCMC) methods. Chapter 2 discusses Bayesian linear models and Bayesian inference. Chapter 3 presents our Bayes variable selection method.

Chapter 1

Basics of Markov chain Monte Carlo methods

This Chapter 1 introduces the basics of Markov Chain Monte Carlo (MCMC) methods. In statistics, MCMC methods include a class of algorithms used for sampling from a probability distribution. By constructing a Markov Chain that has the desired distribution as an equilibrium distribution, one can obtain a sample of the desired distribution by recording the states of the chain. The higher the number of steps, the closer the sample distribution is to the actual desired distribution. There are various algorithms to build chains, including the Metropolis-Hastings algorithm (Chib and Greenberg, 1995) and the Gibbs sampler (Casella and George, 1992). To review the main concepts on this rich subject, we follow the books of Robert and Casella (2004) and Marin and Robert (2007) in this chapter. The paper of Andrieu et al. (2003) is also of main interest.

We explore maximum likelihood methods which attempt to solve optimization problems, and Bayesian methods which tackle integration problems. We decide to include both classes of techniques in this chapter because they share a lot of foundational building blocks.

For the rest of this chapter, we will take for granted the methodology by which we simulate a uniform density. From this assumption, many methods can be developed to simulate from complex and high dimensional distributions, be they for optimization purposes or to overcome integration issues.

The first layer of simulation tools involves the use of the inverse cumulative distribution function of a random variable (r.v.) x in order to simulate it. A more general version of this kind of function will be considered.

The generalized inverse of a cumulative distribution function F is given by:

$$F^{-1}(u) = \inf\{x : F(x) \geq u\}.$$

Lemma 1. *If $u \sim \mathcal{U}(0,1)$, then $F^{-1}(u)$ and $F(x)$ are two random variables with the same distribution.*

Based on this lemma, we only need to generate values $(u_1, ..., u_n)$ from a uniform distribution and derive the samples from the variable of interest $(F^{-1}(u_1), ..., F^{-1}(u_n))$. It is indeed a powerful trick. Unfortunately, it requires to analytically figure out the inverse of a function which is oftentimes too complex for that.

The next methodology that can be used to generate r.v. is from intermediary variables that can easily be sampled. For instance, one can use exponential distribution to generate beta r.v. using the following fact:

$$\text{if } x_i \overset{iid}{\sim} \mathcal{E}xp(1),$$

$$\textit{then}$$

$$y = \frac{\sum_{i=1}^{a} x_i}{\sum_{i=1}^{a+b} x_i} \sim \mathcal{B}(a,b), \quad a, b \in \mathbb{N}^*.$$

If the two methods mentioned above are not applicable to a given problem, then we need more sophisticated techniques to simulate data appropriately.

1.1 Monte Carlo based methods

As shown above, many distributions can be generated by using the inverse transform methods and other general transformations with the uniform distribution. However, there exists a variety of distributions that are difficult, computationally expensive or even impossible to generate using these mentioned procedures. For the latter, many families of algorithms have been developed as alternative ways to get their distribution. Here, we introduce a key result in the world of simulations.

Theorem 1.1.1 (Fundamental theorem of simulation)**.** *(Robert and Casella, 2004)*
The sampling from $x \sim f$ is equivalent to one from:

$$(x,u) \sim \mathcal{U}\{(x,u) : 0 < u < f(x)\}.$$

This theorem is what makes simulation work. The order of sampling makes the difference. Instead of sampling from f and then sample from $\mathcal{U}[0, f(x)]$, the couple (x,u) is jointly sampled (x coming from a pseudo-distribution) and accepted if the condition $0 < u < f(x)$ is satisfied.

For almost all of the following methods, we only need to know the functional form of the density function $f(x)$ we are trying to generate, up to a multiplicative constant (meaning the normalizing constant that turns f into a probability density function is not relevant here).

1.1.1 Accept-Reject algorithm

Here we are concerned with finding a proxy distribution that shares functional characteristics with f but is much simpler (or at least simpler to generate data from). Such distribution whose density is called the candidate density $g(x)$ (or instrumental) is used to generate potential values that are filtered in order to obtain the target distribution. We lay out the principle below:

Theorem 1.1.2. *(Robert and Casella, 2004)*
Let $x \sim f$ and $y \sim g$ such that $f(x) < Mg(x)$, $\forall x$ on the support of f, for some constant $M \geq 1$. For x to be distributed with density f, it is sufficient that:

$$y \sim g \qquad \text{and} \qquad u|y \sim \mathcal{U}[0, Mg(y)].$$

Under the requirements that:

- *$g(x) > 0$ when $f(x) > 0$,*
- *there exists a constant M such that $f(x)/g(x) \leq M$ for all x.*

The idea is to get u, and verify if $f(x) \leq Mg(x)$ holds. Because the inequality implies that $\int f(x)\mathrm{d}x \leq \int Mg(x)\mathrm{d}x$, it is necessary that $M \geq 1$.

Algorithm 1 Accept-Reject algorithm.

```
tracker = N
while tracker > 0 do
    Generate y ~ g and u ~ U_[0,1]
    if u ≤ f(y)/(Mg(y)) then
        x[tracker] = y
        tracker = tracker − 1
    end if
end while
```

It can be shown that the conditional cumulative probability of a value generated from g given a uniformly distributed value u is less that the f/Mg is equal to the cdf of x (see Figure 1.1). A simple calculation of a conditional probability allows us to prove this. We can easily see that:

$$\begin{aligned}\mathbb{P}\left(y \leq x \middle| u \leq \frac{f(y)}{Mg(y)}\right) &= \frac{\mathbb{P}\left(y \leq x, u \leq \frac{f(y)}{Mg(y)}\right)}{\mathbb{P}\left(u \leq \frac{f(y)}{Mg(y)}\right)} = \frac{\int_{-\infty}^{x}\int_{0}^{\frac{f(y)}{Mg(y)}} \mathrm{d}u g(y)\mathrm{d}y}{\int_{-\infty}^{\infty}\int_{0}^{\frac{f(y)}{Mg(y)}} \mathrm{d}u g(y)\mathrm{d}y} \\ &= \frac{\int_{-\infty}^{x}\frac{f(y)}{Mg(y)}g(y)\mathrm{d}y}{\int_{-\infty}^{\infty}\frac{f(y)}{Mg(y)}g(y)\mathrm{d}y} = \frac{\int_{-\infty}^{x} f(y)\mathrm{d}y}{\int_{-\infty}^{\infty} f(y)\mathrm{d}y} = \mathbb{F}(x),\end{aligned}$$

where $\mathbb{F}$ is the cumulative distribution of x.

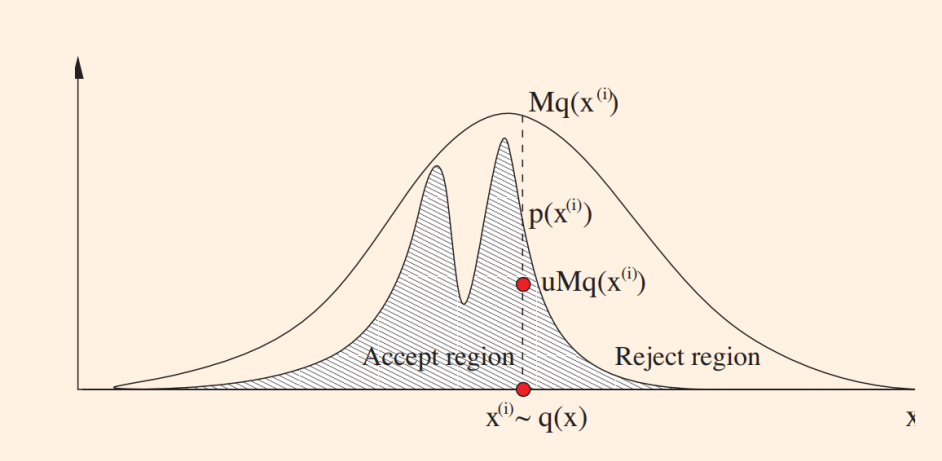


Figure 1.1: Distribution generated by Accept-Reject method (here $q = g$). The y axis represents the density measure (Murphy, 2012).

Remark 1. *Some of the properties of the Accept-Reject algorithm are as follows:*

- *The acceptance-rejection simulation is an exact method since no approximation is done.*
- *We need to know the value of f/M.*
- *From $f \leq Mg$, M can be any upper bound, but the closest it is to the least upper bound, the better.*
- *The acceptance probability is $1/M$.*

One of the weaknesses of this simple Accept-Reject method is that it could discard a significant portion of generated values and therefore waste computational power. So suppose that the function f is too computationally complex to evaluate. We can modify the Accept-Reject algorithm to get a faster one by adding an intermediary step. This slight variant of the algorithm is known as the Envelope Accept-Reject method (see Algorithm 2 for details).

Lemma 2. *(Barbieri and Berger, 2004)*
If there exists h a function such that $h(x) \leq f(x) \leq Mg(x)$, we can speed the algorithm 1 up and still produce samples from f.

The probability of not evaluating f is equal to $\frac{1}{M}\int h(x)dx$. So the closer h is to f, the better.

Algorithm 2 Envelope Accept-Reject algorithm.

$tracker = N$
while $tracker > 0$ **do**
  Generate $y \sim g$ and $u \sim \mathcal{U}_{[0,1]}$
  if $u \leq \frac{h(y)}{Mg(y)}$ **then**
    $x[tracker] = y$
    $tracker = tracker - 1$
  elsewhere if $u \leq \frac{f(y)}{Mg(y)}$ **then**
    $x[tracker] = y$
    $tracker = tracker - 1$
  elsewhere
    Reject y
  end if
end while

1.1.2 Optimization and integration methods

In this section, we explore simulations that are tailored to a specific purpose. Sometimes, we use the power of simulations combined with a couple of heuristics in order to find some global minimum or maximum on some function densities or likelihoods where we are missing some information such as latent variables. In Bayesian applications, we will most likely be interested in computing some statistics about a random variable. However, the empirical Bayes method consists of optimizing some hyperparameters that are to be used to drive inference about other parameters. So we will present a method for each one of these use cases.

Importance Sampling

The idea of Monte Carlo integration is simple and is all about approximating the following:

$$\mathbb{E}_p[h(x)] = \int h(x)p(x)\mathrm{d}x \tag{1.1}$$

$$\approx \frac{1}{n}\sum_{i=1}^{n} h(x_i) \tag{1.2}$$

from data $(x_1, \ldots, x_n)$ generated according to p. However, the latter density might be difficult to sample from directly. To sidestep this issue, we can sample from an intermediary distribution q that is chosen to conform with some properties of p, but also the quantity of interest h itself. We reformulate the problem as follows:

$$\mathbb{E}_p[h(x)] = \int h(x)p(x)\mathrm{d}x \tag{1.3}$$

$$= \int h(x)\frac{p(x)}{q(x)}q(x)\mathrm{d}x = \mathbb{E}_q\Big[h(x)\frac{p(x)}{q(x)}\Big] \tag{1.4}$$

$$\approx \frac{1}{n}\sum_{i=1}^{n} w_i h(x_i), \text{ where } w_i = \frac{p(x_i)}{q(x_i)} \tag{1.5}$$

assuming the same number of samples generated according to q.

This computation will converge almost surely if the variance of $h(x)$ under the new distribution is finite. This means that we additionally need to make sure that:

$$\mathbb{E}_q\Big[h^2(x)\frac{p^2(x)}{q^2(x)}\Big] = \int h^2(x)\frac{p^2(x)}{q(x)}\mathrm{d}x < \infty.$$

This is why it is important to make sure that q has thicker tails than p so that p/q does not blow up. It is also assumed that integrating h with respect to p will also produce a finite mean and finite variance.

An alternative, more stable estimator, which could reduce or eliminate the severity of the variance issue is the sampling importance resampling. Using the samples from the regular importance sampling, we resample according to:

$$\hat{p}(x) = \sum_{i=1}^{n} w_i^* \delta_{x_i}(x),$$
$$\text{such that } w_i^* = \frac{w_i}{\sum_{i=1}^n w_i},$$

where δ_{x_i} is the dirac measure at x_i. Then we can use these samples in the original definition. We can show that $\hat{p} \to p$ as $n \to \infty$ as follows:

$$\begin{aligned}\hat{F}(x_0) &= \sum_{i=1}^{n} w_i^* \mathbb{1}(x_i \le x_0) = \frac{\sum_{i=1}^n \mathbb{1}(x_i \le x_0)\frac{p(x_i)}{q(x_i)}}{\sum_{i=1}^n \frac{p(x_i)}{q(x_i)}} \\ &\to \frac{\int \mathbb{1}(x \le x_0)\frac{p(x)}{q(x)}q(x)\mathrm{d}x}{\int \frac{p(x)}{q(x)}q(x)\mathrm{d}x} = \int \mathbb{1}(x \le x_0)p(x)\mathrm{d}x \\ &= F(x_0),\end{aligned}$$

where $\mathbb{1}$ is the indicator function. When using this technique, one needs to make sure that $\frac{1}{n}\sum_i \frac{p(x_i)}{q(x_i)} \to 1$ (numerically) to get reliable results.

Expectation Maximization algorithm

The Expectation Maximization algorithm is an almost indispensable method specially for problems involving some unobserved mixture components (Murphy, 2012). In general, it is used to solve optimization problems with latent variables or problems with more unknown parameters than we wish to estimate. All those unknown components apart from the parameters of interest may be considered as missing data. This is an iterative method that oscillates between computing the expected value of the missing values in order to be comprehensive, and finding the best parameters that optimizes the objective function conditional on these most recently computed expected values. For the following demonstration however, we will

only consider cases where we have latent variables z to interact with the data x only though the parameters θ as expressed in the graphical model structure:

$$z \to \theta \to x.$$

In mathematical terms, the goal is to find the set of parameters θ that maximize :

$$p(x|\theta) := f(x,\theta) = \int p(x,z|\theta)\mathrm{d}z = \int p(x|z,\theta)p(z|\theta)\mathrm{d}z,$$

where x represents the observed data and z represents all latent variables. For this integral to be evaluated, we need to have known θ which we don't have at this time and are actually trying to estimate. So the strategy is to assume that we have $\hat{\theta}_{(t)}$ at this time step, and use it to estimate the function $p(x,z|\theta)$ which we maximize with respect to θ to get our refined $\hat{\theta}_{(t+1)}$. We summarize these steps as follows:

- Expectation: we define
$$Q(\theta|\hat{\theta}_{(t)}) = \mathbb{E}_{z|\hat{\theta}_{(t)}}[\log\ p(x,z|\theta)].$$
- Maximization: we compute
$$\hat{\theta}_{(t+1)} = \arg\max_{\theta} Q(\theta|\hat{\theta}_{(t)}).$$

It is clear that, as we generate the sequence $\{\hat{\theta}_{(t)}\}$, we are actually improving Q. We will show that the maximization of $f(x,\theta)$ is also getting better. First, let's introduce the Gibbs inequality upon which we will rely.

Definition 1 (Gibb's inequality)**.** *(Mackay, 2005) Let P and Q be two probability defined over the same set Ω. The relative entropy (also known as the Kullback–Leibler divergence) is defined as:*

$$D_{KL}(P||Q) = \sum_{w} p(w)\log\frac{p(w)}{q(w)}.$$

Gibb's inequality states that:

$$D_{KL}(P||Q) \geq 0,$$

with equality only if $P = Q$.

Let's show that improving Q corresponds to improving $f(x,\theta)$:

$$\begin{aligned}
Q(\theta|\hat{\theta}_{(t)}) &= \mathbb{E}_{z|\hat{\theta}_{(t)}}[\log\ p(x,z|\theta)] \\
&= \mathbb{E}_{z|\hat{\theta}_{(t)}}[\log\ p(z|x,\theta)p(x|\theta)] \\
&= \mathbb{E}_{z|\hat{\theta}_{(t)}}[\log\ p(z|x,\theta)] + \mathbb{E}_{z|\hat{\theta}_{(t)}}[\log\ p(x|\theta)] \\
&= \mathbb{E}_{z|\hat{\theta}_{(t)}}[\log\ p(z|\theta)] + \log\ f(x,\theta).
\end{aligned}$$

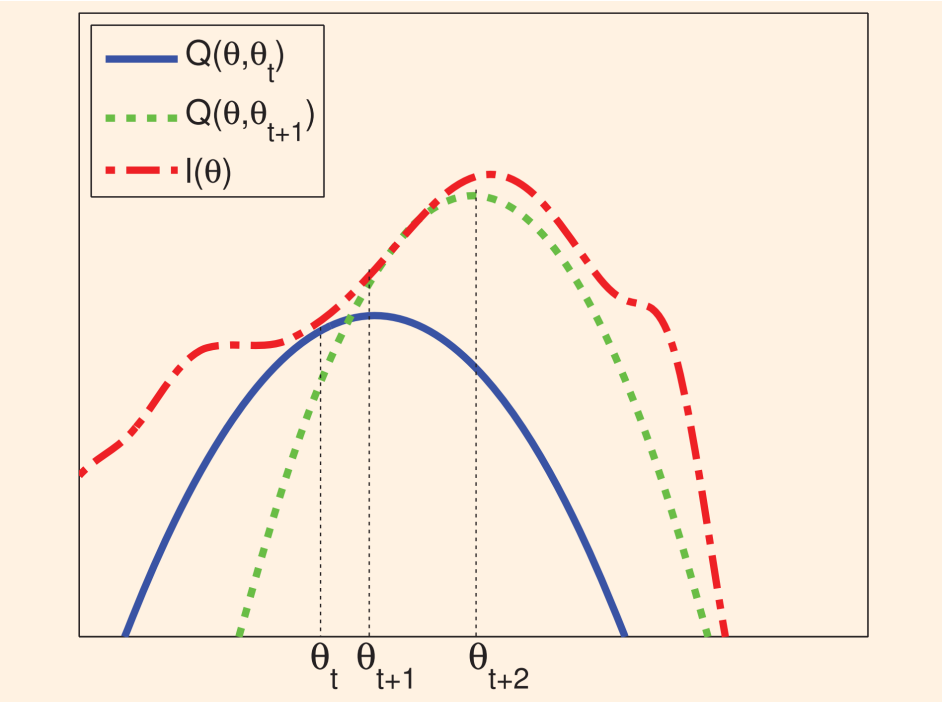


Figure 1.2: Illustration of EM iterations Bishop (2006).

Note that we have that $p(z|x,\theta) = p(z|\theta)$ because x and z are conditionally independent given θ . If we would like to consider z as incorporating missing data (where independence is broken), we will just have to refine the definition $Q(\theta|\hat{\theta}_{(t)}) = \mathbb{E}_{z|x,\hat{\theta}_{(t)}}[\log\ p(x,z|\theta)]$. The steps of the proof remain the same. Let's resume the proof:

$$Q(\theta|\hat{\theta}_{(t)}) - \log\ f(x,\theta) = \mathbb{E}_{z|\hat{\theta}_{(t)}}[\log\ p(z|\theta)].$$

Now we apply the difference operator to both sides:

$$\begin{aligned}\Delta &= Q(\hat{\theta}_{(t+1)}|\hat{\theta}_{(t)}) - Q(\hat{\theta}_{(t)}|\hat{\theta}_{(t)}) - [\log\ f(x,\hat{\theta}_{(t+1)}) - \log\ f(x,\hat{\theta}_{(t)})] \\ &= \mathbb{E}_{z|\hat{\theta}_{(t)}}[\log\ p(z|\hat{\theta}_{(t+1)})] - \mathbb{E}_{z|\hat{\theta}_{(t)}}[\log\ p(z|\hat{\theta}_{(t)})].\end{aligned}$$

According to Gibb's inequality, $\Delta \leq 0$, which implies that:

$$Q(\hat{\theta}_{(t+1)}|\hat{\theta}_{(t)}) - Q(\hat{\theta}_{(t)}|\hat{\theta}_{(t)}) \leq \log\ f(x,\hat{\theta}_{(t+1)}) - \log\ f(x,\hat{\theta}_{(t)}).$$

So it means that $f(x,\theta)$ is increasing at least as much as $Q(\theta|\hat{\theta}_{(t)})$. See Figure 1.2 for more details on Expectation-Maximization principle.

1.2 Introduction to Markov chains

This is the second component behind MCMC methods and it is arguably as important as the first one (Monte Carlo principle). Its convergence properties makes it so that we may know

even much less about the characteristics of f in order to generate values from it. Furthermore, its requirements are pretty much naturally satisfied for most practically encountered problems, which means there is little to no analytical work to be done in order to use it. But this convenience does come with major drawbacks during the verification of chain convergences. Let's begin by presenting some important and simple concepts in Markov theory.

1.2.1 Preliminaries and basic concepts

A discrete time stochastic process is a sequence of evolving points $\{\theta_t\}_{t\geq 0}$ taking their values from a set Ω such that when $\theta_t = i$, the process is in state i at time t for $i \in \Omega$. In general, θ_t may be seen as a stochastic function mapping values from $\mathbb{N} \to \Omega$.

Definition 2. *Let $\{\theta_t\}_{t\geq 0}$ be a discrete time stochastic process with values in the state space Ω. For all $t \geq 0$, if*

$$\mathbb{P}(\theta_{t+1} = j|\theta_t = i, \theta_{t-1} = w_{t-1}, ..., \theta_0 = w_0) = \mathbb{P}(\theta_{t+1} = j|\theta_t = i),$$

then the resulting process is called a Markov chain.

In addition, if $\mathbb{P}(\theta_{t+1} = j|\theta_t = i)$ is independent of t, then it is known as a homogeneous Markov chain (we will focus on this type of Markov chain). We define the transition kernel in general (for both discrete and continuous cases) as follows:

Definition 3. *A transition kernel K is a function on $\Omega \times \mathcal{B}(\Omega)$ with the following properties:*

- $\forall x \in \Omega, K(x, \cdot)$ *is a probability measure,*
- $\forall\ y \in \mathcal{B}(\Omega),\ K(\cdot, y)$ *is measurable.*

When x is discrete, K is equivalent to a matrix like P with elements p_{ij}:

$$p_{ij} = \mathbb{P}(\theta_{t+1} = j|\theta_t = i),\ i, j \in \Omega$$
$$p_{ij} \geq 0 \text{ and } \sum_{l\in\Omega} p_{il} = 1.$$

These properties above make P a stochastic matrix.

Let's define the distribution of θ_t by μ_{θ_t}, which is characterized by the recurrence relation between successive states as follows:

$$\mu_{\theta_t}^T = \mu_{\theta_{t-1}}^T P = \mu_{\theta_0}^T P^t,$$

where P^t is called the t-step transition matrix. We can go further and define the following joint distribution from the Bayes' rule:

$$\begin{aligned}\mathbb{P}(\theta_0 = w_0, \theta_1 = w_1, \ldots, \theta_t = w_t) &= \mathbb{P}(\theta_0 = w_0)\mathbb{P}(\theta_1 = w_1|\theta_0 = w_0) \\ &\quad \times \cdots \times \mathbb{P}(\theta_t = w_t|\theta_{t-1} = w_{t-1}, \ldots, \theta_0 = w_0) \\ &= \mu_{\theta_0}(w_0) \prod_{k=1}^{t} p_{w_{k-1}w_k},\end{aligned}$$

where $w_k \in \Omega$ for $k \in \mathbb{N}$. Using this joint distribution allows us to compute all sorts of information about the process, at least theoretically. So we can for instance compute the probability of ending up in state j at time t_2 given that we were in state i at time t_1. Let $\Delta t = t_2 - t_1$, then:

$$\begin{aligned}p_{ij}(\Delta t) &= \mathbb{P}(\theta_{t+\Delta t} = j|\theta_t = i) \\ &= \sum_{k_1, k_2, \ldots, k_{\Delta t - 1} \in \Omega} p_{ik_1} p_{k_1 k_2} \cdots p_{k_{\Delta t - 1} j} \\ &= \sum_{k_1, k_2, \ldots, k_{\Delta t - 1} \in \Omega} \mathbb{P}(\theta_{t+1} = w_{k_1}, \ldots, \theta_{t+\Delta t - 1} = w_{k_{\Delta t - 1}}, \theta_{t+\Delta t} = j|\theta_t = i).\end{aligned}$$

This is a special case of the more general continuous case known as the Chapman-Kolmogorov equations (Brémaud, 2020).

Definition 4. *State j is said to be accessible from state i if there exists R such that $p_{ij}(R) > 0$. If both states are accessible from each other, they are said to communicate. This is denoted by $i \leftrightarrow j$.*

As a consequence, we have:

$$\begin{gathered}i \leftrightarrow i, \text{ since } p_{ii}(0) = 1 \\ i \leftrightarrow j \Rightarrow j \leftrightarrow i, \text{ since } p_{ij}(R) = p_{ji}(R), \text{for all } R \in \mathbb{N} \\ i \leftrightarrow j, j \leftrightarrow l \Rightarrow i \leftrightarrow l.\end{gathered}$$

So this induced communication is an equivalence relation and it partitions the state space Ω into disjoint communication classes.

Theorem 1. *If there exists only one communication class, then the transition matrix P is irreducible.*

Irreducibility basically means that any state is eventually reachable from any other state with non-zero probability. If two states i and j communicate, then they have the same period, which we define next.

Definition 5. *The period of a state i is defined as the greatest common divisor (gcd):*

$$d_i = gcd\{\psi(i)\},$$
$$d_i = \infty \quad \text{if } \psi(i) = \emptyset, \text{ with } \psi(i) = \{n \geq 1 : p_{ii}(n) > 0\}.$$

If $d_i = 1$, then the state i is aperiodic.

Periodicity is important in the context of defining the concept of recurrence, which is a key element in most chains generated through MCMC algorithms. Before getting to it, let's first define hitting time as:

$$T_i = \inf\{t > 0 : \theta_t = i\}.$$

Definition 6. *The state i is recurrent if:*

$$\mathbb{P}_i(T_i < \infty) = 1.$$

This represents the probability to return to state i from itself. If it is not recurrent, then it is transient. In addition, if:

$$\mathbb{E}(T_i) < \infty,$$

then the state is positive recurrent. Otherwise, it is null recurrent.

In fact, recurrence implies the return to every state or in general, small set with non zero measure, an infinite amount of times. However in MCMC, a stronger recurrence known as Harris recurrence is needed. It implies that every possible path (a particular chain of successive states) of the Markov chain is expected to be traversed an unbounded number of times.

We have defined several concepts because the set of Markov chains is overwhelmingly extensive and we are interested in a particular set called ergodic chains which still has a significant expressive power in that most real life scenarios exhibit chains that fit into this category. For a state to be ergodic, it has to be aperiodic and positive recurrent. We conclude with this theorem:

Theorem 2. *(Brémaud, 2020)*
A chain is ergodic if and only if all its states are aperiodic and it has one recurrent class.

1.2.2 Illustration

Let us consider the process with a transition kernel defined by:

$$\tilde{p}_{i0} = 1,$$
$$\tilde{p}_{ij} = \frac{1}{1 + i^2 j^2},$$

where $i, j \in \Omega = \{0, 1, 2, 3\}$, and $\tilde{p}_{ij}$ corresponds to the unnormalized probability p_{ij}. Let start by plotting μ_{θ_t} over t and compare it with its approximated version from simulations (see Figures 1.3 and 1.4). To be more precise, the left figure shows the probability of being in

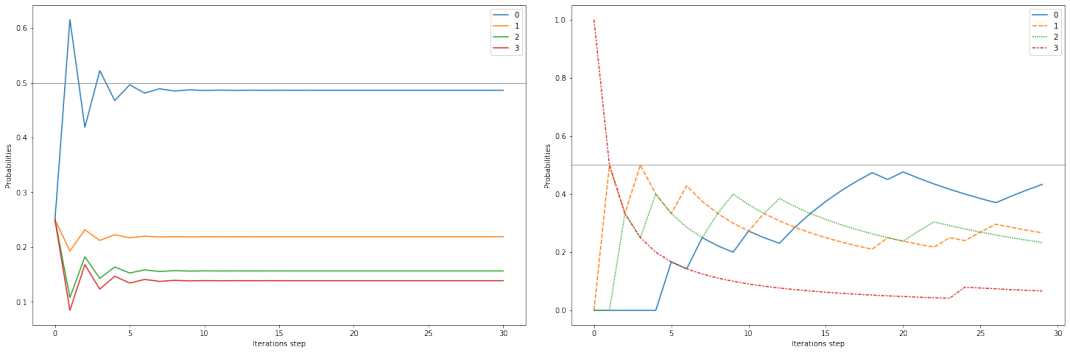


Figure 1.3: The cumulative probability of the chain being in a certain state. The left plot shows exact values computed from the true distribution μ_{θ_t}. The right plot shows the approximation with simulated states.

a given state calculated by using the actual transition kernel . As for the right figure, these probabilities are computed cumulatively from a simulation table where each row contains a 1 and three 0 as to indicate the current state of the chain. The next state is obtaining by simulating the next index given the current state. As we defined in Definition 2, the transition kernel contains only conditional probabilities. What the chain converges to are the probabilities of being a a state regardless of the previous state. We define it as follows:

$$\pi = \lim_{t \to \infty} \mu_{\cdot}^{T} P^{t},$$

where $\mu_{\cdot}$ represents the uniform distribution (equal probabilities for states) for the initial state of the process.

We see that the simulated version is very noisy in the initial steps because it tends to be under strong influence of the arbitrary starting distribution (uniform in this case).

When we run it for a longer time, we observe that it tends to become more stable asymptotically. With a single chain, it can be tough to capture how uncertainty evolves and affect the process.

Let us run multiple chains and see how the stability of the convergence behaves. For each iteration step, we will display the expected mean probability and a 95% confidence interval (see Figure 1.5).

Now we have a much better view around the variability of the distributions that we end up with. Observe the evolving overlap that occurs between the red and green states and notice how it tends to die off during the second phase of the iterations. We will talk about the implication of this behavior when we introduce our variable selection methodology.

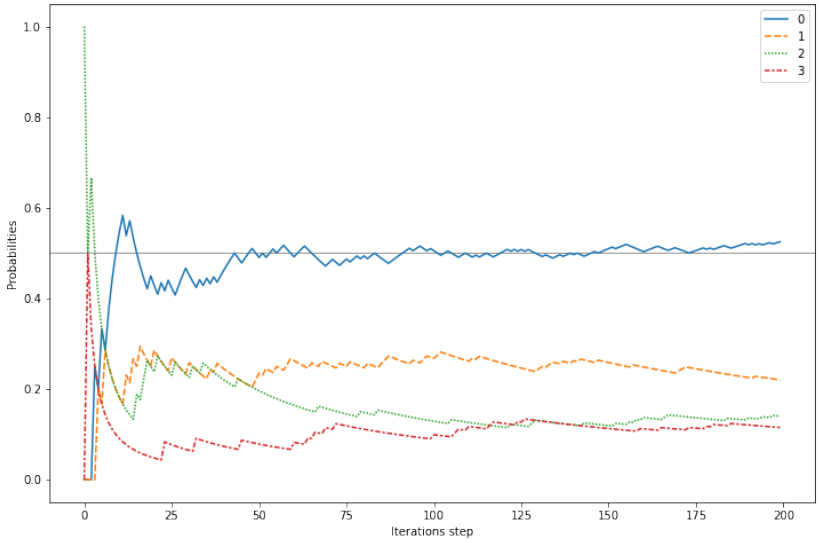


Figure 1.4: Approximation of cumulative probability of the chain being in a certain states over 200 iterations.

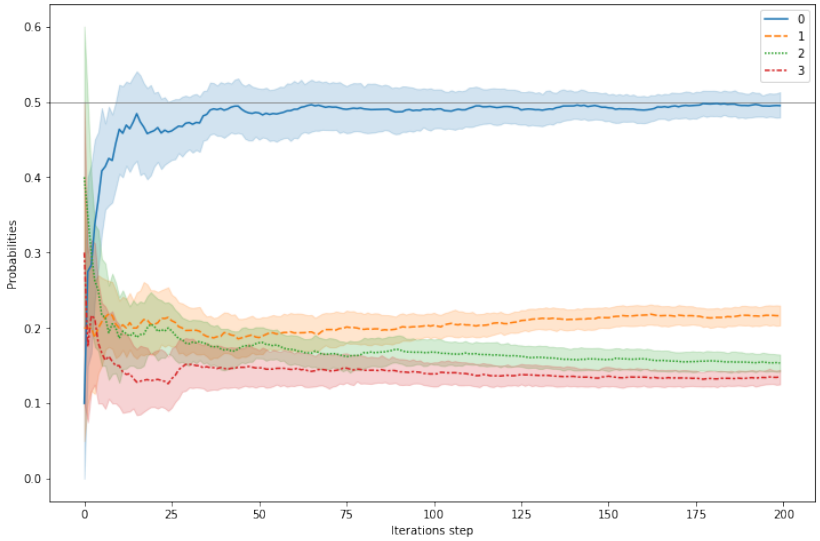


Figure 1.5: Distribution of cumulative states probabilities and their 95% confidence intervals computed from 20 different runs, each of 200 iterations.

1.3 MCMC methods

Markov chain Monte Carlo (MCMC) algorithms aim to provide approximations based on the following equality:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} h(\theta_t) = \mathbb{E}_f[h(\theta)] = \int h(\theta) f(\theta) d\theta,$$

by building an ergodic and stationary Markov chain from a transition kernel P that converges to a distribution f (equivalent to π in the discrete case). In other terms, we need the following

condition to hold:

$$\|\mu P^t - f\| \leq c\lambda^t, \quad \lambda < 1, c > 0$$

for any starting distribution μ. Since practically speaking, the kernel P is almost never available explicitly, MCMC methods differ from other simulations methods in that they rely on a very flexible irreducible transition kernel Q from which the next state j is proposed given the current state is i. Assuming $i \neq j$, the new state j is accepted with probability α_{ij}. From these two phases, the resulting transition kernel P is such that:

$$p_{ij} = q_{ij}\alpha_{ij}.$$

Note that Q is known as the candidate generating distribution (or instrumental distribution). Breaking down the generation process in these two steps is what makes MCMCs more powerful than techniques such as importance sampling which relies on the careful design of an appropriate importance function, a task that is very challenging specially in high dimensional settings. We will focus on a specific type of chain called reversible chains. Those are chains that satisfy the equality:

$$\pi_i p_{ij} = \pi_j p_{ji}.$$

When this condition holds, we easily derive an upper bound for the acceptance probabilities as follows:

$$\begin{aligned}
\pi_i p_{ij} &= \pi_j p_{ji} \Leftrightarrow \\
\pi_i q_{ij} \alpha_{ij} &= \pi_j q_{ji} \alpha_{ji} \Leftrightarrow \\
\pi_i q_{ij} \alpha_{ij} &\leq \pi_j q_{ji} \Leftrightarrow \\
\alpha_{ij} &\leq \frac{\pi_j q_{ji}}{\pi_i q_{ij}}.
\end{aligned}$$

To maximize the acceptance probability, we set it to this upper bound above, or one, since it also has to satisfy the condition $\alpha_{ij} \leq 1$. So we end up with:

$$\alpha_{ij} = \min\left(1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}}\right).$$

This is what drives the algorithms that are described next.

1.3.1 Metropolis–Hastings algorithm

The Metropolis–Hastings algorithm (MHA) is a type of Markov chain Monte Carlo scheme used for generating values of a given random variable (Metropolis et al., 1953). It is used to solve problems that involve manipulating multidimensional distributions (specially when standard conjugate prior distributions are not available or cannot be used, as in most generalized linear models), and dealing with density functions upon which very little information is known. However, there is a trade-off between this latter advantage and the fact that the

generated variables are correlated due to the presence of conditionality in the Markov chain process.

MHA uses the Markov process because of the guarantee of finding a unique stationary distribution given certain initial conditions. The validity of the algorithm relies in choosing an appropriate function f that is proportional to the limiting distribution up to a multiplicative constant, and it is particularly useful in that it avoids the complicated (sometimes computationally infeasible) calculation of the normalizing constant that would turn f into an actual probability density corresponding to the target distribution. To achieve that, an arbitrary distribution q is chosen such that $f(y)/q(y|x)$ is known up to a constant that is independent of x. This means that if we ignore a term in $q(.|x)$ (like a normalizing constant) that depends on x, it will negatively affect the acceptation step unless q is symmetric. Additionally, the support of q must include that of f, namely it must be large enough to cover the whole space of interest.

Now, let us consider the case where f is readily available from the distribution of interest. In practice, it is easy to compute $p(y|\theta)$ (the likelihood of the observed data y given the parameters) and $p(\theta)$. However, deriving the explicit form of

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)d\theta} \propto p(y|\theta)p(\theta) \tag{1.6}$$

is often challenging because of the integral computation. So the trick is to take advantage of the ratio $\frac{p(\theta_1|y)}{p(\theta_2|y)}$, which is free from the normalizing constant $p(y)$. In this type of setting, a natural choice for f is $p(y|\theta)p(\theta)$, and q can be a very simple distribution such as a normal or a uniform, provided that its dispersion is well calibrated for a smooth exploration. Nevertheless, the full Metropolis–Hastings algorithm is not limited to such restrictions, although the more we know about the target distribution, the better our choice of pseudo-distributions will be.

Algorithm 3 Metropolis–Hastings algorithm.

Initialization: choose an arbitrary point $x^{(0)}$
for $t = 0, 1, 2, \ldots, N$ **do**

Generate $y_t \sim q(y|x^{(t)})$ and $u \sim \mathcal{U}_{[0,1]}$
Compute the acceptance-rejection probability: $\rho(x^{(t)}, y_t) = \min\left(\frac{f(y_t)}{f(x^{(t)})}\frac{q(x^{(t)}|y_t)}{q(y_t|x^{(t)})}, 1\right)$
Set

$$x^{(t+1)} = \begin{cases} y_t, & \text{if } u < \rho(x^{(t)}, y_t); \\ x^{(t)}, & \text{otherwise.} \end{cases}$$

end for

To understand the intuition behind the algorithm 3, let us consider the case where q is symmetric such that $q(x|y) = q(y|x)$, which is the original version of this method (Metropolis algorithm, Metropolis and Ulam (1949)). So in this case, the acceptance-rejection probability

reduces simply to $\rho(x, y) = \min(f(y)/f(x), 1)$. If $f(y)/f(x) > 1$, we certainly choose the proposed value y generated from q. But if $\rho < 1$, we sometimes move to regions of lower probability depending on u, or rather on the ratio ρ which represents the relative frequency of the newly proposed value compared to the previous one. This allows for a comprehensive exploration of the whole sample space while keeping the priority on highly probable sets. The key added ingredient that completes the MHA is the breakdown of the transition into two steps: the conditional proposal probability and the acceptance probability.

We can better understand MHA by comparing its properties to that of the Accept-Reject method (A-R). The latter generates a sample of variables that are identically and independently distributed (iid). It rejects many values that could have been useful, hence MHA is more efficient in that at each step a value is kept. Furthermore, A-R methods require the setting of an appropriate value for the hyper-parameter M which can be difficult to find or time consuming. However, if all the conditions required by the A-R method can be satisfied without so much difficulty, it is better to adopt it because the dependence of values by MHA makes it difficult to visit the whole sample space specially when the chosen proposal distribution is not appropriate enough (such as not satisfying the support requirement).

1.3.2 Gibbs sampler

Another algorithm of interest is the Gibbs sampler (GS) and is named after the physicist Josiah Willard Gibbs (1839-1903). It was described by brothers Stuart and Donald Geman in Geman and Geman (1984). It is particularly useful in Bayesian computations. Suppose that we want to generate two jointly distributed variables. One way to do so is by evaluating the joint posterior probabilities on a 2×2 grid and then normalizing by their sum. We can then obtain the marginal distribution of a single parameter by summing over the other dimensions. Although this is a powerful method, it suffers from the curse of dimensionality. GS is an elegant way to perform a similar sampling with robust statistical information without the need to generate an exponentially growing number of values. It uses the full conditional distributions of the parameters to generate their joint distribution.

Before introducing the algorithm, we define some variables and notations.

$$\text{Let } \boldsymbol{x} = \{x_1, x_2, \ldots, x_p\} \quad \text{and} \quad \boldsymbol{x_{i:j}} = \{x_i, \ldots, x_j\}.$$

For convenience, it is needed that we can simulate from the corresponding full conditional densities $\pi_1, \ldots, \pi_p$, that is, we can simulate

$$x_i | x_1 \ldots, x_{i-1}, x_{i+1}, \ldots, x_p \sim \pi_i(x_i | x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_p), \tag{1.7}$$

for $i = 1, 2, \ldots, p$. A general scheme on a large-dimensional target (for parameters with possibly high dimension p) Gibbs sampler is given by Algorithm 4.

Algorithm 4 Gibbs sampler.

Initialization: choose an arbitrary point $\boldsymbol{x^{(0)}}$
for $t = 1, 2, \ldots, N$ **do**
  for $k = 1, 2, \ldots, p$ **do**
    sample $x_k^{(t)} \sim \pi_k\Big(x_k \;\Big|\; \boldsymbol{x^{(t)}_{1:\max(k-1,1)}}, \boldsymbol{x^{(t-1)}_{\min(k+1,p):p}}\Big)$
  end for
end for
Output: $\boldsymbol{\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}}$

The GS procedure is a Markov process in that $x^{(t)}$ is conditionally independent of the previous vectors given $x^{(t-1)}$. In fact, GS is a special case of MHA that accepts the proposed candidate at every iteration and uses full conditional distributions as an instrumental (proposal) distributions.

In many complex statistical problems, we have to combine the two previous MCMC schemes (Metropolis-Hastings and Gibbs sampler) in order to simulate from the joint distribution of all unknown parameters. This type of scheme is known by Metropolis-Hastings within Gibbs sampler (MHGS) and generalizes GS and MA by allowing for the use of instrumental distributions free from the requirement of symmetry or being full conditional distributions. They should however still retain the Markov property of conditional independence. A mix of Gibbs and MHA is particularly useful when not all full conditional distributions are available.

When we consider high dimensional problems, the generalized MHA algorithm differs from the one introduced above in that it generates a vector of parameters sequentially and takes advantage of the newly added information as opposed to sampling the full vector independently in parallel (for instance sampling a full vector of Gaussian variables at once or one at a time). This added feature can significantly speed up convergence (Casella and George, 1992).

1.4 MCMC diagnostics and comparisons

Establishing theoretical convergence results is simply not enough to determine whether a generated chain has converged. Many heuristics have been developed by researchers in order to assess the quality of chains, but these methods are not only about pure deduction, but they tend to rely on a lot of intuitive thinking. In fact, they are useful in detecting chains that have not converged. If the underlying has indeed converged, it is almost impossible to ascertain it.

1.4.1 Before checking for convergences

MCMC methods start from some random initial point which is most likely not sampled from f. Even though some techniques such as using the estimates of an optimizer to choose the initial point may help, it still does not guarantee that the chain will start from a point where a good exploration of the sample space can be done. A way to reduce this problem is to remove the first portion of generated points known as the burn-in phase. The main challenge remains figuring out where to end this burn-in phase.

The amount of time it takes to forget about the initial point is called the mixing time. The mixing time from state x_0 is defined as :

$$\tau_\epsilon(x_0) = \min(t : \|\mu_{x_0} P^t - f\|_1 \leq \epsilon),$$

for all $\epsilon > 0$ and μ_{x_0} is the distribution with all mass on x_0. In general, it could also be just a uniform distribution. The mixing time of the chain is defined as:

$$\tau_\epsilon = \max_{x_0} \tau_\epsilon(x_0).$$

We know that the rate of convergence depends on the second largest eigenvalue of the transition matrix. Defining the eigengap as $\delta = 1 - \lambda_2$, it can be shown that (Murphy, 2012):

$$\tau_\epsilon \leq O\Big[\frac{1}{\delta} \log\big(\frac{k}{\epsilon}\big)\Big],$$

where k is the number of states. Computing the eigengap remains challenging for transition matrices defined only implicitly or that are high in dimension.

In practice, due to the Markov chain property and the random initialization, if we need to generate N values, we first find a sequence of $\hat{\tau}_\epsilon$ values at which point we believe the process to have reached stationarity, then we generate the N values that will represent our empirical distribution. The first step can be seen as a way to remove any influence the starting might have on the chain, so as to make the assumption that the first value in our N generated numbers is coming from the target distribution more plausible.

1.4.2 Converging to the target distribution

Notwithstanding all the guarantees of the theoretical developments about stationarity, the issue is that they are still asymptotic in nature, whereas algorithms have to end at some point. So the main challenge remains making sure, or at least checking, that the algorithms are exploring the space fast enough to provide good approximations of the target distribution. The most straightforward way is to plot the sampled points over iterations and inspect those graphs. However, this method is not suitable for automation and may hide certain subtleties from the one inspecting it.

One way to test for stationarity is by using a non-parametric test known as the Kolmogorov-Smirnov test. It is designed to measure the distance between two distributions. If we have a single chain, we can subsample it to get those two distributions $\{x_{s1}^{(t)}\}_{1\leq t\leq N}$ and $\{x_{s2}^{(t)}\}_{1\leq t\leq N}$ ($s1$ and $s2$ are indicators for the sub-chains). A critical requirement we need to satisfy is the independence of points in each sample in order for the test to be meaningful at all. The test is determined by the following statistic:

$$D = \frac{1}{N}\sup_{\eta}\left|\sum_{i=1}^{N}\mathbb{I}_{[0,\eta]}(x_{s1}^{(i)}) - \sum_{i=1}^{N}\mathbb{I}_{[0,\eta]}(x_{s2}^{(i)})\right|,$$

where $\mathbb{I}$ is the indicator function. Various heuristics can be used to process D in order to specify a stopping rule. When $N \to \infty$:

$$\mathbb{P}(\sqrt{N}D \leq x) = 1 - \sum_{k=1}^{\infty}(-1)^{k-1}e^{-2k^2x^2}. \tag{1.8}$$

A p-value can be derived from this *cdf* (1.8) for instance. A drawback of this method is that if the exploration gets stuck in one region, it will most likely give misleading results.

1.4.3 Convergence of averages

Perhaps a more tractable diagnosis is assessing up to what degree the ergodic theorem is satisfied through Monte Carlo computation of some statistic. However, the assumptions made to develop the Monte Carlo convergence and robustness results may not be met in MCMC samples. An obvious difference is the dependence induced by conditioning on past points. In this situation, it is specially crucial that the chain has explored a significant portion of the space of f. For instance, if it gets stuck at some mode and did not explore others, the approximation could be off by a lot.

For graphical diagnosis, one needs to plot the estimated statistic of interest over iterations and check that it is converging to a single value.

Suppose we have multiple chains $\{\theta_s^{(t)}\}_t$, with $(s,t) \in \{1,\ldots,S\} \times \{1,\ldots,T\}$. First, let's define $\beta_s^{(t)} = h(\theta_s^{(t)})$ and the following computations:

$$\bar{\beta}_s = \frac{1}{T}\sum_{t=1}^{T}\beta_s^{(t)}, \text{ and } \bar{\beta}_\cdot = \frac{1}{S}\sum_{s=1}^{S}\bar{\beta}_s.$$

We restrict h to the set of integrable functions. Then the between chains variance B_T and the within-chain variance W_T are defined by:

$$B_T = \frac{1}{S-1}\sum_{s=1}^{S}(\bar{\beta}_s - \bar{\beta}_\cdot)^2,$$

$$W_T = \frac{1}{S-1}\sum_{s=1}^{S}\hat{\sigma}_s^2 := \frac{1}{(S-1)(T-1)}\sum_{s=1}^{S}\sum_{t=1}^{T}(\beta_s^{(t)} - \bar{\beta}_s)^2.$$

Figure 1.6: MCMC samples from different initial values. Figure generated by mcmcGmmDemo.

We can then use them to study properties of the variance of the target distribution. For instance, we could monitor the estimated potential scale reduction defined as :

$$\hat{R} = \sqrt{\frac{T-1}{T} + \frac{B_T}{W_T}}.$$

This quantity can be interpreted as the degree to which the variance B_T will drop as $T \to \infty$, at which point the chains should have converged to the same target region. When it gets close to 1, we may assume to have reached a stopping point. Now this method assumes one is dealing with normal approximations, a very strong assumption in the context of most interesting MCMC settings. See Figure 1.6 for some examples. The samples come from a mixture of two Gaussians using four methods. To simulate from in this case, the mixture density is considered as f for the first three examples. The first three come from MHA with different proposals q highlighted in the figures (normal distributions with different variances) and the last is from GS. Figure 1.6 (a) shows that the chains have not really mixed as they tend to get stuck in the initial region of their starting points. The others don't have a mixing problem, but they have differences such as the high degree of correlation between samples clearly seen in figure 1.6 (c), which is not a problem with the samples coming from Gibbs (GS).

Since the whole raw chain is likely not to be used for Monte Carlo estimations, an interesting question that arises is how many iid data points can we get from it? This is known as the effective sample size (ESS). Let consider that we are interested in the quantity $\frac{1}{T}\sum_{t=1}^{T} h(\theta^{(t)})$, then the effective sample size is obtained by (Robert and Casella, 2009):

$$\tau = \frac{T}{\kappa(h)}, \tag{1.9}$$

where

$$\kappa(h) = 1 + 2\sum_{t=1}^{\infty} \text{corr}\Big(h(\theta^{(0)}), h(\theta^{(t)})\Big). \tag{1.10}$$

Here $\kappa(h)$ is the autocorrelation function evaluated on the transformed sequence $h(\theta^{(t)})$.

1.4.4 Convergence to i.i.d sampling

The principle is to subsample the sequence $\{\theta_1^{(t)}, \ldots, \theta_n^{(t)}\}$ based on the covariance function $\text{cov}_f(\theta^{(0)}, \theta^{(t)})$. For instance, the chain could be reduced by the selection of every other $\kappa(h)$ (as defined in (1.10), which we will just refer to as κ) variable such that $\eta^{(t)} = \theta^{(t\kappa)}$. Since the bottom line of obtaining the sample is to use it for approximating quantities, it can be shown that considering the whole chain achieves better results toward that goal under certain conditions.

Some studies have explored the possibility of using multiple chains $\{\theta_m^{(t)}, 1 \leq m \leq C\}$ in order to get more robust evaluations. However, the use of multiple chains almost certainly requires that we have some control over the distributions of the starting points $\theta_m^{(0)}$, which paradoxically requires conditioning at least partially on f which we don't know. Taking that distribution to be uniform is a way to represent this prior ignorance or lack of information.

Finally, note that in practice there exists an R package "coda" (Plummer et al., 2006) specific for exploring convergence of Markov chains. This tool includes diagnostic functions producing very explicit evaluations of the number of simulations to use to estimate a given quantile of the target distribution with a given precision. Several theoretical foundations of this convergence diagnostic have been developed in the setting of Markov chains (including time series). For an account on the subject we refer the reader to (Robert and Casella, 2004, Section 12.4.1).

Chapter 2

Bayesian linear models

Linear regression models represent a set of solutions to the statistical problem of finding unknown distributions using data. A key assumption in a linear model is that a linear combination of the explanatory variables $\boldsymbol{X}$ can be used to derive useful information about $\boldsymbol{y}$. Depending on the quality and amount of information we seek to know about the latter variable of interest, two main paradigms are potential choices for this kind of statistical inference. If we only need to know $\mathbb{E}(\boldsymbol{y}|\boldsymbol{X})$ and similar expectations of interest, a basic linear regression model known as the Ordinary Least Squares (OLS) regression is the most used method. However, if we need to obtain the full information, namely $p(\boldsymbol{y}|\boldsymbol{X})$, the Bayesian approach becomes very important.

Before we dive into the subject, first and foremost let us introduce a few fundamental distributions that we will be dealing with throughout the rest of the chapter.

Definition 7 (*Multivariate Normal Distribution*)**.** *Let $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$ a positive semi-definite matrix. The multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ has a density with respect to the Lebesgue measure given by:*

$$p(\boldsymbol{x}) = (2\pi)^{-n/2}|\boldsymbol{\Sigma}|^{-1/2} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}. \tag{2.1}$$

Definition 8 (*Gamma* and *Inverted Gamma*)**.** *Let $a, b > 0$. The gamma distribution $G(a, b)$ has a density with respect to the Lebesgue measure given by:*

$$p(x) = \frac{b^a}{\Gamma(a)} x^{a-1} \exp(-bx), \text{ for all } x \geq 0. \tag{2.2}$$

The inverted gamma distribution $IG(a, b)$ (that is, the distribution of the inverse of a gamma variable) has a density

$$p(x) = \frac{b^a}{\Gamma(a)} (1/x)^{a+1} \exp(-b/x), \text{ for all } x \geq 0. \tag{2.3}$$

Definition 9 (*Multivariate t-distribution*). *Let $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$ a positive semi-definite matrix. The multivariate t-distribution with $v > 0$ degree of freedom denoted $\mathcal{T}_v(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ has a density given by:*

$$p(\boldsymbol{x}) = \frac{\Gamma(\frac{v+n}{2})|\boldsymbol{\Sigma}|^{-1/2}}{\Gamma(v/2)(\pi n)^{n/2}} \left(1 + \frac{1}{v}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)^{-\frac{v+n}{2}}. \tag{2.4}$$

Definition 10 (*Normal Inverse Gamma distribution*). *Let $a, b, \sigma^2 > 0$, $\boldsymbol{\beta} \in \mathbb{R}^p$, $\boldsymbol{\mu} \in \mathbb{R}^p$ and $\boldsymbol{V} \in \mathbb{R}^{p\times p}$ a positive semi-definite matrix. The Normal Inverse Gamma distribution $NIG(\boldsymbol{\mu}, \boldsymbol{V}, a, b)$ has a density:*

$$p(\boldsymbol{\beta}, \sigma^2) = \frac{b^a(\sigma^2)^{-(a+p/2+1)}}{(2\pi)^{p/2}|\boldsymbol{V}|^{1/2}\Gamma(a)} \exp\left(-\frac{1}{2\sigma^2}\Big\{(\boldsymbol{\beta}-\boldsymbol{\mu})^T\boldsymbol{V}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}) + 2b\Big\}\right). \tag{2.5}$$

The purpose of the previous definitions is to provide a self-contained entry into Bayesian linear models.

2.1 Ordinary least square regression

We consider data consisting of a target variable $\boldsymbol{y}$ and regressors $\boldsymbol{X}$, defined respectively as:

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_n) \in \mathbb{R}^n \qquad \text{and} \qquad \boldsymbol{X} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n})^T \in \mathbb{R}^{n\times p},$$

where $\boldsymbol{x_i} = (x_{i1}, x_{i2}, \ldots, x_{ip}) \in \mathbb{R}^p$.

In general, a linear regression is a particular model that ensures the smoothness of the change of the target variable $\boldsymbol{y}$ with respect to the regressors $\boldsymbol{X}$. The key condition for such a model is that $\mathbb{E}(\boldsymbol{y}|\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{\beta}$.

The most commonly used OLS method is based on the *normal linear regression model*. It can be given by the usual linear regression model as follows:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \qquad \text{with} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I_n}), \quad \boldsymbol{\epsilon} \in \mathbb{R}^n$$

where $\boldsymbol{I_n}$ denotes the identity matrix of dimension $n \times n$.

The likelihood can be deduced from the Gaussian distribution of the error term $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$ as $\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2 \sim \boldsymbol{\mathcal{N}}(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I_n})$ where its density with respect to the Lebesgue measure is given by

$$p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2) = (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\right).$$

We have n observations consisting of $(y_i, \boldsymbol{x_i})$ pairs, $\boldsymbol{x_i}$ being a vector of p explanatory variables. The goal is to find a vector $\boldsymbol{\beta}$ of p parameters such that the quantity $\phi(\boldsymbol{\beta}) = (\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})$

is minimized and its associated variance σ^2. We find the optimal $\hat{\boldsymbol{\beta}}$ as follows:

$$\begin{aligned}\phi^{'}(\boldsymbol{\beta}) = \frac{d}{d\boldsymbol{\beta}}\phi(\boldsymbol{\beta}) \qquad &= \frac{d}{d\boldsymbol{\beta}}\left(\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{\beta}\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{\beta}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}\right) \\ &= -2\boldsymbol{X}^T\boldsymbol{y} + \mathbf{2}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}\end{aligned}$$

$$\phi^{'}(\hat{\boldsymbol{\beta}}) = 0 \qquad \Leftrightarrow \quad \hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}.$$

As long as $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ is invertible, this optimal value will be unique. The variance is found to be:

$$\hat{\sigma}^2 = \frac{1}{n-p}||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||^2$$

It is only defined when $n > p$. Otherwise this whole model structure is not applicable. Now suppose that $\boldsymbol{\beta}$ is random and we were interested in its actual distribution or that of any of these variables above except those that are given by definition (in other words, deterministic), the above minimization exercise would not be sufficient. In such scenario, we need to use some Bayesian tools which are based for the most part on the Bayes' rule specified below:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)},$$

where $\theta \in \mathbb{R}^p$ represents the parameters (σ^2 is obtained the same way). Even though we might not have any information on θ to begin with, we need to know $p(\theta)$ in advance since Bayes' rule, even though powerful, is only about updating the parameters while taking into account the new information (the data). That's why in Bayesian inference, we need a particular distribution called the *prior distribution*. Combined with the *likelihood* $p(y|\theta)$, which represents the new information seen through the lens of our initial θ, it yields a *posterior distribution* $p(\theta|y)$ which characterizes the updated information.

Depending on whether or not we have prior information or knowledge, there are two fundamental types of prior distribution. In the informative type, we select a prior that is conjugate or semi-conjugate (this guarantees that we can find the functional form of the posterior) for the sampling model of the data given the parameters whenever possible. Otherwise, the use of any other distribution which fits well the initial knowledge works too, although it is highly likely that finding or doing inference on the posterior would be more challenging. In a situation where we have no significant or no previous information at all, we use non-informative priors, also known as uninformative priors. For more information on the subject, we refer the reader to Marin and Robert (2007).

2.2 Bayesian inference with conjugate prior

The conjugate prior we'll focus on in this section is the $Normal\ Inverse\ Gamma$ prior (NIG). In this setting, we are interested in determining the distribution of both β and σ^2.

Proposition 1. *Assume that* $\sigma^2 \sim IG(a, b)$ *and* $\boldsymbol{\beta}|\sigma^2 \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\boldsymbol{V})$*, then* $(\boldsymbol{\beta}, \sigma^2) \sim NIG(\boldsymbol{\mu}, \boldsymbol{V}, a, b)$*.*

Proof. Under the assumptions of Proposition 1, we compute

$$\begin{aligned} p(\boldsymbol{\beta}, \sigma^2) &= p(\boldsymbol{\beta}|\sigma^2)p(\sigma^2) \\ &= \frac{(2\pi\sigma^2)^{-p/2}}{|\boldsymbol{V}|^{1/2}} \exp\left\{ -\frac{1}{2\sigma^2}(\boldsymbol{\beta}-\boldsymbol{\mu})^T\boldsymbol{V}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu})\right\} \times \frac{b^a}{\Gamma(a)}(\sigma^2)^{-a+1}\exp\left(-\frac{b}{\sigma^2}\right) \\ &= \frac{b^a(\sigma^2)^{-(a+p/2+1)}}{(2\pi)^{p/2}|\boldsymbol{V}|^{1/2}\Gamma(a)} \exp\left(-\frac{1}{2\sigma^2}\{(\boldsymbol{\beta}-\boldsymbol{\mu})^T\boldsymbol{V}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu})+2b\}\right). \end{aligned}$$

This expression can be read from definition 10. □

To deal with the complexity of the following developments, let's define a helper function:

$$\psi(\boldsymbol{\delta}, \mathbf{M}) = \boldsymbol{\delta}^T\mathbf{M}^{-1}\boldsymbol{\delta},$$

where $\boldsymbol{\delta}$ and $\mathbf{M}$ are respectively a vector and a matrix of real numbers. A nice feature of the joint distribution in Proposition 1 is the explicit term of the marginal distribution of $\sigma^2|\boldsymbol{\beta}$ and $\boldsymbol{\beta}$.

Proposition 2. *Let us assume the joint distribution* $(\boldsymbol{\beta}, \sigma^2) \sim NIG(\boldsymbol{\mu}, \boldsymbol{V}, a, b)$*. Then, we have*

$$\begin{cases} \boldsymbol{\beta} \sim \mathcal{T}_{2a}(\boldsymbol{\mu}, \frac{b}{a}\boldsymbol{V}), \\ \sigma^2|\boldsymbol{\beta} \sim IG\left(a + p/2, \frac{\psi(\boldsymbol{\beta}-\boldsymbol{\mu}, \boldsymbol{V})}{2} + b\right). \end{cases}$$

Proof. Starting from the joint distribution of $(\boldsymbol{\beta}, \sigma^2)$, we have

$$p(\boldsymbol{\beta}, \sigma^2) = \frac{b^a}{(2\pi)^{p/2}|\boldsymbol{V}|^{1/2}\Gamma(a)} \times (\sigma^2)^{-(a+p/2+1)} \exp\left(-\frac{1}{2\sigma^2}\{(\boldsymbol{\beta}-\boldsymbol{\mu})^T\boldsymbol{V}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu})+2b\}\right).$$

A direct calculation gives

$$p(\sigma^2|\boldsymbol{\beta}) = C(\sigma^2)^{-(a+p/2+1)} \exp\left\{-\frac{1}{2\sigma^2}[\psi(\boldsymbol{\beta}-\boldsymbol{\mu}, \boldsymbol{V}) + 2b]\right\},$$

where

$$C = \frac{1}{\Gamma(a+p/2)}\left[\frac{\psi(\boldsymbol{\beta}-\boldsymbol{\mu}, \boldsymbol{V})}{2} + b\right]^{a+p/2}.$$

Hence, we deduce that:

$$\sigma^2|\boldsymbol{\beta} \sim IG\left\{a + p/2, \frac{\psi(\boldsymbol{\beta}-\boldsymbol{\mu}, \boldsymbol{V})}{2} + b\right\}.$$

Since $p(\boldsymbol{\beta}, \sigma^2) = p(\sigma^2|\boldsymbol{\beta})p(\boldsymbol{\beta})$, we can derive the explicit expression of $p(\boldsymbol{\beta})$ easily

$$\begin{aligned} p(\boldsymbol{\beta}) &= \frac{1}{(2\pi)^{p/2}|\boldsymbol{V}|^{1/2}\Gamma(a)}\frac{b^a}{C} \\ &= \frac{b^a\Gamma(2a+p/2)}{(2\pi)^{p/2}|\boldsymbol{V}|^{1/2}\Gamma(a)}\Big[\frac{\psi(\boldsymbol{\beta}-\boldsymbol{\mu},\boldsymbol{V})}{2}+b\Big]^{-(a+p/2)} \\ &= \frac{\Gamma(\frac{2a+p}{2})}{(2\pi)^{p/2}|2a\boldsymbol{V}\frac{b}{a}|^{1/2}\Gamma(a)}\Big[\frac{\psi(\boldsymbol{\beta}-\boldsymbol{\mu},\frac{b}{a}\boldsymbol{V})}{2a}+1\Big]^{-(\frac{2a+p}{2})}. \end{aligned}$$

Finally, we see that

$$\boldsymbol{\beta} \sim \mathcal{T}_{2a}\Big(\boldsymbol{\mu}, \frac{b}{a}\boldsymbol{V}\Big),$$

as seen in definition 9. □

As a reminder, let's write down the likelihood function, which represents the probability of observing the data given the values of the parameters:

$$p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2) = (2\pi\sigma^2)^{-n/2}\exp\Big\{-\frac{1}{2\sigma^2}\psi(\boldsymbol{y}-\boldsymbol{X\beta}, \boldsymbol{I_n})\Big\}.$$

Now that we have specified the marginal distributions of all parameters, let's move to the derivation of the joint posterior distribution $p(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y})$. We will use a very neat identity called the *multivariate completion of squares*.

Lemma 3. *Let $\boldsymbol{u} \in \mathbb{R}^k$, $\boldsymbol{\alpha} \in \mathbb{R}^k$ and $\boldsymbol{A}$ be a square matrix in $\mathbb{R}^{k\times k}$. We have the following identity:*

$$\boldsymbol{u}^T\boldsymbol{Au} - 2\boldsymbol{\alpha}^T\boldsymbol{u} = (\boldsymbol{u}-\boldsymbol{A}^{-1}\boldsymbol{\alpha})^T\boldsymbol{A}(\boldsymbol{u}-\boldsymbol{A}^{-1}\boldsymbol{\alpha}) - \boldsymbol{\alpha}^T\boldsymbol{A}^{-1}\boldsymbol{\alpha}. \tag{2.6}$$

Proof. We simply expend the following quadratic term:

$$\begin{aligned} (\boldsymbol{u}-\boldsymbol{A}^{-1}\boldsymbol{\alpha})^T\boldsymbol{A}(\boldsymbol{u}-\boldsymbol{A}^{-1}\boldsymbol{\alpha}) &= \boldsymbol{u}^T\boldsymbol{Au} - \boldsymbol{u}^T\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\boldsymbol{u} + \boldsymbol{\alpha}^T\boldsymbol{A}^{-1}\boldsymbol{\alpha} \\ &= \boldsymbol{u}^T\boldsymbol{Au} - 2\boldsymbol{\alpha}^T\boldsymbol{u} + \boldsymbol{\alpha}^T\boldsymbol{A}^{-1}\boldsymbol{\alpha} \end{aligned}$$

A rearrangement of terms leads to (2.6). □

Using the identity (2.6), we may derive the posterior distribution of $(\boldsymbol{\beta}, \sigma^2)$ as follows:

Proposition 3. *Let us consider the following priors and likelihood*

$$\begin{aligned} \boldsymbol{y}|\boldsymbol{\beta}, \sigma^2 &\sim \boldsymbol{\mathcal{N}}(\boldsymbol{X\beta}, \sigma^2\boldsymbol{I_n}), \\ \boldsymbol{\beta}|\sigma^2 &\sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\boldsymbol{V}), \\ \sigma^2 &\sim IG(a, b). \end{aligned} \tag{2.7}$$

.

Then, the posterior distribution of $(\boldsymbol{\beta}, \sigma^2)$ *is given by*

$$\boldsymbol{\beta}, \sigma^2 | \boldsymbol{y} \sim NIG(\boldsymbol{\mu}^*, \boldsymbol{V}^*, a^*, b^*), \tag{2.8}$$

where

$$\begin{aligned} \boldsymbol{V}^* &= (\boldsymbol{V}^{-1} + \boldsymbol{X}^T\boldsymbol{X})^{-1} \\ \boldsymbol{\mu}^* &= \boldsymbol{V}^*(\boldsymbol{V}^{-1}\boldsymbol{\mu} + \boldsymbol{X}^T\boldsymbol{y}) \\ b^* &= b + \frac{1}{2}\{-\boldsymbol{\mu}^{*T}\boldsymbol{V}^{*-1}\boldsymbol{\mu}^* + \boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{\mu}^T\boldsymbol{V}^{-1}\boldsymbol{\mu}\} \\ a^* &= a + n/2. \end{aligned} \tag{2.9}$$

Proof. Using the priors in (2.7) together with Bayes' rule we can write

$$\begin{aligned} p(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y}) &\propto p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2)p(\boldsymbol{\beta}, \sigma^2) \\ &\propto \frac{b^a(\sigma^2)^{-n/2}(\sigma^2)^{-(a+p/2+1)}}{(2\pi)^{n/2}(2\pi)^{p/2}|V|^{1/2}\Gamma(a)} \exp\left\{-\frac{1}{2\sigma^2}[\psi(\boldsymbol{\beta}-\boldsymbol{\mu}, \boldsymbol{V}) + \psi(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{I_n}) + 2b]\right\}. \end{aligned}$$

Direct computation gives

$$\begin{aligned} \psi(\boldsymbol{\beta}-\boldsymbol{\mu}, \boldsymbol{V}) + \psi(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{I_n}) &= (\boldsymbol{\beta}-\boldsymbol{\mu})^T\boldsymbol{V}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}) + (\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}) \\ &= \boldsymbol{\beta}^T\boldsymbol{V}^{-1}\boldsymbol{\beta} - 2\boldsymbol{\mu}^T\boldsymbol{V}^{-1}\boldsymbol{\beta} + \boldsymbol{\mu}^T\boldsymbol{V}^{-1}\boldsymbol{\mu} + \boldsymbol{y}^T\boldsymbol{y} \\ &\quad - 2\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} \\ &= \boldsymbol{\beta}^T(\boldsymbol{V}^{-1} + \boldsymbol{X}^T\boldsymbol{X})\boldsymbol{\beta} - 2(\boldsymbol{\mu}^T\boldsymbol{V}^{-1} \\ &\quad + \boldsymbol{y}^T\boldsymbol{X})\boldsymbol{\beta} + \boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{\mu}^T\boldsymbol{V}^{-1}\boldsymbol{\mu}. \end{aligned}$$

In the sequel, let us put

$$\boldsymbol{u} = \boldsymbol{\beta}, \quad \boldsymbol{A} = \boldsymbol{V}^{-1} + \boldsymbol{X}^T\boldsymbol{X}, \quad \textit{and} \quad \boldsymbol{\alpha}^T = \boldsymbol{\mu}^T\boldsymbol{V}^{-1} + \boldsymbol{y}^T\boldsymbol{X}.$$

We obtain straightforwardly

$$\begin{aligned} \frac{1}{2}\{\psi(\boldsymbol{\beta}-\boldsymbol{\mu}, \boldsymbol{V}) + \psi(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{I_n})\} + b &= \frac{1}{2}\{(\boldsymbol{\beta}-\boldsymbol{\mu}^*)^T\boldsymbol{V}^{*-1}(\boldsymbol{\beta}-\boldsymbol{\mu}^*) - \boldsymbol{\alpha}^T\boldsymbol{A}^{-1}\boldsymbol{\alpha} \\ &\quad + \boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{\mu}^T\boldsymbol{V}^{-1}\boldsymbol{\mu}\} + b \\ &= \frac{1}{2}\psi(\boldsymbol{\beta}-\boldsymbol{\mu}^*, \boldsymbol{V}^*) + b^*, \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{V}^* &= (\boldsymbol{V}^{-1} + \boldsymbol{X}^T\boldsymbol{X})^{-1} = \boldsymbol{A}^{-1} \\ \boldsymbol{\mu}^* &= \boldsymbol{V}^*(\boldsymbol{V}^{-1}\boldsymbol{\mu} + \boldsymbol{X}^T\boldsymbol{y}) = \boldsymbol{A}^{-1}\boldsymbol{\alpha}^T \\ b^* &= b + \frac{1}{2}\{-\boldsymbol{\alpha}^T\boldsymbol{A}^{-1}\boldsymbol{\alpha} + \boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{\mu}^T\boldsymbol{V}^{-1}\boldsymbol{\mu}\} \\ &= b + \frac{1}{2}\{-\boldsymbol{\mu}^{*T}\boldsymbol{V}^{*-1}\boldsymbol{\mu}^* + \boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{\mu}^T\boldsymbol{V}^{-1}\boldsymbol{\mu}\} \\ a^* &= a + n/2. \end{aligned}$$

Finally, we have

$$p(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y}) \propto (\sigma^2)^{-(a^*+p/2+1)} \exp\left\{-\frac{1}{2\sigma^2}[\psi(\boldsymbol{\beta}-\boldsymbol{\mu^*}, \boldsymbol{V^*}) + 2b^*]\right\}.$$

We conclude the proof by writing

$$\boldsymbol{\beta}, \sigma^2|\boldsymbol{y} \sim NIG(\boldsymbol{\mu^*}, \boldsymbol{V^*}, a^*, b^*).$$

□

Now, we will deduce the prior predictive distribution (marginal distribution of y).

Proposition 4. *The prior predictive distribution given σ^2 is:*

$$\boldsymbol{y}|\sigma^2 \sim \boldsymbol{\mathcal{N}}(\boldsymbol{X\mu}, \sigma^2(\boldsymbol{I_n} + \boldsymbol{XVX}^T)).$$

Proof. Let $\boldsymbol{\epsilon_0} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{V})$ and $\boldsymbol{\epsilon_1} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I_n})$. Suppose that $\boldsymbol{\epsilon_0}$ and $\boldsymbol{\epsilon_1}$ are independent. We have $\boldsymbol{\beta}|\sigma^2 \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\boldsymbol{V})$. This means that we can write $\boldsymbol{\beta} = \boldsymbol{\mu} + \boldsymbol{\epsilon_0}$. Since the likelihood is $\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2 \sim \boldsymbol{\mathcal{N}}(\boldsymbol{X\beta}, \sigma^2\boldsymbol{I_n})$, then we have $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon_1} = \boldsymbol{X\mu} + \boldsymbol{X\epsilon_0} + \boldsymbol{\epsilon_1}$. So by the Gaussian property, we conclude that:

$$\boldsymbol{y}|\sigma^2 \sim \boldsymbol{\mathcal{N}}(\boldsymbol{X\mu}, \sigma^2(\boldsymbol{I_n} + \boldsymbol{XVX}^T)).$$

□

Proposition 5. *The prior predictive distribution is given by:*

$$\boldsymbol{y} \sim \mathcal{T}_{2a}\left\{\boldsymbol{X\mu}, \frac{b}{a}(\boldsymbol{I_n} + \boldsymbol{XVX}^T)\right\}.$$

Proof. We use the results of Proposition 2 to find $p(\boldsymbol{y})$. By replacing $\boldsymbol{\beta}$ with $\boldsymbol{y}$, we get:

$$\boldsymbol{y}|\sigma^2 \sim \boldsymbol{\mathcal{N}}(\boldsymbol{X\mu}, \sigma^2(\boldsymbol{I_n} + \boldsymbol{XVX}^T)) \qquad \textit{and} \quad \sigma^2 \sim IG(a, b).$$

We see immediately that

$$(\boldsymbol{y}, \sigma^2) \sim NIG(\boldsymbol{X\mu}, (\boldsymbol{I_n} + \boldsymbol{XVX}^T), a, b).$$

Finally, we have

$$\boldsymbol{y} \sim \mathcal{T}_{2a}\left\{\boldsymbol{X\mu}, \frac{b}{a}(\boldsymbol{I_n} + \boldsymbol{XVX}^T)\right\}.$$

□

Now suppose that we have new data $\tilde{\boldsymbol{X}}$ for which we want to predict the values $\tilde{\boldsymbol{y}}$. Ideally the latter should be distributed according to a normal with mean $\tilde{\boldsymbol{X}}\boldsymbol{\beta}$ and a dispersion parameter equal to σ^2. However, $\boldsymbol{\beta}$ and σ are not known. So we will get the desired quantities of interest through the predictive posterior distribution of $\boldsymbol{y}$, which will include the estimation of the two unknown parameters.

Proposition 6. *The posterior predictive distribution is given by:*

$$\tilde{\boldsymbol{y}}|\boldsymbol{y} \sim \mathcal{T}_{2a^*}\left\{\tilde{\boldsymbol{X}}\boldsymbol{\mu}^{\boldsymbol{*}}, \frac{b^*}{a^*}(\tilde{\boldsymbol{I_n}} + \tilde{\boldsymbol{X}}\boldsymbol{V}^{\boldsymbol{*}}\tilde{\boldsymbol{X}}^T)\right\}.$$

Proof. We use the fact that the following density integrates to 1 over $\boldsymbol{\beta}$ and σ^2:

$$p(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2)p(\boldsymbol{\beta}, \sigma^2)}{p(\boldsymbol{y})} = \frac{\mathcal{N}(\boldsymbol{X\beta}, \sigma^2\boldsymbol{I_n}) * NIG(\boldsymbol{\mu}, \boldsymbol{V}, a, b)}{\mathcal{T}_{2a}(\boldsymbol{X\mu}, \frac{b}{a}(\boldsymbol{I_n} + \boldsymbol{XVX}^T))}.$$

Integrating the numerator with respect to $\boldsymbol{\beta}$ and σ^2 yields exactly the denominator. So, we have

$$\begin{aligned} p(\tilde{\boldsymbol{y}}|\boldsymbol{y}) &= \int p(\tilde{\boldsymbol{y}}|\boldsymbol{\beta}, \sigma^2)p(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y})d\boldsymbol{\beta}d\sigma^2 \\ &= \int \mathcal{N}(\tilde{\boldsymbol{X}}\boldsymbol{\beta}, \sigma^2\boldsymbol{I}_{\tilde{\boldsymbol{n}}}) \times NIG\left\{\boldsymbol{\mu}^{\boldsymbol{*}}, \boldsymbol{V}^{\boldsymbol{*}}, a^*, b^*\right\} d\boldsymbol{\beta}d\sigma^2. \end{aligned}$$

We obtain finally

$$\tilde{\boldsymbol{y}}|\boldsymbol{y} \sim \mathcal{T}_{2a^*}\left\{\tilde{\boldsymbol{X}}\boldsymbol{\mu}^{\boldsymbol{*}}, \frac{b^*}{a^*}(\boldsymbol{I}_{\tilde{\boldsymbol{n}}} + \tilde{\boldsymbol{X}}\boldsymbol{V}^{\boldsymbol{*}}\tilde{\boldsymbol{X}}^T)\right\}.$$

□

Using simulation methods such as the MCMC algorithms, we can sample from these distributions by proceeding as follows. We sample $(\sigma^2)^{(t)} \sim IG(a^*, b^*)$, then $\boldsymbol{\beta}^{(t)}|(\sigma^2)^{(t)} \sim \mathcal{N}(\boldsymbol{\mu}^{\boldsymbol{*}}, (\sigma^2)^{(t)}\boldsymbol{V}^{\boldsymbol{*}})$. The sequence $\left\{\boldsymbol{\beta}^{(t)}, (\sigma^2)^{(t)}\right\}_{t=1}^{T}$ gives us both the joint and marginal distributions. For each $(\boldsymbol{\beta}^{(t)}, (\sigma^2)^{(t)})$, we can sample $\tilde{\boldsymbol{y}}^{(t)} \sim \mathcal{N}(\tilde{\boldsymbol{X}}\boldsymbol{\beta}^{(t)}, (\sigma^2)^{(t)}\boldsymbol{I_n})$ which gives us the posterior predictive observations. Of course, we can always sample directly from the derived distributions above as well.

To sum up, if $\sigma^2 \sim IG(a, b)$, $\boldsymbol{\beta}|\sigma^2 \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\boldsymbol{V})$ and $\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2 \sim \boldsymbol{\mathcal{N}}(\boldsymbol{X\beta}, \sigma^2\boldsymbol{I_n})$, then we have the following results:

$$\begin{aligned} \boldsymbol{\beta} \sim \mathcal{T}_{2a}\left(\boldsymbol{\mu}, \frac{b}{a}\boldsymbol{V}\right) \quad &\& \quad \sigma^2|\boldsymbol{\beta} \sim IG\left\{a + p/2, \frac{\psi(\boldsymbol{\beta} - \boldsymbol{\mu}, \boldsymbol{V})}{2} + b\right\} \\ (\boldsymbol{\beta}, \sigma^2) \sim NIG(\boldsymbol{\mu}, \boldsymbol{V}, a, b) \quad &\& \quad \boldsymbol{\beta}, \sigma^2|\boldsymbol{y} \sim NIG\left\{\boldsymbol{\mu}^{\boldsymbol{*}}, \boldsymbol{V}^{\boldsymbol{*}}, a^*, b^*\right\} \\ (\sigma^*)^2 := \sigma^2|\boldsymbol{y} \sim IG(a^*, b^*) \quad &\& \quad \boldsymbol{\beta}|(\sigma^*)^2 \sim \mathcal{N}\left\{\boldsymbol{\mu}^{\boldsymbol{*}}, (\sigma^*)^2\boldsymbol{V}^{\boldsymbol{*}}\right\} \\ \boldsymbol{\beta}^{\boldsymbol{*}} := \boldsymbol{\beta}|\boldsymbol{y} \sim \mathcal{T}_{2a^*}\left(\boldsymbol{\mu}^{\boldsymbol{*}}, \frac{b^*}{a^*}\boldsymbol{V}^{\boldsymbol{*}}\right) \quad &\& \quad \sigma^2|\boldsymbol{\beta}^{\boldsymbol{*}} \sim IG\left\{a^* + p/2, \frac{\psi(\boldsymbol{\beta}^{\boldsymbol{*}} - \boldsymbol{\mu}^{\boldsymbol{*}}, \boldsymbol{V}^{\boldsymbol{*}})}{2} + b^*\right\} \\ \boldsymbol{y} \sim \mathcal{T}_{2a}\left\{\boldsymbol{X\mu}, \frac{b}{a}(\boldsymbol{I_n} + \boldsymbol{XVX}^T)\right\} & \\ \boldsymbol{y}|\sigma^2 \sim \boldsymbol{\mathcal{N}}(\boldsymbol{X\mu}, \sigma^2(\boldsymbol{I_n} + \boldsymbol{XVX}^T)) \quad &\& \quad \tilde{\boldsymbol{y}}|\boldsymbol{y} \sim \mathcal{T}_{2a^*}\left\{\tilde{\boldsymbol{X}}\boldsymbol{\mu}^{\boldsymbol{*}}, \frac{b^*}{a^*}(\boldsymbol{I}_{\tilde{\boldsymbol{n}}} + \tilde{\boldsymbol{X}}\boldsymbol{V}^{\boldsymbol{*}}\tilde{\boldsymbol{X}}^T)\right\}. \end{aligned}$$

2.3 Bayesian inference with non-informative priors

A non-informative prior is a kind of prior associated with a density function that is not characteristic of a probability distribution. Improper priors are effective because they tend to minimize the impact of prior assumptions. There are various ways to construct such priors. A very important improper prior is obtained by taking parameters of the NIG distribution to their extreme limits. As a result, we get:

$$\begin{aligned} &when \quad \boldsymbol{V}^{-1} \to \mathbf{0}, \qquad a \to -p/2, \quad and \quad b \to 0, \\ &then \quad \boldsymbol{\beta} \to \boldsymbol{\mu} \qquad \Rightarrow \qquad p(\boldsymbol{\beta}, \sigma^2) \propto 1/\sigma^2. \end{aligned}$$

The posterior distributions don't change, however, their computed parameters do. For instance, $p(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y})$ still has a NIG distribution, but:

$$\begin{aligned} \boldsymbol{V^*} &\to (\boldsymbol{X^T X})^{-1}, \\ \boldsymbol{\mu^*} &\to (\boldsymbol{X^T X})^{-1}(\boldsymbol{X}^T\boldsymbol{y}) = \hat{\boldsymbol{\beta}}_{\boldsymbol{ols}}, \\ b^* &\to \frac{1}{2}\Big\{ - [(\boldsymbol{X^T X})^{-1}(\boldsymbol{X}^T\boldsymbol{y})]^T(\boldsymbol{X}^T\boldsymbol{y}) + \boldsymbol{y}^T\boldsymbol{y}\Big\}, \\ &\to \frac{1}{2}\{\boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{X}\hat{\boldsymbol{\beta}}_{\boldsymbol{ols}}\} = \frac{1}{2}\boldsymbol{y}^T(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}_{\boldsymbol{ols}}), \\ a^* &\to (n-p)/2. \end{aligned}$$

2.4 Application of Bayesian inference

We will use simulated data to see how Bayesian inference works. The dataset consists of n identically and independently generated observations of p variables. Each variable follows a normal distribution. The target variable is a linear combination of the latter variables plus a constant and some added noise distributed according to a normal centered around 0 with a variance σ^2. The parameters are chosen as such:

$$n = 50, \qquad p = 4, \qquad \sigma^2 = 10, \qquad \boldsymbol{\beta} = [45, 2, 8, -5]$$

So we get:

$$\mathbf{y} = 45\mathbf{x_0} + 2\mathbf{x_1} + 8\mathbf{x_2} - 5\mathbf{x_3} + \epsilon, \text{ such that } \epsilon \sim \mathcal{N}(0, 10)$$

Assuming we don't know the values of these parameters, the goal here is focused on finding the posterior distribution of $\boldsymbol{\beta}$ and σ^2, and their joint distribution. Since we have all the information we need about the distributions of these variables, we will start by performing a plain Monte Carlo sampling to find the posteriors. Additionally, we will use non-informative priors which means we have no previous information whatsoever about the parameters. As a reminder, we will have that

$$(\sigma^2)^{(t)} \sim IG(a^*, b^*) \qquad and \qquad \boldsymbol{\beta}|(\sigma^2)^{(t)} \sim \mathcal{N}(\boldsymbol{\mu^*}, (\sigma^2)^{(t)}\boldsymbol{V^*}).$$

Figure 2.1: Variable generations under Monte Carlo sampling. The left plot shows the intercept β_0 over 1000 iterations. The right plot shows the variance σ^2 over 1000 iterations.

These parameters are defined in (2.9). We will run 1000 samples and obtain the joint and marginal distributions from the couple $(\boldsymbol{\beta}^{(t)}, (\sigma^2)^{(t)})$. In this case, since we know certainly that the expected value is equal to the OLS estimated mean, the sequence converges almost instantly. Some trace plots result are presented in Figure 2.1.

We see that the structure of these two different random variables are different as expected. Let's look at their distributions.

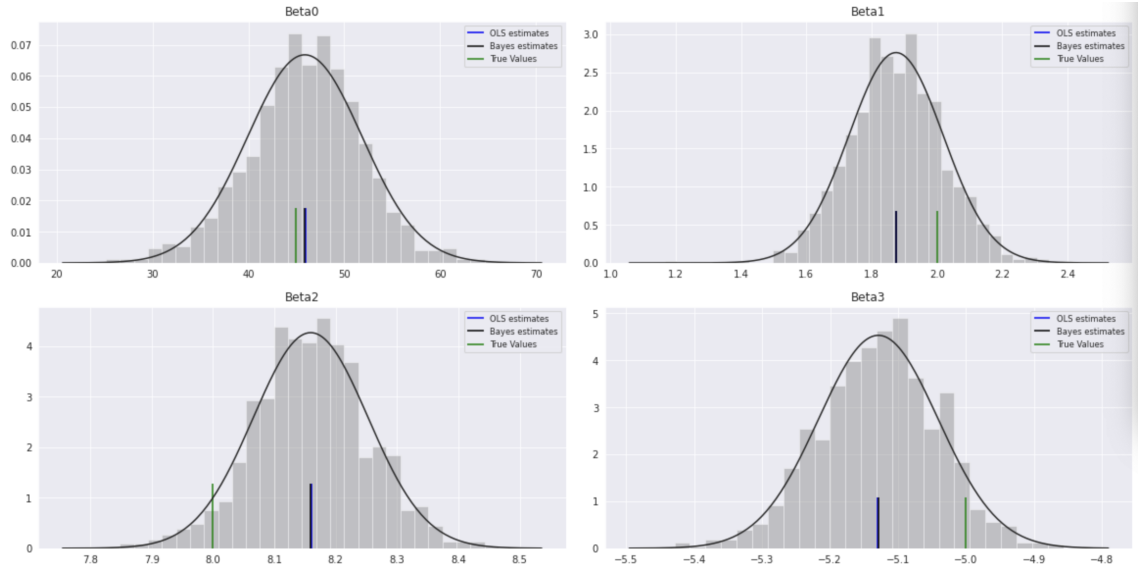


Figure 2.2: Distribution of the regression parameters and their associated mean compared with the true values and the OLS estimates.

As expected the distributions of the parameter values of $\boldsymbol{\beta}$ are normally distributed. Some of them are underestimated since the OLS estimates are to the left of the true value of the parameters, and others are overestimated to balance the predicted value. In practice, looking at the mean squared might help in summarizing the effect of these deviations. We next take a look at some features of σ^2.

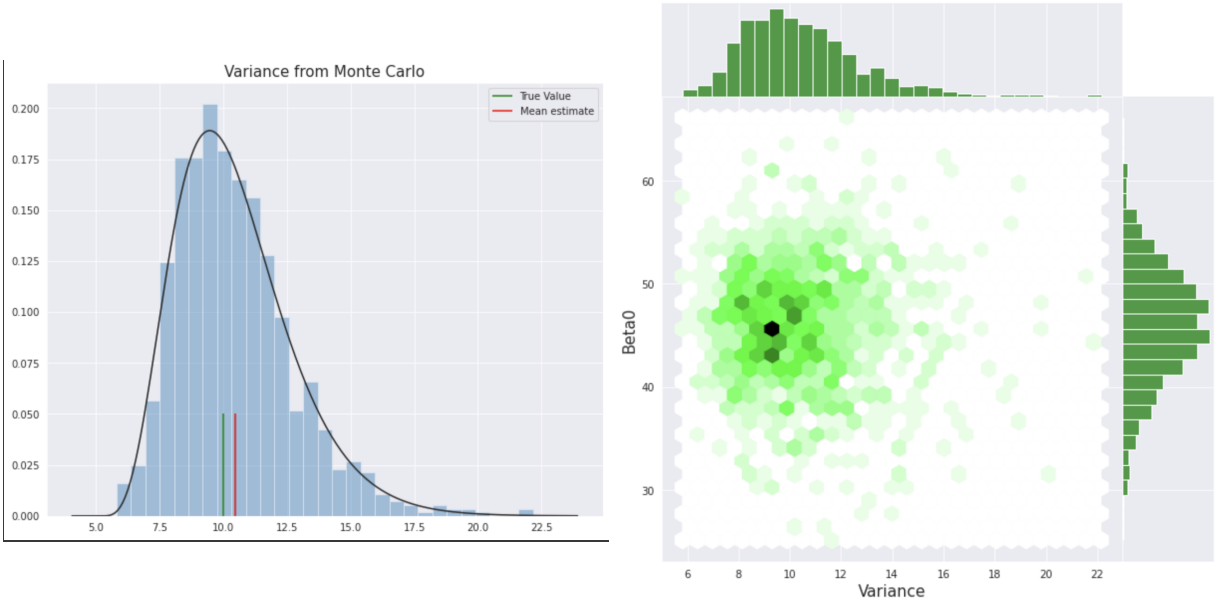


Figure 2.3: The left plot shows distribution of the variance σ^2. The right plot shows joint distribution of the intercept β_0 and the variance σ^2.

The gamma density fits very well the histogram although the expected mean is higher than the true value. This is not always the case as sometimes, it predicts a lower value. In addition, both plots (Figure 2.3) show that σ^2 is not so positively skewed. This is due to a combination of a high variance and a small number of observations. So the uncertainty remains fairly large.

Since we have the full conditional distributions of the parameters of interest, let's try using the Gibbs sampling method, which would be necessary if the evaluation of the integral to get $p(\sigma^2)$ was intractable. In this case, we use the following distributions:

$$(\sigma^2)^{(t+1)}|\boldsymbol{\beta}^{(t)} \sim IG\left\{a^* + p/2, \frac{\psi(\boldsymbol{\beta}^{(t)} - \boldsymbol{\mu^*}, \boldsymbol{V^*})}{2} + b^*\right\},$$

and

$$\boldsymbol{\beta}^{(t+1)}|(\sigma^2)^{(t+1)} \sim \mathcal{N}\left\{\boldsymbol{\mu^*}, (\sigma^2)^{(t+1)}\boldsymbol{V^*}\right\}.$$

There is not much change in the expected values of $\boldsymbol{\beta}$ but the individual variances of its component are generally bigger.

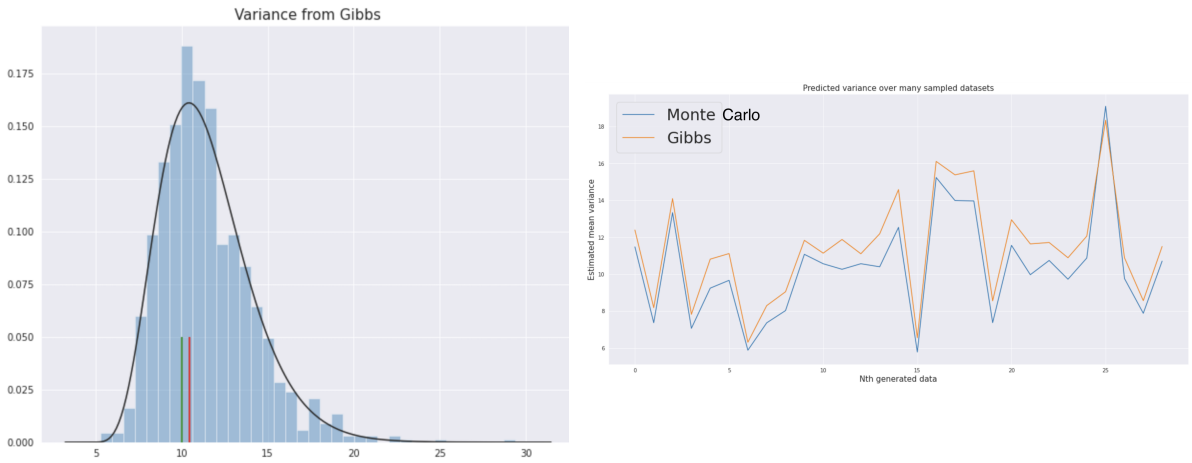


Figure 2.4: The left plot shows distribution of the variance σ^2 from Gibbs sampler. The right plot shows generative chains of the variance σ^2 from Monte Carlo sampling (in blue) and from Gibbs iterations (in orange).

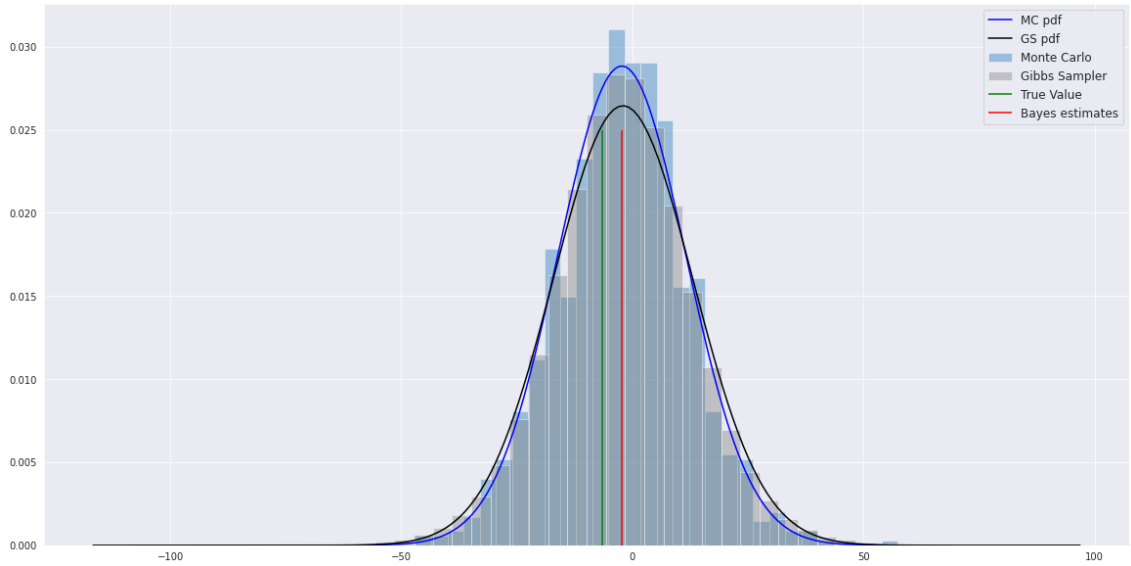


Figure 2.5: Empirical distributions of β_4 from Monte Carlo sampling and Gibbs generation with their respective estimated kernel density.

Figure 2.4 shows the expected values of σ^2 calculated from samples of different generated datasets. We see that the Gibbs sampler tends to predict a higher value for σ^2 since its hyperparameters are updated on every iteration because they dependent on the previous values of $\boldsymbol{\beta}$.

To see how this affects the predictive distributions, let's take a look at the density of β_4 obtained from Gibbs sampling and plain Monte Carlo sampling in Figure 2.5.

Figure 2.5 shows that the distribution found using the Gibbs sampler puts more weight on extreme values and has a lower peak around the mode. This is correlated with the fact that

it predicts higher expected values for the variance. The predictive variance depends on the magnitude of the regressors.

Chapter 3

Bayesian adaptive variable selection with general g-prior

3.1 Introduction to model selection

In the previous chapters, we explored how to conduct inference on parameters once the model has been clearly defined. However, in practice, the model specification itself is not available a priori. So a mechanism is needed in order to select the model structure that would best describe the observed data. This is known as the model selection problem.

One popular approach in statistical learning literature is the use of cross validation techniques which consist of evaluating each model K times on a **test** set (different for each iteration) separate from the **training** set in order to capture its generalization error. The problem with this approach is that only one model is kept and the rest are discarded. A more efficient method is the use of Bayesian tools to compute the posterior over all possible models $p(m|D)$ defined as:

$$p(m|D) = \frac{p(D|m)p(m)}{\sum_{\mathcal{M}} p(m, D)},$$

where m is a model and belongs to the set of all possible models $\mathcal{M}$, and D represents the observed data. The Bayesian model selection procedure is generally about computing the maximum a posteriori (MAP) estimate $\hat{m}$. If we assume all models are a priori equally likely, then it reduces to finding $\hat{m} = \int p(D|\theta)p(\theta|m)d\theta$, where θ represents the parameters of model m. This quantity is known as the **evidence** for model m. Since we are integrating all the model parameters, the selected model is shielded up to a certain degree against overfitting. This is known as the Bayesian Occam's Razor effect (Murray and Ghahramani, 2005). So simpler models are very likely and this makes the Bayesian approach very attractive.

In this chapter, we will concentrate on a more specific type of model selection problem. We will define our hypothesis space to be the set of all combinations of variables x_i. Each subset

is a potential group of regressors that can be used to explain the target variable y. One key property of Bayesian selection procedures is the fact that for a given group G of features, the probability of the latter is computed over a space with dimension $\dim(G)$. As a consequence of that, larger groups will automatically have lower probabilities because they exist in a much larger space over which their probabilities are spread out uniformly a priori. This is how regularization is partially achieved in these Bayesian procedures. Of course the choice of priors is another way to affect it. We'll illustrate these points later with examples.

3.2 Bayesian model selection

In data modeling, it is typical to start by selecting a subset of regressors which will be the only relevant features that explain a target variable y. Mathematically, we will have data of the form $\{y, X_1, ..., X_p\}$ which contains useful features $U := \{X_{i_1}, ..., X_{i_k}\}$, but also features $L := \{X_{j_1}, ..., X_{j_l}\}$ that are not related to the data such that $k + l = p$. U and L constitute a partition of the whole set of features. This separation into two groups is hard to achieve because regressors U are potentially correlated with those in L. The subject of variable selection consists of methods and techniques conceived to find a good partition $[\hat{U}, \hat{L}]$ of the initial set of regressors that best approximates the true partition defined above with respect to some objective function to be optimize. This latter observation is quite important because it means that $[\hat{U}, \hat{L}]$ is not unique, but conditional on the objective function. We will define later more precisely what uniqueness means in this context and how it relates to robustness of a selection algorithm.

Traditional model optimization algorithms suffer from the fact that they are trying to find the mode of the posterior distribution, which is usually an untypical summary of the latter. It is a point of measure zero, unlike the mean or median which take into account the full support distribution. In general, the quantity we are looking for depends on the context of the problem and what we are trying to achieve through solving it. Decision theory provides us with tools we can use to address this issue. Instead of specifying a particular raw summary statistic, it takes into account a function that defines the quality of the predicted values $\hat{y}$ as compared with the observed y. Within the Bayesian context, decision theory revolves around the posterior expected loss defined as:

$$\rho(m_\theta|x) = \mathbb{E}_\theta[\mathcal{L}(y, \theta)|x] = \int \mathcal{L}(y, \theta) dP_\theta(y|x), \tag{3.1}$$

where m_θ is the model specification associated with the parameters θ. The quantity of interest is known as the Bayes estimator and is defined as

$$\delta(x) = \arg \min_{m_\theta \in \mathcal{M}} \rho(m_\theta|x),$$

where $\mathcal{M}$ is the set of all possible models. To the best of my knowledge, getting the full information $p(y|x)$ (whether it be Gaussian or not) is only possible with the use of Bayesian

methods. However, computing $p(y|x)$ analytically is often intractable. A more practical way to approximate it numerically with samples obtained through generative algorithms, such as MCMC methods for instance. The key strength of this method is that the samples are not specific to any statistic we are interested in computing, as opposed to the optimization paradigm which yields a point estimate of only one statistic. We will be using MCMC methods for our models.

3.3 Theoretical framework

Model selection is not independent from data modeling. So we generally start by specifying all distributions underlying the model parameters. Then we introduce a new variable that will allow us to control how the selection will be done. Remember above that the goal of model selection is to separate the regressors into two groups represented as $[U, L]$. The various methods that exist differ in how they treat the parameters related to the irrelevant group L. We reconsider the hierarchical model defined in the previous chapter and add the selection variable γ to the structure. It is defined as a vector of 0 and 1 indicating the group association of each variable (0 when it belongs to L and 1 when it belongs to U). We get the following model:

$$\begin{aligned} y|\beta_\gamma, \gamma &\sim \mathcal{N}_n(X_\gamma \beta_\gamma, V), \\ \beta_\gamma &\sim \mathcal{N}_{p_\gamma}(\mu_\gamma, W_\gamma), \\ V &\sim \mathcal{F}_V, \\ W_\gamma &\sim \mathcal{F}_{W_\gamma}, \\ \gamma &\sim \mathcal{U}_{\{0,1\}^p}, \end{aligned} \quad .$$

where $p_\gamma = \sum_{j=1}^{p} \gamma_j$. The prior distributions $\mathcal{F}_V$ and $\mathcal{F}_W$ will be specified in the sequel. As no previous information about the variables x_i is assumed, we use a uniform prior on γ. We note that a beta prior could be marginally used for each γ_i (we can generate a value and set γ_i to 1 if the value is greater than 0.5, otherwise 0), or more generally a Dirichlet prior tying them all together. In this context, as in most Bayesian selection models, the selection variable interacts with the other variables and parameters in an implicit way. Let's see how to derive a good posterior for γ.

3.3.1 Towards deriving the posterior of γ

We are interested in $p(\gamma|y, \cdots)$, but we will just use $p(\gamma|D)$ or $p(\gamma|y)$ for simplicity. We know that:

$$p(\gamma \mid y) = \frac{p(y \mid \gamma)p(\gamma)}{p(y)} = \frac{p(y \mid \gamma)p(\gamma)}{\sum_{\gamma'} p(y \mid \gamma')p(\gamma')}, \tag{3.2}$$

where $\gamma' \in \{0, 1\}^p$. Let $\gamma_{-j} = (\gamma_1, \ldots, \gamma_{j-1}, \gamma_{j+1}, \ldots, \gamma_p)$. Hence γ_{-j} is the vector representing all gamma values except γ_j. Since it will be difficult or almost impossible to enumerate all

possible γ values (for large p), we will instead be focusing on the individual γ_j which has only two possible states:

$$p(\gamma_j \mid y, \gamma_{-j}) = \frac{p(\gamma_j, \gamma_{-j} \mid y)}{p(\gamma_{-j} \mid y)} = \frac{p(\gamma \mid y)}{p(\gamma_{-j} \mid y)}.$$

Let $a, b \in \{0, 1\}$ such that $a + b = 1$. Suppose γ_j is equal to state a and can potentially jump to its opposite state b. we define the odds r_{ab} as:

$$\begin{aligned} r_{ab}(j) &= \frac{p(\gamma_j = a \mid y, \gamma_{-j})}{p(\gamma_j = b \mid y, \gamma_{-j})} = \frac{p(\gamma_j = a, \gamma_{-j} \mid y)}{p(\gamma_{-j} \mid y)} \frac{p(\gamma_{-j} \mid y)}{p(\gamma_j = b, \gamma_{-j} \mid y)} \\ &= \frac{p(\gamma_j = a, \gamma_{-j} \mid y)}{p(\gamma_j = b, \gamma_{-j} \mid y)} = \frac{p(y|\gamma_j = a, \gamma_{-j})}{p(y|\gamma_j = b, \gamma_{-j})} \frac{p(\gamma_j = a, \gamma_{-j})}{p(\gamma_j = b, \gamma_{-j})}. \end{aligned}$$

So we see that the quantity r_{ab} is obtained by the Bayes factor of the two hypothesis multiplied by a likelihood ratio.

3.3.2 Derivation of the regression parameters β

Using the classic Bayesian computations we obtain the following posterior:

$$\beta_\gamma \mid \gamma, y, V, W_\gamma \sim N_{p_\gamma}(\mu^*_\gamma, W^*_\gamma), \tag{3.3}$$

where

$$\begin{aligned} W^*_\gamma &= (W^{-1}_\gamma + X'_\gamma V^{-1} X_\gamma)^{-1}, \\ \mu^*_\gamma &= W^*(W^{-1}_\gamma \mu_\gamma + X'_\gamma V^{-1} y). \end{aligned}$$

Now let's show the steps that led to this posterior. Let's consider

$$\begin{aligned} p(\beta_\gamma \mid \gamma, y, V, W_\gamma) &\propto p(y \mid \beta_\gamma, V, W_\gamma, \gamma)\ p(\beta_\gamma \mid W_\gamma, \gamma) \\ &\propto (2\pi)^{-\frac{n}{2}} |V|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(y - X_\gamma\beta_\gamma)' V^{-1}(y - X_\gamma\beta_\gamma)\right\} \\ &\times (2\pi)^{-\frac{p_\gamma}{2}} |W_\gamma|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\beta_\gamma - \mu_\gamma)' W^{-1}_\gamma(\beta_\gamma - \mu_\gamma)\right\} \\ &\propto (2\pi)^{-\frac{p_\gamma}{2}} \exp\left\{-\frac{1}{2}[(y - X_\gamma\beta_\gamma)' V^{-1}(y - X_\gamma\beta_\gamma) + (\beta_\gamma - \mu_\gamma)' W^{-1}_\gamma(\beta_\gamma - \mu_\gamma)]\right\} \\ &\propto (2\pi)^{-\frac{p_\gamma}{2}} \exp\Big\{-\frac{1}{2}[\beta'_\gamma(X'_\gamma V^{-1} X_\gamma + W^{-1}_\gamma)\beta_\gamma - 2\beta'_\gamma(X'_\gamma V^{-1} y + W^{-1}_\gamma \mu_\gamma) \\ &+ y' V^{-1} y + \mu'_\gamma W^{-1}_\gamma \mu_\gamma]\Big\} \\ &\propto (2\pi)^{-\frac{p_\gamma}{2}} \exp\Big\{-\frac{1}{2}[\beta'_\gamma W^{*-1}_\gamma \beta_\gamma - 2\beta'_\gamma W^{*-1}_\gamma \mu^*_\gamma + y' V^{-1} y + \mu'_\gamma W^{-1}_\gamma \mu_\gamma]\Big\} \\ &\propto (2\pi)^{-\frac{p_\gamma}{2}} \exp\Big\{-\frac{1}{2}[(\beta_\gamma - \mu^*_\gamma)' W^{*-1}_\gamma(\beta_\gamma - \mu^*_\gamma) - \mu^{*'}_\gamma W^{*-1}_\gamma \mu^*_\gamma + y' V^{-1} y + \mu'_\gamma W^{-1}_\gamma \mu_\gamma]\Big\}. \end{aligned}$$

In order to compute r_{ab}, we need to find $p(y|\gamma)$ by integrating with respect to everything else such that:

$$p(y \mid \gamma) = \int\int\int p(y \mid \beta_\gamma, V, W_\gamma, \gamma)\, p(\beta_\gamma \mid W_\gamma, \gamma)\, p(V, W_\gamma)\, \mathrm{d}\beta_\gamma \mathrm{d}V \mathrm{d}W_\gamma,$$

assuming of course that $p(V, W_\gamma) = p(V, W_\gamma \mid \gamma)$. This integral is difficult if not impossible to evaluate. So we will try to integrate only with respect to β_γ as follows:

$$\begin{aligned}
p(y \mid V, W_\gamma, \gamma) &= \int_{\mathbb{R}^{p_\gamma}} p(y \mid \beta_\gamma, V, W_\gamma, \gamma)\, p(\beta_\gamma \mid W_\gamma, \gamma) \mathrm{d}\beta_\gamma \\
&= \int_{\mathbb{R}^{p_\gamma}} (2\pi)^{-\frac{(p_\gamma+n)}{2}} |V|^{-\frac{1}{2}} |W_\gamma|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} [(\beta_\gamma - \mu_\gamma^*)' W_\gamma^{*-1} (\beta_\gamma - \mu_\gamma^*) \right. \\
&\left. - \mu_\gamma'^* W_\gamma^{*-1} \mu_\gamma^* + y' V^{-1} y + \mu_\gamma' W_\gamma^{-1} \mu_\gamma] \right\} \mathrm{d}\beta_\gamma \\
&= (2\pi)^{-\frac{(n)}{2}} |V|^{-\frac{1}{2}} |W_\gamma|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} [- \mu_\gamma'^* W_\gamma^{*-1} \mu_\gamma^* + y' V - 1 y + \mu_\gamma' W_\gamma^{-1} \mu_\gamma] \right\} \\
&\times \int_{\mathbb{R}^{p_\gamma}} (2\pi)^{-\frac{p_\gamma}{2}} \exp\left\{ -\frac{1}{2} [(\beta_\gamma - \mu_\gamma^*)' W_\gamma^{*-1} (\beta_\gamma - \mu_\gamma^*)] \right\} \mathrm{d}\beta_\gamma \\
&= (2\pi)^{-\frac{n}{2}} |V|^{-\frac{1}{2}} |W_\gamma|^{-\frac{1}{2}} |W_\gamma^*|^{\frac{1}{2}} \exp\left\{ -\frac{1}{2} [- \mu_\gamma'^* W_\gamma^{*-1} \mu_\gamma^* + y' V^{-1} y + \mu_\gamma' W_\gamma^{-1} \mu_\gamma] \right\}.
\end{aligned}$$

3.4 Narrowing down the Bayesian model selection

We have defined a general framework and would now like to develop a specific Bayesian selection which can be coupled with a few selection techniques. So we specify the following variables:

$$W_\gamma = \left(\lambda \frac{X_\gamma' X_\gamma}{c} + \lambda I_{p_\gamma} \right)^{-1}, \quad V = \sigma^2 I_n, \text{ with } \lambda, \sigma^2, c > 0.$$

The matrix I_p is to avoid ending up with a singular matrix, c controls the relative importance of the variance and covariance estimates $X_\gamma' X_\gamma$ against the identity matrix (a higher c value is a sign of bigger uncertainty), and λ controls the overall strength of the prior values of β_γ (whose covariance matrix is W_γ). So the refined model becomes:

$$\begin{aligned}
y | \beta_\gamma, \gamma, \sigma^2, \lambda, c &\sim \mathcal{N}_n(X_\gamma \beta_\gamma, \sigma^2 I_n), \\
\beta_\gamma &\sim \mathcal{N}_{p_\gamma}\left(\mu_\gamma, \left(\lambda \frac{X_\gamma' X_\gamma}{c} + \lambda I_{p_\gamma} \right)^{-1} \right), \\
\sigma^2 &\sim \sigma^{-2}, \\
\lambda &\sim \lambda^{-1}, \\
c &\sim c^{-1}, \\
\gamma &\sim \mathcal{U}_{\{0,1\}^p}.
\end{aligned}$$

We will refer to this model as BVE (as in Bayesian variable elimination). Since we assume no prior information about c, λ and σ^2 except that they are positive, we will attach non-informative priors to them. We will use some Jeffrey (defined in the next paragraph) prior for these two parameters.

Motivation for Jeffreys prior They are a class of noninformative priors which have the additional quality of being invariant with respect to re-parameterization. The key insight is

that if $q(\phi)$ is uninformative, then any transformation $\theta = f(\phi)$ should also be uninformative. By change of variables rules, we have that $p(\theta)|d\theta| = q(\phi)|d\phi|$. Let $q(\phi) \propto \mathbb{I}(\phi)^{1/2}$ where $\mathbb{I}$ is the Fisher information. So we get:

$$q(\phi)^2 = \mathbb{I}(\phi) = \mathbb{E}\left\{\left(\frac{d\ \log(p(y|\phi))}{d\phi}\right)^2\right\} = \mathbb{E}\left\{\left(\frac{d\ \log(p(y|\theta))}{d\theta}\frac{d\theta}{d\phi}\right)^2\right\} = \mathbb{I}(\theta)\left(\frac{d\theta}{d\phi}\right)^2.$$

So it follows that $p(\theta) = \mathbb{I}(\theta)^{1/2}$ and p remains the same as q after transformation. Now let's derive the appropriate priors for λ and σ^2.

3.4.1 Noninformative prior derivation

Since both variables are tied in similar ways to data modelled as normally distributed, we will focus on a generic example. Consider:

$$y|\tau \sim \mathcal{N}_n(\mu, \tau M),$$

where $\tau > 0$ and M is a positive definite matrix. To derive an appropriate prior for τ, we perform the following computations. By taking the logarithm, we have

$$\begin{aligned}\log\Big(p(y|\tau)\Big) &= -\frac{1}{2}\log(|\tau M|) - \frac{1}{2}(y-\mu)'(\tau M)^{-1}(y-\mu) + \ \cdots \\ &= -\frac{n}{2}\log(\tau) - \frac{1}{2\tau}(y-\mu)'(M)^{-1}(y-\mu) + \ \cdots\end{aligned}$$

By taking the derivative, we have

$$\frac{d}{d\tau}\log\Big(p(y|\tau)\Big) = -\frac{n}{2\tau} + \frac{1}{2\tau^2}(y-\mu)'(M)^{-1}(y-\mu).$$

The Fisher information can be calculated by

$$\begin{aligned}\mathbb{I}(\tau) = \mathbb{E}\left(-\frac{d^2}{d\tau^2}\log\Big(p(y|\tau)\Big)\right) &= \mathbb{E}\left(-\frac{n}{2\tau^2} + \frac{1}{\tau^3}(y-\mu)'(M)^{-1}(y-\mu)\right) \\ &= -\frac{n}{2\tau^2} + \frac{1}{\tau^2}\mathbb{E}\left((y-\mu)'(\tau M)^{-1}(y-\mu)\right) \\ &= -\frac{n}{2\tau^2} + \frac{n}{\tau^2} = \frac{n}{2\tau^2}.\end{aligned}$$

So we arrive to the result:

$$p(\tau) \propto \mathbb{I}(\tau)^{1/2} \propto \frac{\sqrt{n}}{\tau}.$$

We note that this also corresponds to $IG(0,0)$, an inverse gamma with null parameters. It is an improper prior, however as long as the posterior is proper, there will be no issues.

3.4.2 Computing the posteriors of σ^2 and λ

The posterior distributions for the ridge parameter λ and the variance σ^2 are summarised in the following statement.

Theorem 3. *By considering the noninformative priors* $p(\lambda) \propto \frac{1}{\lambda}$ *and* $p(\sigma^2) \propto \frac{1}{\sigma^2}$*, we have the posteriors:*

$$\sigma^2 \mid y, \beta_\gamma, \lambda, \gamma, c \sim IG\left(\frac{n}{2}\ ,\ \frac{\|y - X_\gamma\beta_\gamma\|^2}{2}\right)$$
$$\lambda \mid y, \beta_\gamma, \sigma^2, \gamma, c \sim \Gamma\left(\frac{p_\gamma}{2}\ ,\ \frac{(\beta_\gamma - \mu_\gamma)'(c^{-1}(X'_\gamma X_\gamma) + I_p)(\beta_\gamma - \mu_\gamma)}{2}\right).$$

Proof. Direct calculation enables us to write

$$\begin{aligned}
p(\sigma^2, \lambda, \beta_\gamma, \gamma, c \mid y) &\propto p(y \mid \beta_\gamma, \sigma^2, \gamma, \lambda, c)p(\beta_\gamma, \sigma^2, \gamma, \lambda, c) \\
&= p(y \mid \beta_\gamma, \sigma^2)p(\beta_\gamma \mid \sigma^2, \lambda, c)p(\sigma^2, \lambda, c) \\
&\propto (\sigma^2)^{-\frac{n}{2}} \exp\left\{-\frac{\|y - X_\gamma\beta_\gamma\|^2}{2\sigma^2}\right\}(\sigma^2)^{-1} \times \lambda^{-1} \\
&\times \left|(c^{-1}\lambda(X'_\gamma X_\gamma) + \lambda I_{p_\gamma})^{-1}\right|^{-1/2} \exp\left\{-\frac{1}{2}(\beta_\gamma - \mu_\gamma)'(c^{-1}\lambda(X'_\gamma X_\gamma) + \lambda I_{p_\gamma})(\beta_\gamma - \mu_\gamma)\right\} \\
&\propto (\sigma^2)^{-(\frac{n}{2}+1)} \exp\left\{-\frac{1}{\sigma^2}\frac{\|y - X_\gamma\beta_\gamma\|^2}{2}\right\} \\
&\times \lambda^{\frac{p_\gamma}{2}-1} \exp\left\{-\lambda\frac{(\beta_\gamma - \mu_\gamma)'(c^{-1}(X'_\gamma X_\gamma) + I_{p_\gamma})(\beta_\gamma - \mu_\gamma)}{2}\right\}.
\end{aligned}$$

□

As for c, we have not been able to derive a recognizable full conditional posterior distribution. So we simulate using the Metropolis Hastings algorithm with a gamma distribution as a proposal. A further refinement of the model consists of assuming all prior models to be equally probable. So r_{ab} reduces to the Bayes factor. Since the latter depends on $p(y|\gamma)$, which we were not able to derive, we approximate it by:

$$r_{ab} \approx \frac{p(y \mid \gamma_j = a, \gamma_{-j}, \sigma^2, \lambda, c)}{p(y \mid \gamma_j = b, \gamma_{-j}, \sigma^2, \lambda, c)}. \tag{3.4}$$

To get a good estimate of r_{ab}, we may choose to run an inner loop over the values of γ given λ and σ^2 before getting back to generating the latter. The number of iterations for this inner loop is labeled N_γ.

3.4.3 Effects of the parameters c, λ, and σ^2

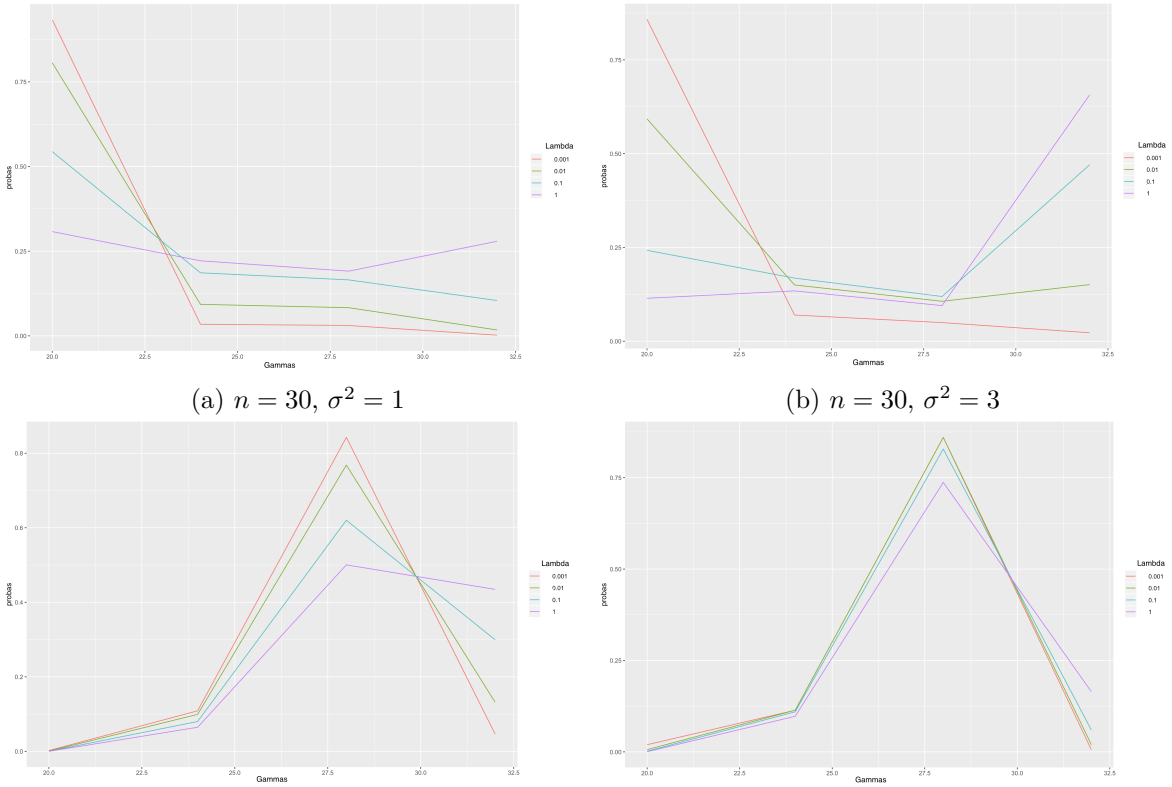
The values of these parameters have very important repercussions on the form of the posterior. To illustrate that, we will explore their effects by studying a dataset containing 5 variables

(all identically and independently generated) among which only 3 are relevant (with different coefficients). We simulate for different combination of n and σ^2. We will consider different values for each parameter. In practice though, they will be generated from their conditional posteriors. As for λ, its initial value λ_0 will actually influence the converged distribution no matter how long the chain is (so the initial value is a hyper-parameter that can be tuned). Although this is an undesirable effect, the generation of different λ will impact the space exploration and therefore still makes the estimation of other parameters more robust.

For all the plots in this section, the x-axis Gammas represent a mapping of the values of γ to an index value:

$$g(\gamma) = \sum_{i=1}^{p} 2^i \gamma_i,$$

where g is a function on $\{0,1\}^p$. Let's note that we removed values with probability close to 0 (less than 10^{-3}) to make the plots look better and clearer.



(a) $n = 30$, $\sigma^2 = 1$

(b) $n = 30$, $\sigma^2 = 3$

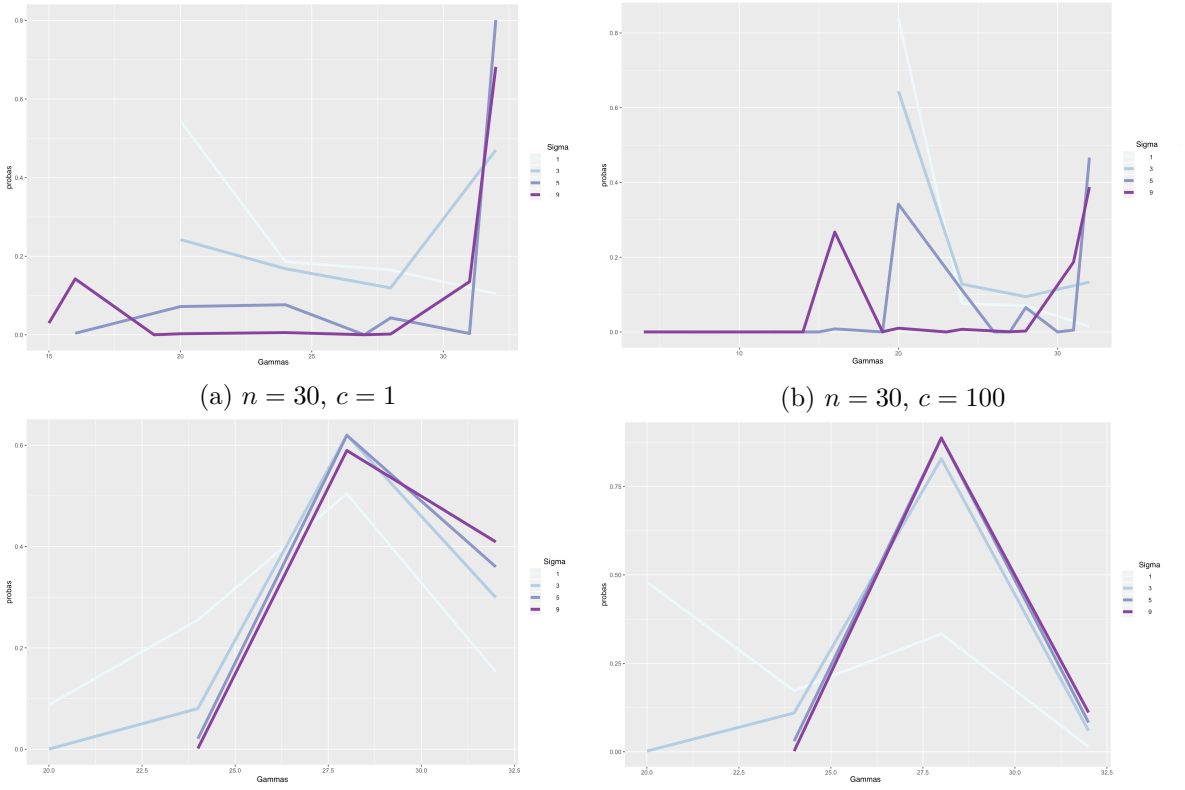
(c) $n = 300$, $\sigma^2 = 3$, $c = 1$

(d) $n = 300$, $\sigma^2 = 3$, $c = 100$

Figure 3.1: Distribution of γ over a custom topological order for different λ values

Figure 3.1 above represents the true distribution $p(y|\gamma, \lambda, \sigma^2, c)$ computed for all the possible values of γ. Since λ is a fixed value here, we considered 4 potentials values that we plotted within different conditions. The first two plots highlight the variations when there is more

noise in the dataset. We can see that smaller values of λ produce more concentrated distributions, which means that more probability weight is put on some few points as opposed to being spread out when λ is not small. It makes sense in that it controls the strength of the prior over the likelihood, which tends to be highly concentrated on a particular mode. However, with more noise on the background, the posterior tends toward the full gamma value (all ones) since there is more risk of estimation precision loss when good variables are removed than when useless ones are introduced. One obvious reason is that there is always a possibility to eliminate them during the further inference steps by setting their corresponding β values to 0. The bottom two plots show what happens when n and c change. Naturally, a high number of observations will tend to push the posterior structure toward that of the likelihood, while a high c value clearly limits the variability caused by the change in λ values. So the higher c is, the less important λ is or the prior as a matter of fact.



(a) $n = 30$, $c = 1$

(b) $n = 30$, $c = 100$

(c) $n = 300$, $c = 1$

(d) $n = 300$, $c = 100$

Figure 3.2: Plot of the distribution of γ over a custom topological order for different σ^2 values

The Figure 3.2 compares different values of σ^2. All remarks about n and c above also apply here. We notice that the higher the noise, the wigglier the posterior becomes. Although the posterior mode on all these plots don't always point the true value of $g(\gamma) = 28$, the marginal inclusion probabilities of each variable are well defined (see Figure 3.3).

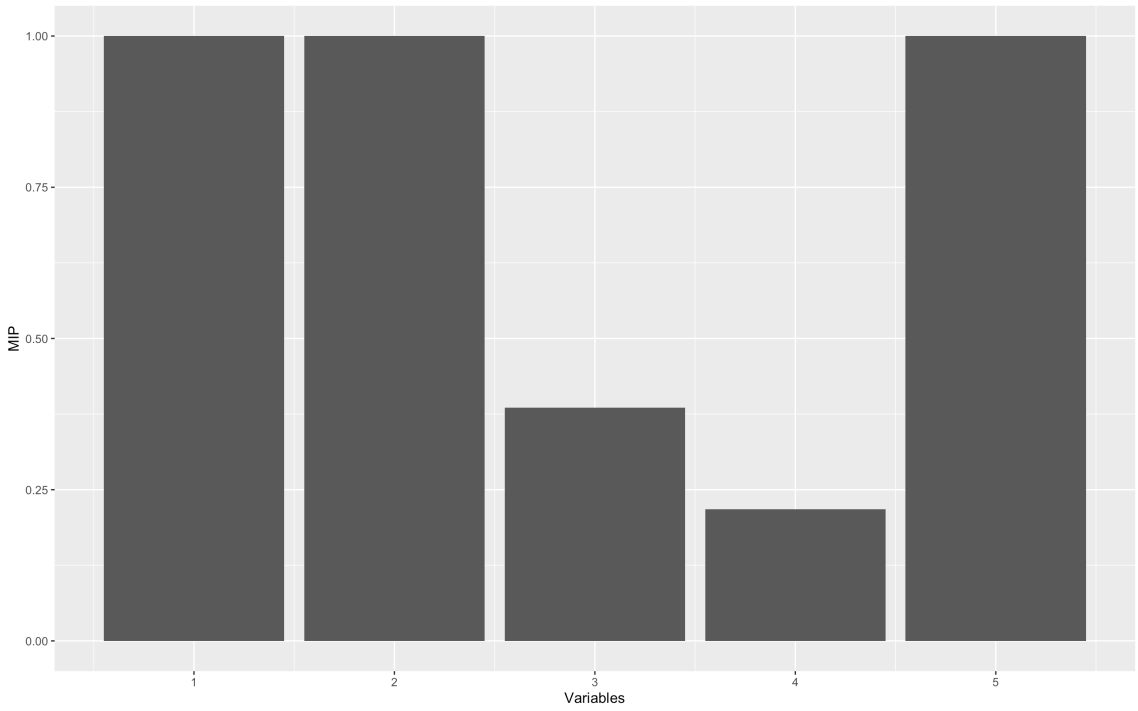


Figure 3.3: Marginal inclusion probabilities (MIP) of each variable.

3.5 Replacing λ with a randomly scaling Wishart matrix

We propose now a model formulation that works for data where we have less regressors than observations ($p < n$). In basic regression, the way the regressors are weighted does not matter because the betas parameters adjust accordingly. However, with the presence of regularization terms, the outcome of the inference itself is affected. This is the reason why the adaptive Lasso produces results different from the regular Lasso (Kundu and Dunson, 2014).

Definition 11 (*Wishart Distribution*)**.** *Let $M \in \mathbb{R}^{p\times p}$ be a random variable that is positive definite, $V \in \mathbb{R}^{p\times p}$ be a fixed positive definite matrix and $k \in \mathbb{N}$. The Wishart distribution $\mathcal{W}_p(V, k)$ has a density with respect to the Lebesgue measure given by:*

$$p(M) = \frac{|M|^{(k-p-1)/2}}{2^{kp/2}|V|^{k/2}\Gamma_p(\frac{k}{2})} \exp\left\{-\frac{1}{2}\mathrm{Tr}(V^{-1}M)\right\},$$

where Γ_p is the multivariate gamma function and Tr *is the trace function. Its mean and its variance are:*

$$\mathbb{E}(M) = kV,$$
$$\mathbb{V}(M_{ij}) = k(v_{ij}^2 + v_{ii}v_{jj}).$$

Definition 12 (*Inverse Wishart Distribution*)**.** *Let $M \in \mathbb{R}^{p\times p}$ be a random variable that is positive definite, $U \in \mathbb{R}^{p\times p}$ be a fixed positive definite matrix and $k \in \mathbb{N}$. The inverse Wishart distribution $\mathcal{W}_p^{-1}(U,k)$ has a density with respect to the Lebesgue measure given by:*

$$p(M) = \frac{|M|^{-(k-p-1)/2}}{2^{kp/2}|U|^{-k/2}\Gamma_p(\frac{k}{2})} \exp\left\{-\frac{1}{2}\mathrm{Tr}(UM^{-1})\right\}.$$

Its mean is given by:

$$\mathbb{E}(M) = \frac{U}{k-p-1}.$$

Property 1. *If $M \sim \mathcal{W}_p(V,k)$, and $R = M^{-1}$, then $R \sim \mathcal{W}_p^{-1}(V^{-1},k)$.*

We will use these definitions and properties to develop our general g-prior model next.

3.5.1 Setting up the model hierarchy

We proceed as follows:

$$y = XW^{-1}\tilde{\beta} + \epsilon,$$

where W is a positive definite matrix. The variables in the matrix X are first scaled before being introduced in the regression problem. Then we attach a prior to the coefficients $\tilde{\beta}$ which gives the following:

$$\begin{aligned}\tilde{\beta} &\sim \mathcal{N}_p(\mu_{\tilde{\beta}}, \sigma^2[X'X]^{-1}),\\ \beta := W^{-1}\tilde{\beta} &\sim \mathcal{N}_p(W^{-1}\mu_{\tilde{\beta}}, \sigma^2 W^{-1}[X'X]^{-1}W^{-1}).\end{aligned}$$

So we end up with a formulation similar to what we had before but with a different prior. Now we add the other priors into a new hierarchical manner:

$$y|\beta_\gamma, \gamma, \sigma^2, W_\gamma \sim \mathcal{N}_n(X_\gamma\beta_\gamma, \sigma^2 I_n), \tag{3.5}$$

$$\beta_\gamma \sim \mathcal{N}_{p_\gamma}(\mu_\gamma, \sigma^2[W_\gamma \mathrm{Kr}(X_\gamma)W_\gamma]^{-1}), \tag{3.6}$$

$$W_\gamma^{-2} \sim \mathcal{W}_{p_\gamma}(Z_\gamma, d), \tag{3.7}$$

$$\sigma^2 \sim \frac{p_\gamma^{1/2}}{\sigma^{-2}}, \tag{3.8}$$

$$\gamma \sim \mathcal{U}_{\{0,1\}^p}, \tag{3.9}$$

where $d \in \mathbb{N}$ and $Z_\gamma \in \mathbb{R}^{p_\gamma \times p_\gamma}$. We will refer to this model as WaLasso (as in weighted adaptive Lasso). Note that $W_\gamma' = W_\gamma$, Z_γ is a symmetric diagonal matrix by design and Kr is a kernel function. Let's define $\Psi_\gamma = W_\gamma \mathrm{Kr}(X_\gamma)W_\gamma$.

Doing similar computations as above, we get these full posterior distributions:

$$\beta_\gamma|y, \sigma^2, \gamma, W_\gamma \sim \mathcal{N}_{p_\gamma}(\mu_\gamma^*, C_\gamma^*),$$

where

$$C_\gamma^* = \left[\frac{1}{\sigma^2}\Psi_\gamma + \frac{1}{\sigma^2}X_\gamma' X_\gamma\right]^{-1},$$
$$\mu_\gamma^* = C_\gamma^*\left[\frac{1}{\sigma^2}\Psi_\gamma\mu_\gamma + \frac{1}{\sigma^2}X_\gamma' y\right] = \left[\Psi_\gamma + X_\gamma' X_\gamma\right]^{-1}\left[\Psi_\gamma\mu_\gamma + X_\gamma' y\right].$$

We just filled up the general expression derived on the section of beta related computations. The other posteriors are

$$W_\gamma^{-2}|y,\beta_\gamma,\gamma,\sigma^2 \sim \mathcal{W}_{p_\gamma}^{-1}\left(\frac{1}{\sigma^2}(\beta_\gamma-\mu_\gamma)(\beta_\gamma-\mu_\gamma)'\mathrm{Kr}(X_\gamma) + Z_\gamma^{-1},\ d+1\right),$$
$$\sigma^2|y,\beta_\gamma,\gamma,W_\gamma \sim \mathcal{IG}\left(\frac{n}{2},\ \frac{1}{2}[y'y + \mu_\gamma'\Psi_\gamma\mu_\gamma - \left[\Psi_\gamma\mu_\gamma + X_\gamma' y\right]'\left[\Psi_\gamma + X_\gamma' X_\gamma\right]^{-1}\left[\Psi_\gamma\mu_\gamma + X_\gamma' y\right]\right).$$

We will layout the proof for the derivation of the inverse Wishart posterior of W_γ^{-2}. Let's start by writing the full joint distribution that the hierarchical decomposition above yields:

$$\begin{aligned} p(y,\beta_\gamma,\gamma,\sigma^2,W_\gamma) &= p(\beta_\gamma,\sigma^2,W_\gamma)p(y|\beta_\gamma,\sigma^2,W_\gamma) \\ &\propto (\sigma^2)^{-n/2}\exp\left(-\frac{\|y-X_\gamma\beta_\gamma\|^2}{2\sigma^2}\right)\frac{p_\gamma^{1/2}}{\sigma^2}|W_\gamma^2|^{\frac{d-p_\gamma-1}{2}}\exp\left(-\frac{1}{2}\mathrm{Tr}(Z_\gamma^{-1}W_\gamma^2)\right) \\ &\times |\sigma^2[W_\gamma\mathrm{Kr}(X_\gamma)W_\gamma]^{-1}|^{-\frac{1}{2}}\exp\left(-\frac{(\beta_\gamma-\mu_\gamma)'W_\gamma\mathrm{Kr}(X_\gamma)W_\gamma(\beta_\gamma-\mu_\gamma)}{2\sigma^2}\right). \end{aligned}$$

The marginal of σ^2 can be read from the expression above. Let's isolate the terms needed to derive the Wishart posterior. We deduce that:

$$\begin{aligned} p(W_\gamma^{-2}|\ y,\beta_\gamma,\sigma^2,\gamma) &\propto |W_\gamma^2|^{\frac{d-p_\gamma-1}{2}}\exp\left(-\frac{1}{2}\mathrm{Tr}(Z_\gamma^{-1}W_\gamma^2)\right) \\ &\times |[W_\gamma\mathrm{Kr}(X_\gamma)W_\gamma]^{-1}|^{-\frac{1}{2}}\exp\left(-\frac{(\beta_\gamma-\mu_\gamma)'W_\gamma\mathrm{Kr}(X_\gamma)W_\gamma(\beta_\gamma-\mu_\gamma)}{2\sigma^2}\right). \end{aligned}$$

We use the following properties from Linear algebra to perform the right calculations:

Property 2. *Let $A \in \mathbb{R}^{p\times p}$ and $B \in \mathbb{R}^{p\times p}$ be two square matrices. Let $a \in \mathbb{R}^p$ and $b \in \mathbb{R}^p$ be two vectors of reals. Then, we have*

$$\begin{aligned} |AB| &= |BA| = |A||B|, \\ |A^{-1}| &= |A|^{-1}, \\ \mathrm{Tr}(A+B) &= \mathrm{Tr}(A) + \mathrm{Tr}(B), \\ \mathrm{Tr}(ba') &= a'b. \end{aligned}$$

So we get:

$$
\begin{aligned}
p(W_\gamma^{-2}|\ y, \beta_\gamma, \sigma^2, \gamma) &\propto |W_\gamma^2|^{\frac{d-p_\gamma-1}{2}} \exp\left(-\frac{1}{2}\mathrm{Tr}(Z_\gamma^{-1}W_\gamma^2)\right) \\
&\quad \times |\mathrm{Kr}(X_\gamma)|^{\frac{1}{2}} W_\gamma W_\gamma|^{\frac{1}{2}} \exp\left(-\frac{(\beta_\gamma-\mu_\gamma)'W_\gamma \mathrm{Kr}(X_\gamma) W_\gamma(\beta_\gamma-\mu_\gamma)}{2\sigma^2}\right) \\
&= |W_\gamma^2|^{\frac{d-p_\gamma}{2}} \exp\left(-\frac{1}{2}\mathrm{Tr}(Z_\gamma^{-1}W_\gamma^2)\right) \exp\left(-\mathrm{Tr}\left\{\frac{(\beta_\gamma-\mu_\gamma)(\beta_\gamma-\mu_\gamma)'W_\gamma \mathrm{Kr}(X_\gamma) W_\gamma}{2\sigma^2}\right\}\right) \\
&= |W_\gamma^2|^{\frac{d-p_\gamma}{2}} \exp\left(-\frac{1}{2}\mathrm{Tr}(Z_\gamma^{-1}W_\gamma^2) - \mathrm{Tr}\left\{\frac{(\beta_\gamma-\mu_\gamma)(\beta_\gamma-\mu_\gamma)'\mathrm{Kr}(X_\gamma) W_\gamma^2}{2\sigma^2}\right\}\right) \\
&= |W_\gamma^2|^{\frac{d-p_\gamma}{2}} \exp\left(-\frac{1}{2}\mathrm{Tr}\left\{[Z_\gamma^{-1} + \frac{(\beta_\gamma-\mu_\gamma)(\beta_\gamma-\mu_\gamma)'\mathrm{Kr}(X_\gamma)}{\sigma^2}]W_\gamma^2\right\}\right).
\end{aligned}
$$

The posterior of either W_γ^{-2} or W_γ^2 can be read from the last expression.

Next, we will derive the marginal density $p(\sigma^2|y, W_\gamma, \gamma)$

$$p(\sigma^2|y, W_\gamma, \gamma) = p(\sigma^2)p(y|\sigma^2, W_\gamma),$$

where the quantity $p(y|\sigma^2, W_\gamma)$ is obtained by integrating β_γ as follows (we just fill up the results we already computed):

$$
\begin{aligned}
p(y|\sigma^2, W_\gamma, \gamma) &= \int p(y|\beta_\gamma, \sigma^2, W_\gamma) p(\beta_\gamma|\sigma^2, W_\gamma) d\beta_\gamma \\
&= (2\pi)^{-\frac{n}{2}} (\sigma^2)^{-\frac{n}{2}} |\sigma^2[\Psi_\gamma]^{-1}|^{-\frac{1}{2}} \left|\frac{1}{\sigma^2}\Psi_\gamma + \frac{1}{\sigma^2}X_\gamma' X_\gamma\right|^{-\frac{1}{2}} \\
&\times \exp\left\{-\frac{1}{2}\left(\frac{1}{\sigma^2}y'y + \frac{1}{\sigma^2}\mu_\gamma'\Psi_\gamma\mu_\gamma\right)\right. \\
&\times \left.\exp\left\{-\frac{1}{2}\left(-\left[\frac{1}{\sigma^2}\Psi_\gamma\mu_\gamma + \frac{1}{\sigma^2}X_\gamma' y\right]' \left[\frac{1}{\sigma^2}\Psi_\gamma + \frac{1}{\sigma^2}X_\gamma' X_\gamma\right]^{-1} \left[\frac{1}{\sigma^2}\Psi_\gamma\mu_\gamma + \frac{1}{\sigma^2}X_\gamma' y\right]\right)\right\} \right. \\
&= (2\pi)^{-\frac{n}{2}} (\sigma^2)^{-\frac{n}{2}} |\Psi_\gamma|^{\frac{1}{2}} \left|\Psi_\gamma + X_\gamma' X_\gamma\right|^{-\frac{1}{2}} \\
&\times \exp\left\{-\frac{1}{2\sigma^2}\left(y'y + \mu_\gamma'\Psi_\gamma\mu_\gamma - \left[\Psi_\gamma\mu_\gamma + X_\gamma' y\right]' \left[\Psi_\gamma + X_\gamma' X_\gamma\right]^{-1} \left[\Psi_\gamma\mu_\gamma + X_\gamma' y\right]\right)\right\}.
\end{aligned}
$$

Let's define some new terms to simplify the upcoming equations. Let $\mu_\gamma^- = \Psi_\gamma\mu_\gamma + X_\gamma' y$ and $C_\gamma^- = \left[\Psi_\gamma + X_\gamma' X_\gamma\right]^{-1}$. Let's proceed by:

$$
\begin{aligned}
p(\sigma^2|y, W_\gamma, \gamma) &\propto \frac{p_\gamma^{1/2}}{\sigma^{-2}} (\sigma^2)^{-\frac{n}{2}} \exp\left\{-\frac{1}{2\sigma^2}\left(y'y + \mu_\gamma'\Psi_\gamma\mu_\gamma - \mu_\gamma^{-'} C_\gamma^- \mu_\gamma^-\right)\right\} \\
&\propto (\sigma^2)^{-\frac{n}{2}-1} \exp\left\{-\frac{1}{2\sigma^2}\left(y'y + \mu_\gamma'\Psi_\gamma\mu_\gamma - \mu_\gamma^{-'} C_\gamma^- \mu_\gamma^-\right)\right\}
\end{aligned}
$$

from the expression of $p(y|\sigma^2, W_\gamma, \gamma)$ that we can integrate with respect to σ^2.

Finally, we will compute the likelihood function upon which the exploration of γ is based. We get it by calculating $p(y|W_\gamma, \gamma)$ as follows:

$$\begin{aligned}
p(y|W_\gamma,\gamma) &= \int_{\mathbb{R}_+^*} p(y|\sigma^2, W_\gamma, \gamma)p(\sigma^2)d\sigma^2 \\
&= \int \frac{\sqrt{p_\gamma}}{\sigma^{-2}}(2\pi)^{-\frac{n}{2}}(\sigma^2)^{-\frac{n}{2}}|\Psi_\gamma|^{\frac{1}{2}}\left|C_\gamma^-\right|^{\frac{1}{2}}\exp\left\{-\frac{1}{2\sigma^2}\left(y'y+\mu_\gamma'\Psi_\gamma\mu_\gamma-\mu_\gamma^{-'}C_\gamma^-\mu_\gamma^-\right)\right\}d\sigma^2 \\
&= \sqrt{p_\gamma}(2\pi)^{-\frac{n}{2}}|\Psi_\gamma|^{\frac{1}{2}}\left|C_\gamma^-\right|^{\frac{1}{2}}\int(\sigma^2)^{-\frac{n}{2}-1}\exp\left\{-\frac{1}{2\sigma^2}\left(y'y+\mu_\gamma'\Psi_\gamma\mu_\gamma-\mu_\gamma^{-'}C_\gamma^-\mu_\gamma^-\right)\right\}d\sigma^2.
\end{aligned}$$

Let's put $\zeta_\gamma = \frac{1}{2}(y'y + \mu_\gamma'\Psi_\gamma\mu_\gamma - \mu_\gamma^{-'}C_\gamma^-\mu_\gamma^-)$. Hence, we can write

$$\begin{aligned}
p(y|W_\gamma,\gamma) &= \sqrt{p_\gamma}(2\pi)^{-\frac{n}{2}}|\Psi_\gamma|^{\frac{1}{2}}\left|C_\gamma^-\right|^{\frac{1}{2}}\int_{\mathbb{R}_+^*}(\sigma^2)^{-\frac{n}{2}-1}\exp\left\{-\frac{\zeta_\gamma}{\sigma^2}\right\}d\sigma^2 \\
&= \sqrt{p_\gamma}(2\pi)^{-\frac{n}{2}}|\Psi_\gamma|^{\frac{1}{2}}\left|C_\gamma^-\right|^{\frac{1}{2}}\Gamma(n/2)\zeta_\gamma^{-\frac{n}{2}}.
\end{aligned}$$

Let's summarize the posterior derivations using the new intermediary terms we defined along the way:

$$\begin{aligned}
\beta_\gamma|y, W_\gamma, \gamma, \sigma^2 &\sim \mathcal{N}_{p_\gamma}(\mu_\gamma^*, C_\gamma^*), \\
W_\gamma^{-2}|y, \beta_\gamma, \gamma, \sigma^2 &\sim \mathcal{W}_{p_\gamma}^{-1}\left(\frac{1}{\sigma^2}(\beta_\gamma-\mu_\gamma)(\beta_\gamma-\mu_\gamma)'\mathrm{Kr}(X_\gamma)+Z_\gamma^{-1},\ d+1\right), \\
\sigma^2|y, \beta_\gamma, \gamma, W_\gamma &\sim \mathcal{IG}\left(\frac{n}{2}, \zeta_\gamma\right),
\end{aligned}$$

where

$$\begin{aligned}
C_\gamma^* &= \sigma^2 C_\gamma^-, \\
\mu_\gamma^* &= C_\gamma^-\mu_\gamma^-, \\
\zeta_\gamma &= \frac{1}{2}(y'y + \mu_\gamma'\Psi_\gamma\mu_\gamma - \mu_\gamma^{-'}C_\gamma^-\mu_\gamma^-).
\end{aligned}$$

3.5.2 Derivation of Zellner's g-prior

The g-prior is an objective prior for the regression coefficients in Bayesian regression. It was introduced by Arnold Zellner (Zellner, 1986). This prior is a key tool in Bayes and empirical Bayes variable selection. Let us consider at first the following theorem.

Theorem 4. *Let us put $Z_\gamma = \frac{1}{gd}I_{p_\gamma}$, where I_{p_γ} is the identity matrix of size p_γ, and $g, d > 0$. If $d \to \infty$, then the priors (3.6)-(3.7) have a Zellner's g-prior structure.*

Proof. Let's prove the above statement by showing that W_γ converges to an appropriate constant matrix. Straightforwardly we have

$$\begin{aligned}
\mathbb{E}[W_\gamma^{-2}|y, \beta_\gamma, \gamma, \sigma^2] &= \frac{1}{\sigma^2(d-p_\gamma)}(\beta_\gamma-\mu_\gamma)(\beta_\gamma-\mu_\gamma)'\mathrm{Kr}(X_\gamma)+\frac{1}{d-p_\gamma}Z_\gamma^{-1} \\
&= \frac{1}{\sigma^2(d-p_\gamma)}(\beta_\gamma-\mu_\gamma)(\beta_\gamma-\mu_\gamma)'\mathrm{Kr}(X_\gamma)+\frac{gd}{d-p_\gamma}I_{p_\gamma}.
\end{aligned}$$

When $d \to \infty$, then :

$$\mathbb{E}[W_\gamma^{-2}|y, \beta_\gamma, \gamma, \sigma^2] \to gI_{p_\gamma}.$$

For any inverse Wishart matrix $M \sim \mathcal{W}_p^{-1}(\boldsymbol{U}, d)$, its variance is defined as:

$$\mathbb{V}(m_{ij}) = \frac{(d-p+1)u_{ij}^2 + (d-p-1)u_{ii}u_{jj}}{(d-p)(d-p-1)^2(d-p-3)}.$$

It is clear that $\mathbb{V}(m_{ij}) \to 0$ as $d \to \infty$. So $\mathbb{V}(W_\gamma^{-2}|y, \beta_\gamma, \gamma, \sigma^2) \to 0$ and we can safely conclude that :

$$W_\gamma|y, \beta_\gamma, \gamma, \sigma^2 \to \frac{1}{\sqrt{g}} I_{p_\gamma}.$$

In our model specification, we have:

$$\beta_\gamma \sim \mathcal{N}_{p_\gamma}(\mu_\gamma, \sigma^2[W_\gamma \mathrm{Kr}(X_\gamma) W_\gamma]^{-1}).$$

If $W_\gamma = \frac{1}{\sqrt{g}} I_{p_\gamma}$ and $\mathrm{Kr}(X_\gamma) = X_\gamma' X_\gamma$, then it is easily seen that

$$\beta_\gamma \sim \mathcal{N}_p(\mu_\gamma, g\sigma^2[X_\gamma' X_\gamma]^{-1}).$$

This completes the proof. □

Remark 2. *As* $\beta_\gamma - \mu_\gamma \to 0$ *and assuming that* $W_0^2 = dZ_\gamma$*, we get:*

$$\mathbb{E}(W_\gamma^{-2}|y, \beta_\gamma, \gamma, \sigma^2) = \frac{Z_\gamma^{-1}}{d+1} = \frac{d}{d+1} W_0^{-2}.$$

3.5.3 Resolving the non-uniqueness of the squared root of a matrix with negative eigenvalues

Let $M_\gamma = \frac{1}{\sigma^2}(\beta_\gamma - \mu_\gamma)(\beta_\gamma - \mu_\gamma)' \mathrm{Kr}(X_\gamma) + Z_\gamma^{-1}$. Since it is symmetric, it can also be written as $M_\gamma = Q_\gamma \Lambda_\gamma Q_\gamma'$. We need to calculate $M_\gamma^{1/2}$ and we need something that is unique. So we will choose our prior hyperparameter Z_γ in a way that fits such a need. Let's recall two important properties:

- If A is a square matrix, x a vector and ξ a scalar such that $Ax = \xi x$, we get that $(A + sI)x = (\xi + s)x$, where s is a scalar.
- A positive definite matrix has all positive eigenvalues.

Let's use these two properties to refine our matrix M_γ. Let $\delta = \min\{diag(\Lambda_\gamma)\}$ and suppose that $\delta \leq 0$. Define

$$M_\gamma^* = \frac{1}{\sigma^2}(\beta_\gamma - \mu_\gamma)(\beta_\gamma - \mu_\gamma)' \mathrm{Kr}(X_\gamma) + Z_\gamma^{*-1},$$

such that $Z_\gamma^{*-1} = Z_\gamma^{-1} + (h - \delta)$, $h > 0$, where h is a hyperparameter (default $h = 10^{-4}$). Since we are generating W_γ^{-2}, it needs to be inverted in order to get W_γ. So to insure this

invertability, we added the constant $h > 0$ so that we are sure that all eigenvalues are strictly positive. So as long as we refine our Z_γ should negative eigenvalues be present, we can be sure that the refined M_γ^* will be positive definite. Note that the transformation above is only applied when $\delta \leq 0$.

3.6 Selecting the variables

After generating some MCMC samples, there are various techniques that are applied to make it more probable that we end up with something closer to the true distribution of the variables of interest. For instance, one may truncate the first generated values since the exploration most likely started from a random point in the vector space and will take some iterations before getting to the appropriate region. Another technique called "thinning" consists of selecting only every other k-th points in the chain in order to reduce the correlation between successive samples which are supposed to be independent in a perfect scenario. So we know that there is nothing perfect about generating samples from a random variable with unknown distribution. So when it comes to variable selection, it may be necessary to apply similar techniques in order to perform the best selection possible.

Let's remind ourselves of what variable selection consists of. The goal is to find:

$$U := \{X_{i_1}, \ldots, X_{i_k}\} \text{ and } L := \{X_{j_1}, \ldots, X_{j_l}\}.$$

3.6.1 Vanilla selection

The most widely used selection technique is very simple. We include all variables such that the marginal proportion given by (3.10) is greater than 0.5. This means that if we let:

$$p(X_i \in U) = \frac{\sum_{k=1}^{n} \mathbb{1}(\gamma_i^{(k)} = 1)}{n}, \tag{3.10}$$

where $\gamma_i^{(k)}$ is the value of γ_i at the k-th iteration. If

$$\frac{\sum_{k=1}^{n} \mathbb{1}(\gamma_i^{(k)} = 1)}{n} > \frac{1}{2},$$

then we obtain

$$p(X_i \in U) > p(X_i \in L).$$

So it translates into the probability of belonging in U superior to that of belonging to L. As simple as this heuristic is, it suffers from prespecifying a threshold that makes it rigid and is therefore not robust to minor perturbations along that line of demarcation.

Figure 3.4 is obtained from a run with a dataset containing 100 variables (all indentically and independently generated) among which only 10 are relevant (with the same coefficient).

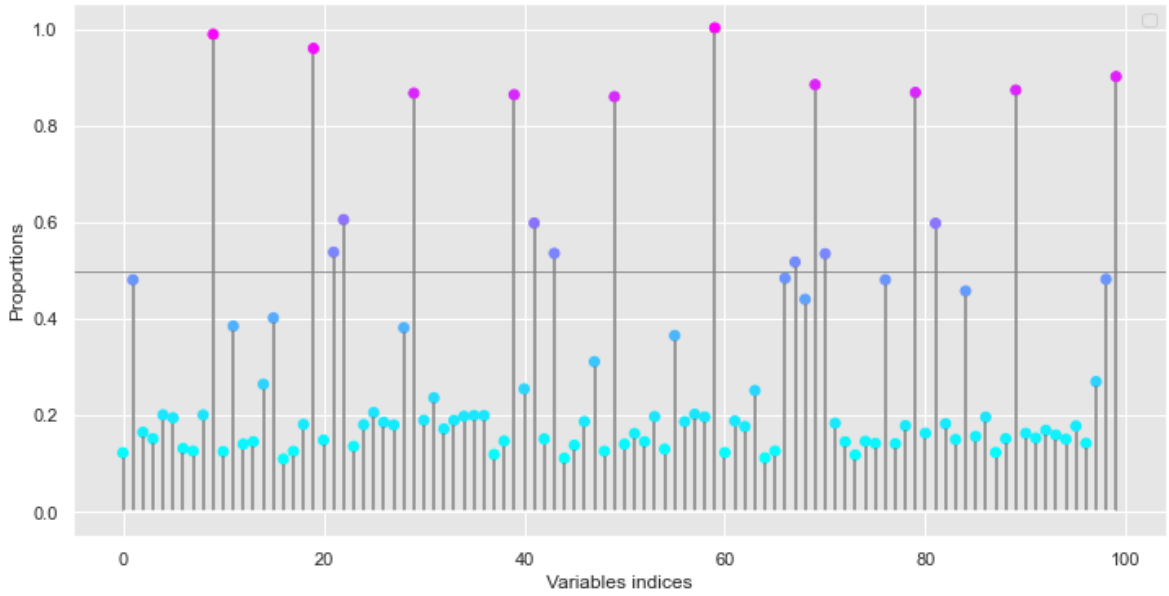


Figure 3.4: An example of posterior inclusion probabilities.

There are $n = 50$ data points and the standard deviation of the errors (Gaussian) is set to $\sigma = 3$. We see a clear separation and a big gap in marginal proportions between around 0.6 to 0.8. And yet, the vanilla method takes variables with proportions below that gap and assumes it to be part of the "good" variables.

So the vanilla selection leaves the heavy work to the Bayesian algorithm which is supposed to keep marginal proportions of all irrelevant variables below 0.5. Since the algorithm just explores the space, it is hard to achieve such result with consistency. We suggested that the modelling and selection algorithms work together. So the main role for the Bayesian model is to focus on providing points that are separable into two clusters.

3.6.2 Likelihood based selection

In MCMC settings, we can consider the generated samples as joint distributions or marginal distributions. The latter consideration is more probable when not enough samples are obtained or the dimension of the parameter space is too high. However, even these marginals do not converge to a fixed distribution, but rather to a distribution of distributions since the transition kernel, which is approximated as indicated in (3.4), is constantly changing at every iteration. The other signal that proves this is demonstrated by the simulations, since we know that the correct variables should have about the same importance (which translates to equal proportions or approximate marginal probabilities). Yet we observe different values for them in Figure 3.4 which shows the subjectivity of the observed sample. All Bayesian selection models we have considered almost certainly yield different samples for different runs, which highly suggests that we account for the variability within each sample.

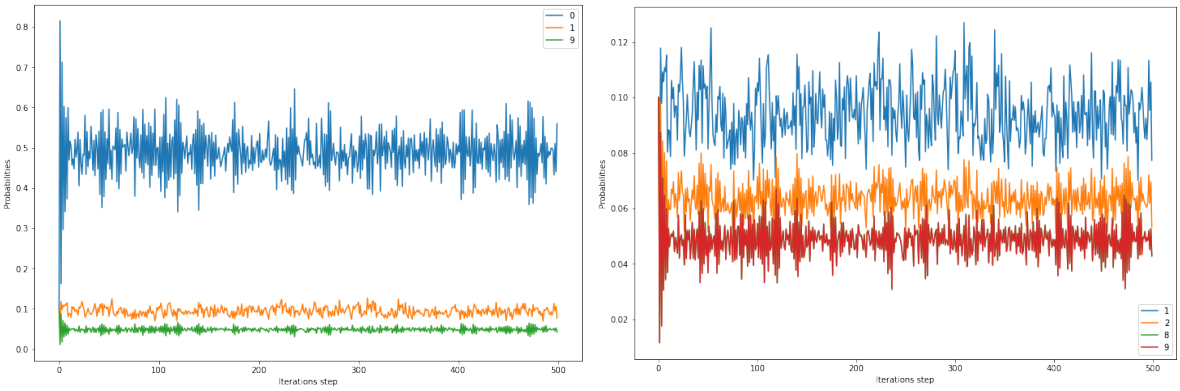
This variability is due to the fact that the relationship between γ_{i-1} and γ_i is evaluated on a marginal distribution different from the one used to compare the likelihood of γ_k and γ_{k-1} because of the other unknown variables that are being explored simultaneously. In other words, we are dealing with a non-homogeneous Markov chain and this type of chain presents some extra challenges.

An illustration

As we explained, we are dealing with a random transition matrix. Since the latter is only defined implicitly (computing the full matrix of just 10 variables would require storing $2^{10} \times 2^{10}$ numbers, doing any additional computation that is non linear in complexity would just blow up), we will illustrate its effect with an example that we explored in the first chapter concerning the introduction to Markov Chains. There will be a slight modification in formulation:

$$\tilde{p}_{i0} \sim \Gamma(2.7, 0.5),$$
$$\tilde{p}_{ij} = \frac{1}{1 + i^2 j^2},$$

where $i, j \in \Omega = \{0, 1, ..., 9\}$ and $\tilde{p}_{ij}$ corresponds to the unnormalized probability p_{ij}. So we have that the transition matrix is conditioned on some gamma variable in order to replicate an environment with a random transition matrix. As usual, let's plot the true stochastic distribution of the chain μ_{θ_t} over t in Figure 3.5.



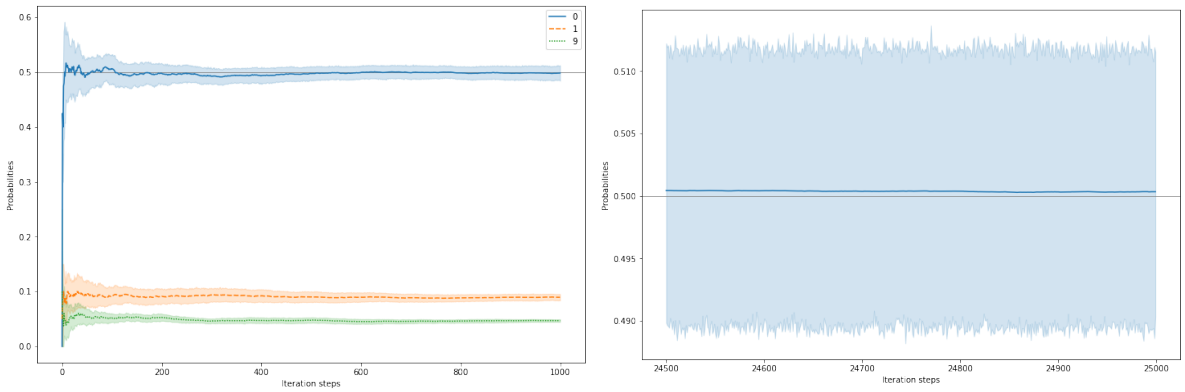
(a) states : w_0, w_1, w_9 (b) states : w_1, w_2, w_8, w_9

Figure 3.5: Distribution of cumulative states probabilities

Again, the left figure shows the probability of being in a given state calculated by using the actual transition kernel conditioned on the random variable $\tilde{p}_{i0}$. As for the right figure, these probabilities are computed cumulatively from a simulation table where each row contains a 1 and nine 0 as to indicate the current state of the chain. The next state is obtaining by simulating the next index given the current state. So we observe that instead of converging to a single value, the probabilities converge to a distribution which explains why the graphs

are wiggly. This is the case for all states since they are all affected by the gamma variable through the normalization procedure.

Now let's take a look at the simulated version of the chain. We run 20 different chains and compute the cumulative probabilities for each state.



(a) states : w_0, w_1, w_9 (b) zoom on the last 500 iterations for w_0

Figure 3.6: Multiple distributions of cumulative states probabilities

We see that even around 25000 iterations, the confidence interval consistently keeps the same gap, while the mean is consistently above 0.5 as well. So one chain can easily converge below 0.5 even though the true value is above that level. And even if we can afford to run multiple chains for our specific problem no matter how computationally expensive it could be, the data sample that we use can still induce a similar convergence behavior. So in the context of variable selection, we need to account for the variability coming from the fact that we may be limited by the data sample we have.

A potential solution

So what we can do to tackle this added complexity is to treat these found proportions as random variables themselves coming from a common distribution which is assumed to be a mixture of simple distributions. The number of distributions will depend on the number of clusters (partitions) we have. For our purpose of selection, we will consider two clusters, U and L .

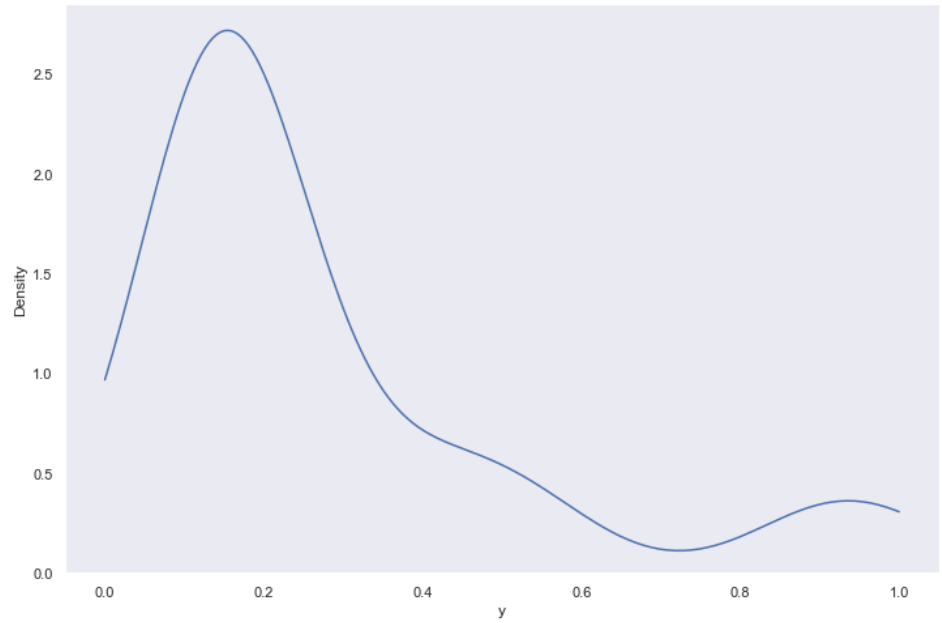


Figure 3.7: Distribution of posterior inclusion probabilities (for 100 variables)

The Figure 3.7 is a plot of the kernel density estimation of gamma proportions. Our goal is to find two appropriate clusters that best reproduce this density. To achieve that, we will maximize the likelihood of a mixture of two densities. Since all proportions $\pi \in [0, 1]$, we will assume that they come from a beta distribution for each group U and L. We have that:

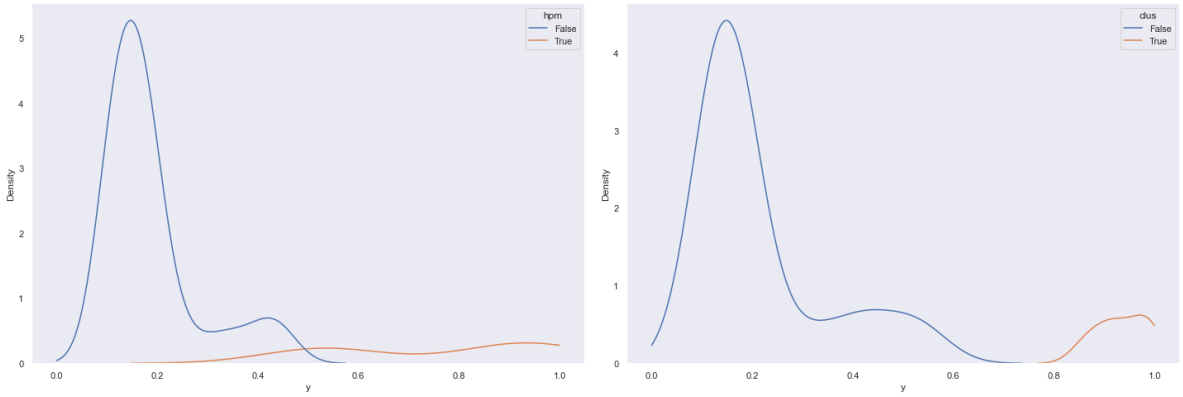
$$\begin{aligned}
\pi &\sim w \times \pi^u + (1 - w) \times \pi^l, \\
\pi^u &\sim \mathcal{B}(a_u, b_u), \\
\pi^l &\sim \mathcal{B}(a_l, b_l).
\end{aligned}$$

We may use a EM algorithm to estimate the hyper-parameters a_u, b_u, a_l, b_l. Since we are in a one-dimensional setting, it is not hard to build an appropriate algorithm for the task using the method of moments, so that's what we used. Let's derive the estimated parameters for $\pi^o \sim \mathcal{B}(a, b)$:

$$\begin{aligned}
\mathbb{E}(\pi^o) &= \frac{a}{a+b} = \frac{1}{n}\sum_1^n \pi_i^o = \bar{\pi}^o, \\
\mathbb{V}(\pi^o) &= \frac{\bar{\pi}^o(1-\bar{\pi}^o)}{a+b+1} = v.
\end{aligned}$$

It is clear that

$$\begin{aligned}
a &= \bar{\pi}^o \left(\frac{\bar{\pi}^o(1-\bar{\pi}^o)}{v} - 1 \right), \\
b &= (1-\bar{\pi}^o) \left(\frac{\bar{\pi}^o(1-\bar{\pi}^o)}{v} - 1 \right),
\end{aligned}$$



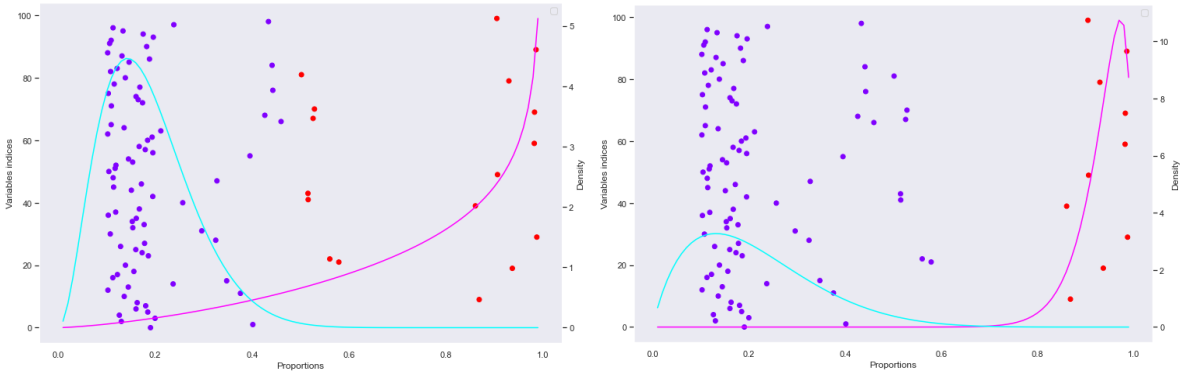
(a) using the Vanilla selection

(b) using the Likelihood based selection

Figure 3.8: Distribution of posterior inclusion probabilities as a mixture of two clusters.

given that $v < \bar{\pi}^o(1 - \bar{\pi}^o)$. We compare the two clusters found by using the two grouping methods. The results are summarized in the Figure 3.8. We can see clearly in Figure 3.8 that the density on the right (likelihood based) is closer to the real distribution shown in Figure 3.7. For both partitions, we find that the left cluster seems to itself be another mixture of two subclusters. However, we may not worry about that depending on the requirements of our application. For this example, it makes sense to observe what looks like a middle cluster since many of the irrelevant variables are highly correlated with the correct variables. But in a real life scenario, we might want to find more than two clusters if they have the potential to be useful for some other purposes.

Figure 3.9 shows the fitted beta distributions in both scenarios as well. We can see that there is a trade-off between the number of points and the concentration of the distribution. For instance, considering the right cluster, including points with proportions in the range $[0.5, 0.6]$ will stretch the density compared to when they are not. Another remark is that for the vanilla selection, the two densities intersect at a point below 0.5. However, these are unweighted distributions and as such they should not be used directly to classify points in their correct clusters.



(a) using the Vanilla selection

(b) using the Likelihood based selection

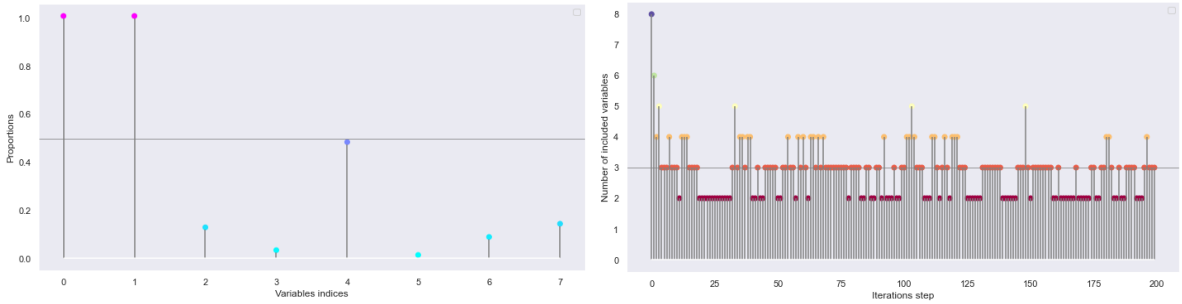
Figure 3.9: Scatter plot of clustered posterior inclusion probabilities with associated density plots.

We can see that this method has a lot of value and is quite robust (since its cutoff value for marginal proportions is not fixed and therefore values fluctuating around 0.5 do not change clusters back and forth depending on different runs). However, it requires that we have enough data points in order to calculate the likelihoods. If we have a small amount of variables, we can either run it for many more iterations and use the vanilla method. Or we can use the following method.

3.6.3 Top k-variables

As indicated by its title, this method is simply about selecting the variables with the highest proportions. The main question is how many variables should we select?

To better illustrate how the method works, let's examine the results obtained from a run with a dataset containing 8 variables (all indentically and independently generated) among which only 3 are relevant (with different coefficients).



(a) Marginal inclusion probabilities

(b) Number of variables selected over iterations

Figure 3.10: Top k-variables selection.

Using the vanilla method, we would have selected the first 2 variables. The proportion of the

variable x_4 is equal to 0.495. This is just below 0.5 and could easily be oscillating around it for various iterations steps (see Figure 3.10). Now let's take a look at the distribution of the number of selected variables throughout the iterations:

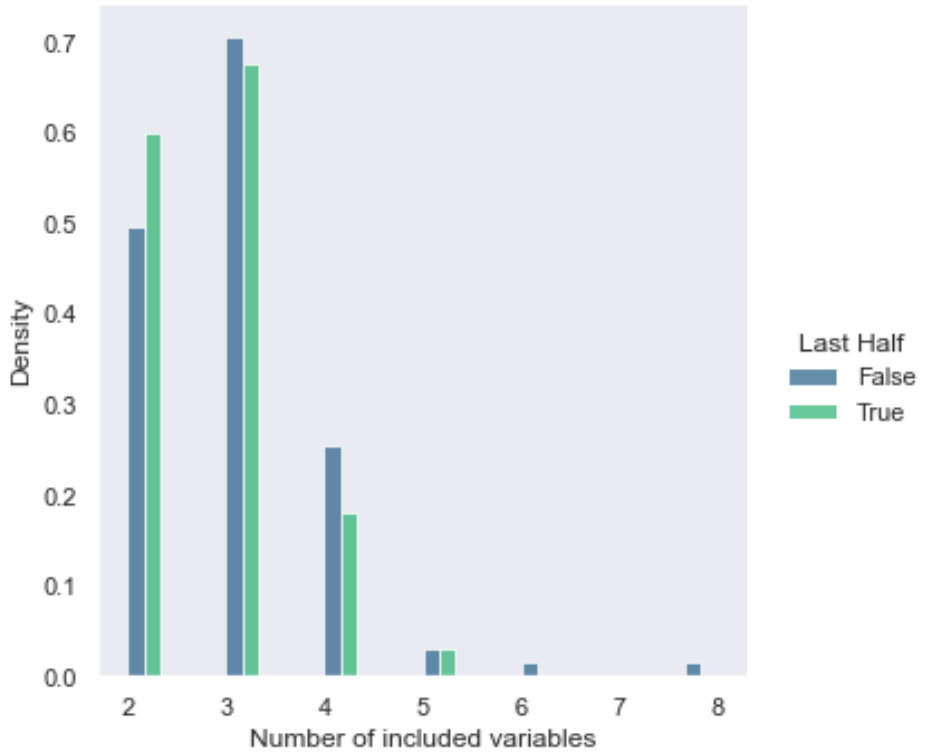


Figure 3.11: Distribution of number of selected variables for the first and last halves of iterations.

We can see that variables are coming in and out without stabilizing to a single set over iterations. However, we see that most of the time, three variables are selected. To verify this assumption, let's take a look at the sample distribution: we see that in both the first half and the last of iteration steps, the mode is equal to 3. The mean is also equal to 2.86, which if rounded is about 3. So we just found the number of variables we should select. We note however that under a certain degree of strong correlations between variables, this method does not generalize well or even fails completely in variable selection. The predictive performance remains strong based on our simulation studies.

3.7 Simulation Study

We will now study the behavior of the algorithm using simulated datasets. In particular, we will be focusing on the three examples studied previously in Leng et al. (2014). We will also compare our selection consistency results with those obtained in Leng et al. (2014) who

studied various variants of the least absolute shrinkage and selection operator (Lasso). It is a remarkable alternative to a convex optimization problem which has a non differentiable component that would best capture the complexity of a model. However, it is not without its drawbacks since high correlation within the design matrix, which causes the choice of an inappropriate hyper-parameter λ, can lead to quite consistently wrong model selection and parameter estimations in addition to giving us more point estimates. So multiple models have been developed to improve it among which the adaptive Lasso (aLasso), the Bayesian Lasso (which will not be considered), the Bayesian adaptive Lasso, and other penalty based methods like ElasticNet.

Before we dive into the simulated datasets, let's specify the hyperparameters that we used for our models. For the first model labeled as by BVE (presented in 3.4), a very important hyper-parameter is the number of inner iterations N_γ for γ. For Example 1 and 2, we set it to $N_\gamma = 1$. For example 3, we set between 1 and 5. The rule of thumb is that the higher the variance (estimated from OLS), the longer that inner loop should run (which means higher N_γ). As for the second model labeled as Walasso (presented in 3.5), there is no inner loop for γ so far (we did not have time to explore such implementation). We set two hyperparameters $d = p + 1$, and $h = 0.0001$ for all simulations. As regards the priors μ_γ, we set them to the OLS estimates. For prediction simulations however, for Walasso, we set it to a vector of zeros. As for the algorithms themselves, we implemented a lot computational tricks (such as having different priors for different γ) that would be too difficult to enumerate all. So we invite you to read the code in our github repository.

3.7.1 Example 1

In this example, we will be working with the following dataset:

$$y = X\beta + \sigma\epsilon.$$

We define corr as the correlation function. And we set the parameter values as :

$$\beta = (3, 1.5, 0, 0, 2, 0, 0, 0)', X = (X_1, \ldots, X_8),$$
$$x_j \sim \mathcal{N}(0, 1), \qquad \text{corr}(x_k, x_l) = 0.5^{|k-l|},$$
$$\epsilon \sim \mathcal{N}_n(0, I_n).$$

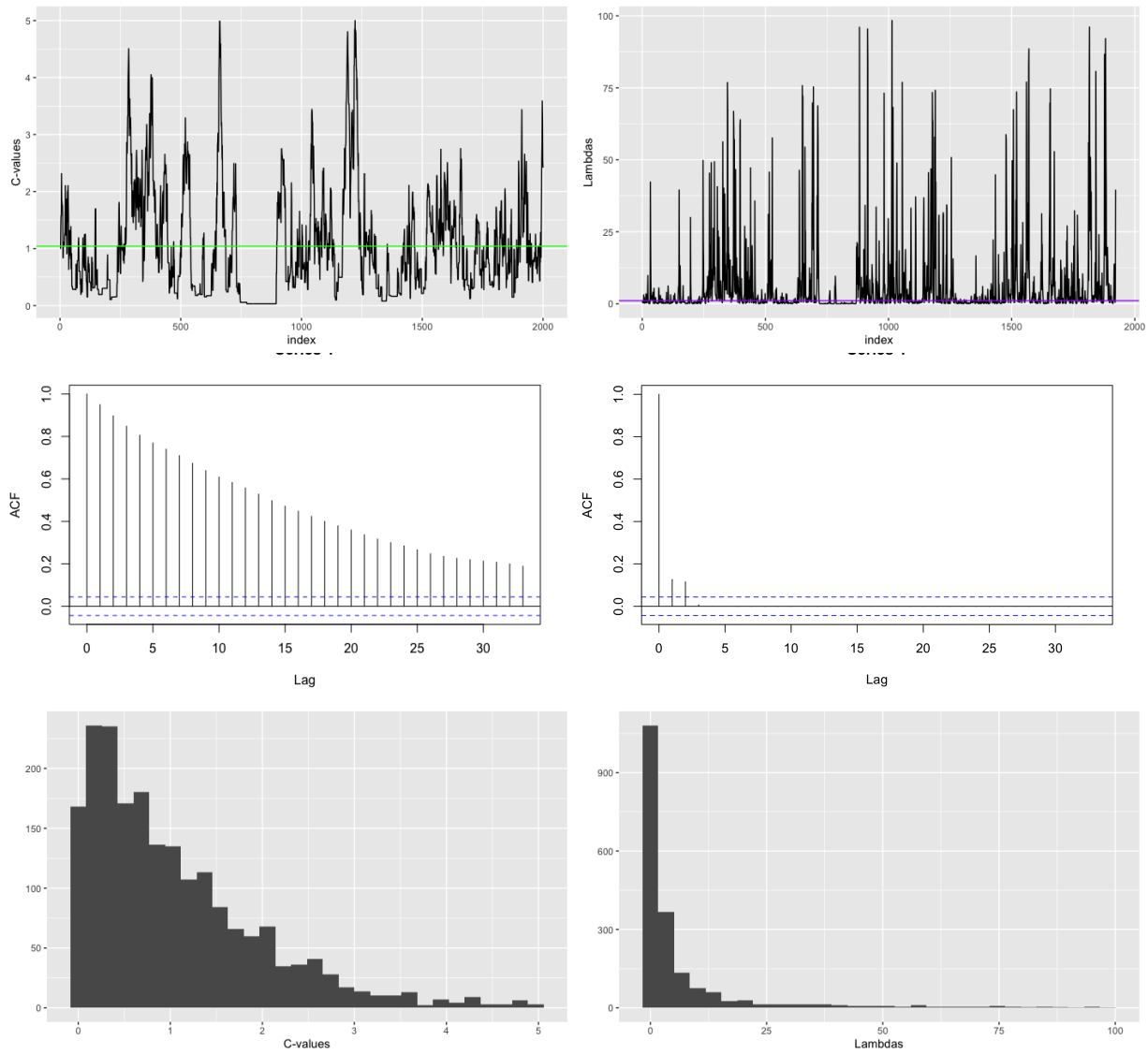


Figure 3.12: Convergence graph, auto-correlation plot and histograms of c (left) and λ (right) for example 1.

We sample 100 datasets from the distribution above and try to select the non-zero variables. We report the frequency of correct selection in the Table 3.1.

Table 3.1: Frequency of correct selections over 100 replications for Example 1.

n	σ	Lasso	aLasso	BaLasso	BVE	WaLasso
30.0	1.0	32.0	85.0	**97.0**	93.0	**97.0**
30.0	3.0	24.0	30.0	1.0	24.0	**37.0**
30.0	5.0	8.0	3.0	0.0	2.0	**11.0**
60.0	1.0	56.0	**100.0**	**100.0**	98.0	**100.0**
60.0	3.0	51.0	63.0	25.0	55.0	**67.0**
60.0	5.0	**25.0**	20.0	0.0	18.0	24.0
120.0	1.0	63.0	**100.0**	**100.0**	**100.0**	**100.0**
120.0	3.0	63.0	88.0	68.0	76.0	**89.0**
120.0	5.0	53.0	43.0	3.0	42.0	**60.0**

Figure 3.12 shows the trace plots, auto correlation graphs and distributuions of c and λ. The trace plots show how correlated these two parameters are (they both get stuck from iteration 750 to around 900).

For evaluating the predictive power of the models, we consider the above data generation specification with $\beta = (3, 1.5, 0.1, 0.1, 2, 0, 0, 0)'$ in order to add more model uncertainty. The results (mean squared error) are presented in Table 3.2.

Table 3.2: Predictive results using Mean squared error over 100 replications for Example 1.

n	σ	Lasso	aLasso	BaLasso	BVE	WaLasso
30.0	1.0	1.497	1.378	1.221	1.18	**1.157**
30.0	3.0	14.11	13.62	18.08	11.64	**10.79**
30.0	5.0	38.017	36.907	42.15	32.235	**29.34**
30.0	10.0	121.82	121.688	120.5771	126.01	**111.52**
100.0	1.0	1.172	1.152	1.0835	**1.03**	1.05
100.0	3.0	10.482	10.453	10.1861	9.68	**9.44**
100.0	5.0	29.104	29.038	31.222	26.09	**26.56**
100.0	10.0	113.329	111.7663	119.249	107.40	**105.44**
200.0	1.0	1.118	1.1227	1.0522	**1.02**	1.04
200.0	3.0	10.032	10.004	**9.1782**	9.203	9.30
200.0	5.0	27.899	27.782	26.7241	25.67	**26.05**

3.7.2 Example 2

We consider another example in relatively low dimensions but which presents other difficulties. Let's lay out the sampling parameters:

$$\beta = (5.6, 5.6, 5.6, 0)',$$
$$\text{corr}(x_k, x_l) = -0.39 \quad when \quad k < l < 4,$$
$$and \quad \text{corr}(x_k, x_4) = 0.23 \quad when \quad k < 4,$$
$$\epsilon \sim \mathcal{N}_n(0, I_n).$$

This turns out to be a quite complicated setting. No matter how many samples are drawn, the Lasso does not give consistent model selection for any λ value chosen. Let's analyze the performance of our proposed method.

Table 3.3: Frequency of correctly-fitted models over 100 replications for Example 2.

n	σ	Lasso	aLasso	BaLasso	BVE	WaLasso
60.0	9.0	2.0	24.0	5.0	**53.0**	48.0
120.0	5.0	0.0	95.0	57.0	91.0	**97.0**
300.0	3.0	0.0	**100.0**	99.0	97.0	**100.0**
300.0	1.0	0.0	**100.0**	97.0	**100.0**	**100.0**

For the following Figures 3.14, 3.15, 3.16 and 3.17, let's note that the first column represents the marginal inclusion probabilities, while the second one shows statistics about the parameter value itself. They are both plotted against the sample sizes. We will be comparing BVE to another Bayesian algorithm known as Bayesian Adaptive Sampling (BAS).

Case β_1

The most striking phenomenon is the fact that sample size is not necessarily related to the marginal inclusion probability of a gamma value. This is very surprising as we expect all statistics to remain at least consistent when n grows large. The Bayesian Variable Selection method yields a pretty narrow confidence interval which consistently traps the true value provided that the corresponding gamma is correctly inferred. When $n = 30$, the expected gamma is wrong. As a result, the predicted β mean is closer to zero and the associated confidence interval misses the true value. However, since the mean is much closer to the lower bound, this is an indication that a longer chain might yield better estimates.

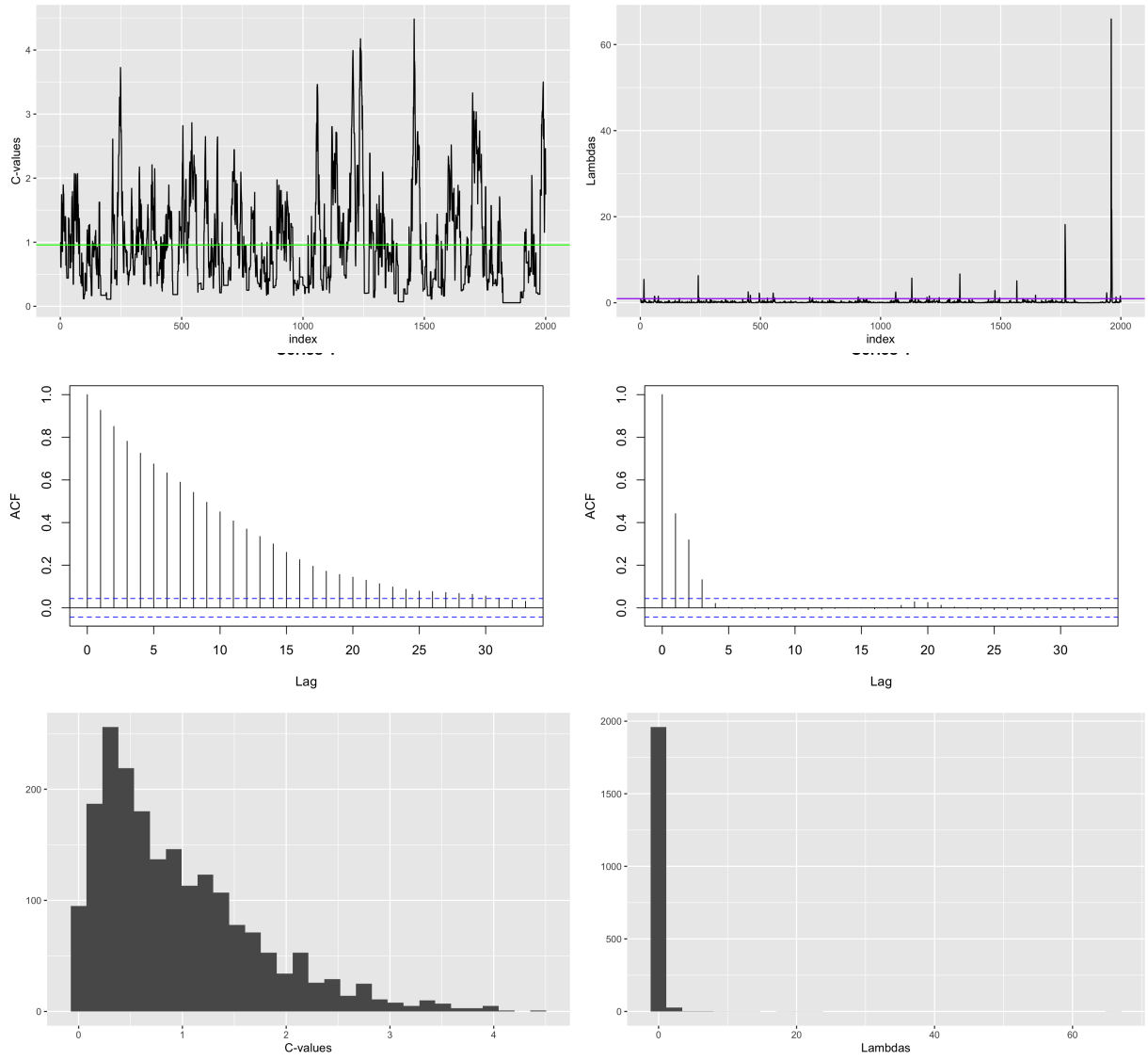
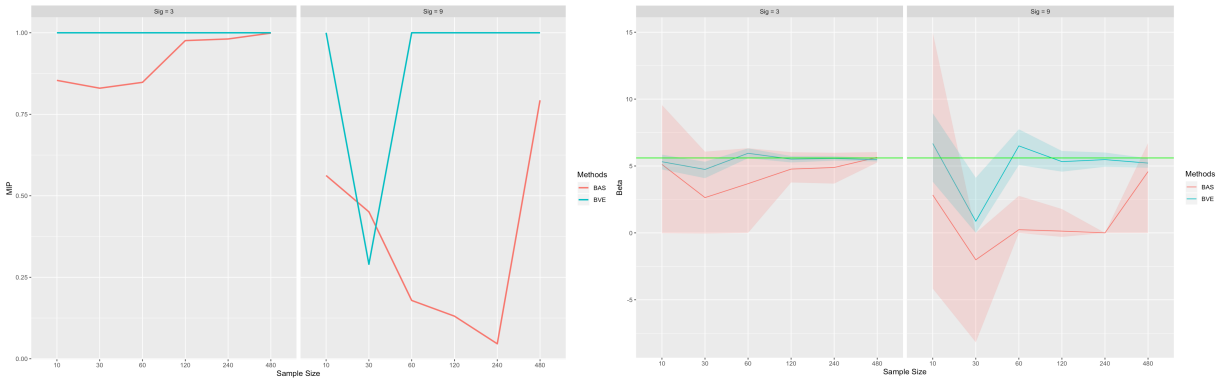


Figure 3.13: Convergence graph, auto-correlation plot and histograms of c (left) and λ (right) for example 2.

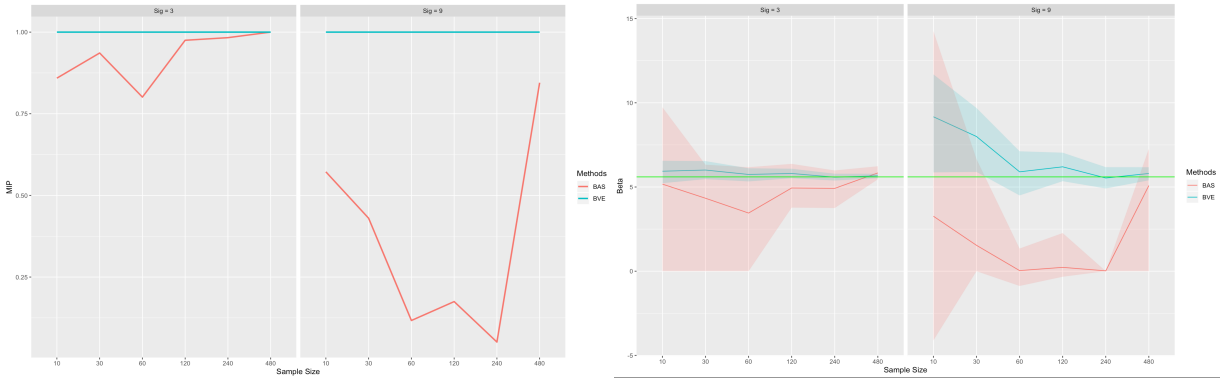


(a) Posterior inclusion probabilities (b) Estimated value (mean 95 % interval)

Figure 3.14: β_1 parameter plots for different sample sizes.

Case β_2

Let's draw our attention on the effect of the systemic error σ. When it is relatively low, we see that the estimates get progressively more accurate and the confidence interval shrinks as well. However, when it is too large, we can see that it does consistently get shorter CIs for the BAS method. As for BVE, it is robust enough and yields expected trends. When n is too low however, it struggles to trap the true value within the CI. It is quite tolerable though due to the randomness of the sampled data, which might be a highly improbable dataset.

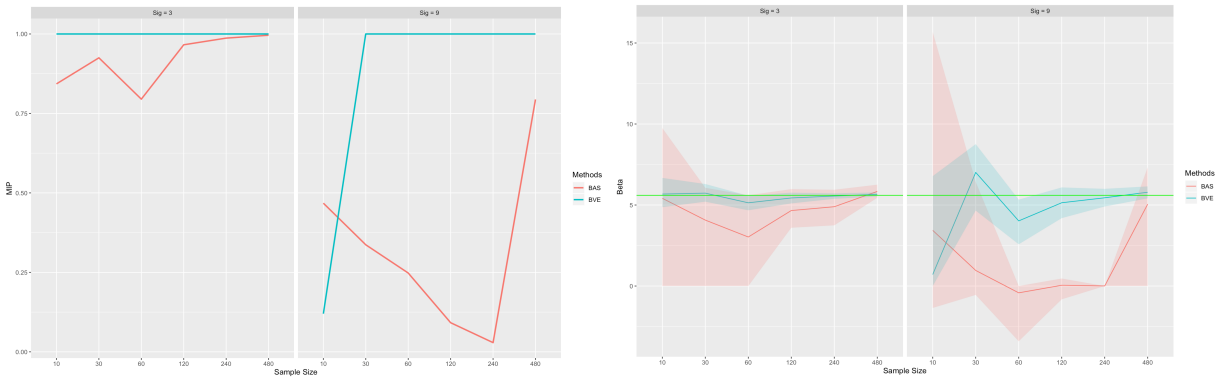


(a) Posterior inclusion probabilities (b) Estimated value (mean 95 % interval)

Figure 3.15: β_2 parameter plots for different sample sizes.

Case β_3

We notice the same situation as in β_1. When n is low ($n = 10$ in this case), the MIP is badly inferred. As a result, the observed values are closer to zero. However its position relative to the Confidence Interval suggests that the variable was activated somewhere in the chain. As for BAS, when n is equal to 240, the credible interval is concentrated around zero. This is because the corresponding gamma remained mostly inactive in the chain.

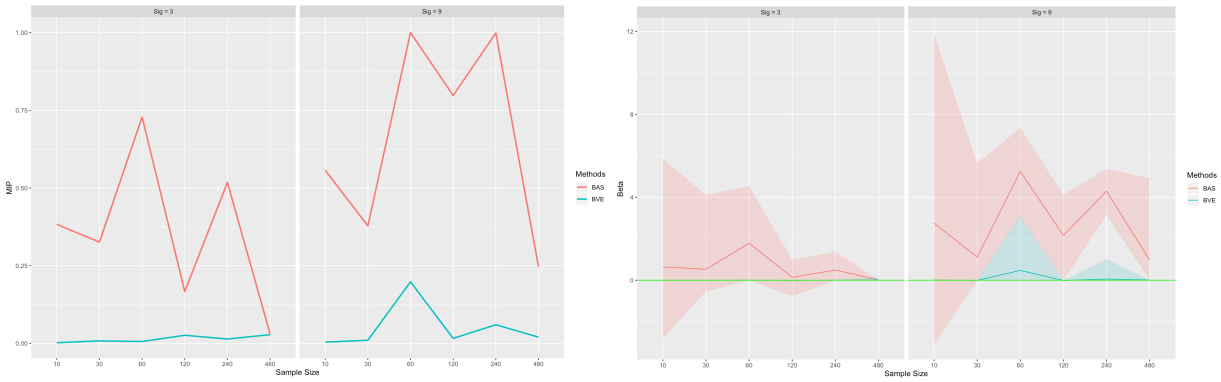


(a) Posterior inclusion probabilities (b) Estimated value (mean 95 % interval)

Figure 3.16: β_3 parameter plots for different sample sizes.

Case β_4

The coefficient β_4 is the only variable that is zero in this example. BVE yields MIP values that reflect its absence for any sample size. As a result, it strongly predicts that $\beta 4 = 0$. Even when the systemic error σ is huge, the expected probabilities fall quickly to low levels even though β_4 oscillates around non zero values. The BAS method struggles to give robust results in this situation. When we inspect it MIP values, we see that they are spread all over the interval $[0, 1]$ as opposed to being concentrated on one side of the 0.5 line. In other words, BAS activates gamma sometimes and does not at other times, independently from the sample size with no apparent trend.



(a) Posterior inclusion probabilities (b) Estimated value (mean 95 % interval)

Figure 3.17: β_4 parameter plots for different sample sizes.

3.7.3 Example 3

This is an extended version of example 1 where we consider much higher dimensions ($\times 10$). The data generation for the design matrix is similar and all coefficients are set to 0 except the following: $\beta_j = 5, \quad \text{for } j = 10, 20, \ldots, 100.$

As in the other examples, we report the selection performance in Table 3.4. We note that there are missing values for WaLasso when $n < p$ because it is not designed for those cases. The same applies for BaLasso because the Matlab package we used failed for these cases and we did not want to modify the authors' code in order to avoid any subjectivity.

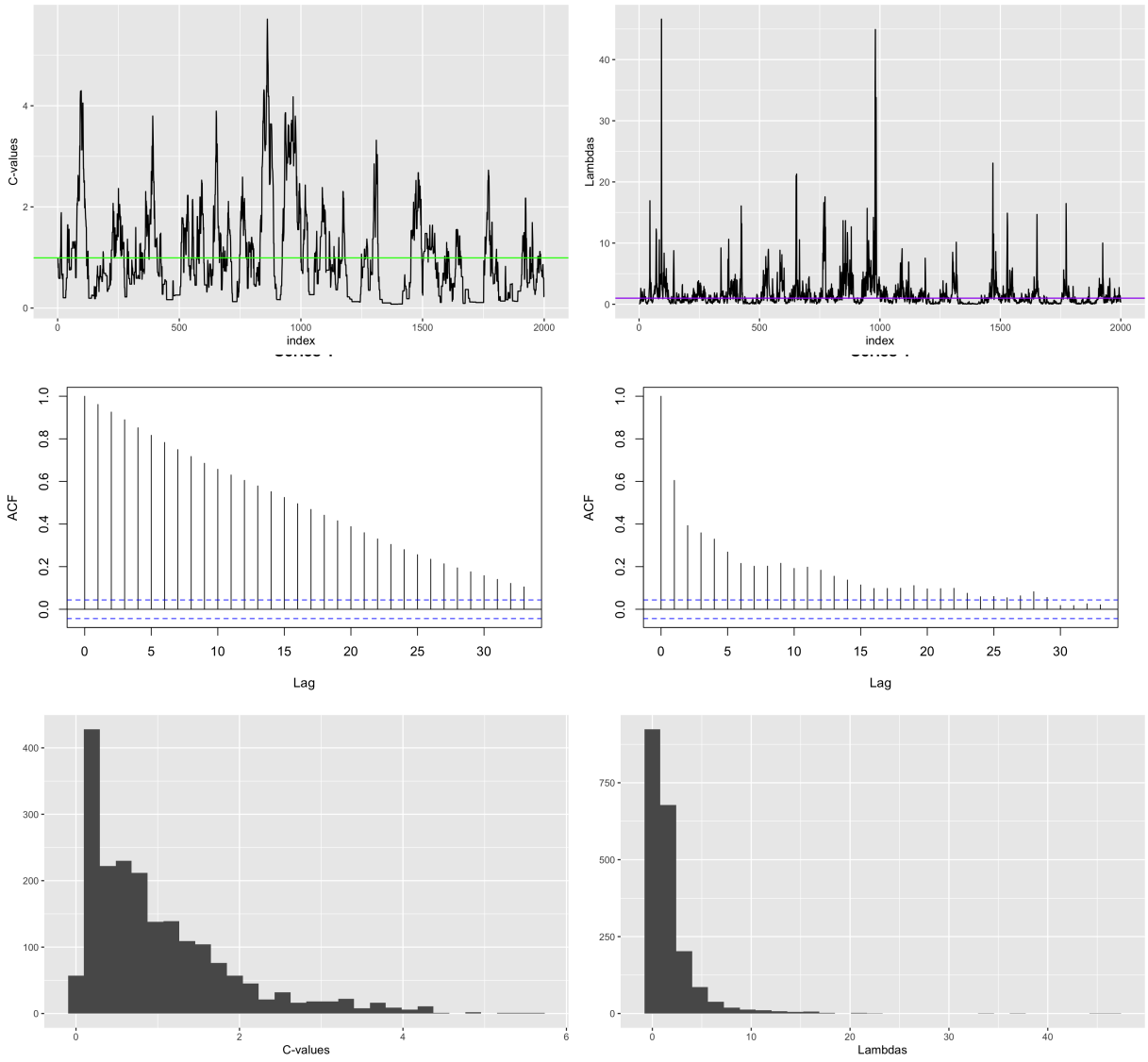


Figure 3.18: Convergence graph, auto-correlation plot and histograms of c (left) and λ (right) for example 3.

For the predictive performances, we modify the following parameter values:

$$\beta_j = 0.5, \quad for\ j = 10, 20, \ldots, 50.$$

This is a type of sparse recovery problem. The Table 3.5 provides a summary of the results (mean squared errors). Note that for BaLasso, we used 110 data points in reality since it fails

for values close to or less than the number of parameters (so it has a bit of advantage). This is the case for Table 3.4 as well.

Table 3.4: Frequency of correctly-fitted models over 100 replications for Example 3

n	σ	aLasso	BaLasso	BVE	WaLasso
30.0	1.0	32.0	nan	38.0	nan
30.0	3.0	1.0	nan	2.0	nan
50.0	1.0	100.0	nan	93.0	nan
50.0	3.0	18.0	nan	35.0	nan
100.0	1.0	42.0	93.0	99.0	**100.0**
100.0	3.0	0.0	91.0	96.0	**97.0**
100.0	5.0	0.0	**92.0**	79.0	81.0
200.0	1.0	**100.0**	**100.0**	**100.0**	**100.0**
200.0	3.0	82.0	**100.0**	98.0	98.0
200.0	5.0	68.0	98.0	97.0	**99.0**

Table 3.5: Predictive results using Mean squared error for Example 3

n	σ	Lasso	aLasso	BaLasso	BVE	WaLasso
100.0	1.0	1.665	4.88	**1.43**	1.6	1.58
100.0	3.0	13.61	21.509	12.54	10.51	**10.37**
100.0	5.0	36.251	53.754	**28.57**	30.47	31.29
100.0	10.0	144.515	179.579	**133.49**	147.37	146.79
200.0	1.0	1.28	1.1451	1.0733	**1.064**	1.085
200.0	3.0	11.369	10.448	10.5434	10.356	**10.145**
200.0	5.0	30.384	27.941	27.1563	26.81	**26.03**
200.0	10.0	118.68	110.69	104.497	104.469	**104.33**

3.7.4 Example 4

In this example, we will be comparing our model with a semi-parametric method (SLM) and a closely linked normal linear model (NLM) (Kundu and Dunson, 2014) with the following main parameter prior $\beta_\gamma \sim N_{p_\gamma}(0, g\tau^{-1}(X_\gamma^{'}\Sigma^{-1}X_\gamma)^{-1})$. The data generation is as follows:

$$y_i = x_i\beta^{'} + \epsilon_i, \quad \epsilon_i \sim 0.5N(2.5, 1) + 0.5N(-2.5, 1)$$
$$\beta = (3, 2, -1, 0, 1.5, 1, 0, -4, -1.5, 0),$$

where each x_{ij} of x_i is generated independently from $U(-1, 1)$, for $j = 1, \ldots, 10$. For the selection part, our model is able to consistently select the same set of non-zero variables about 85% of the time. As for the parameters estimation, we can see the results reported on Table 3.6.

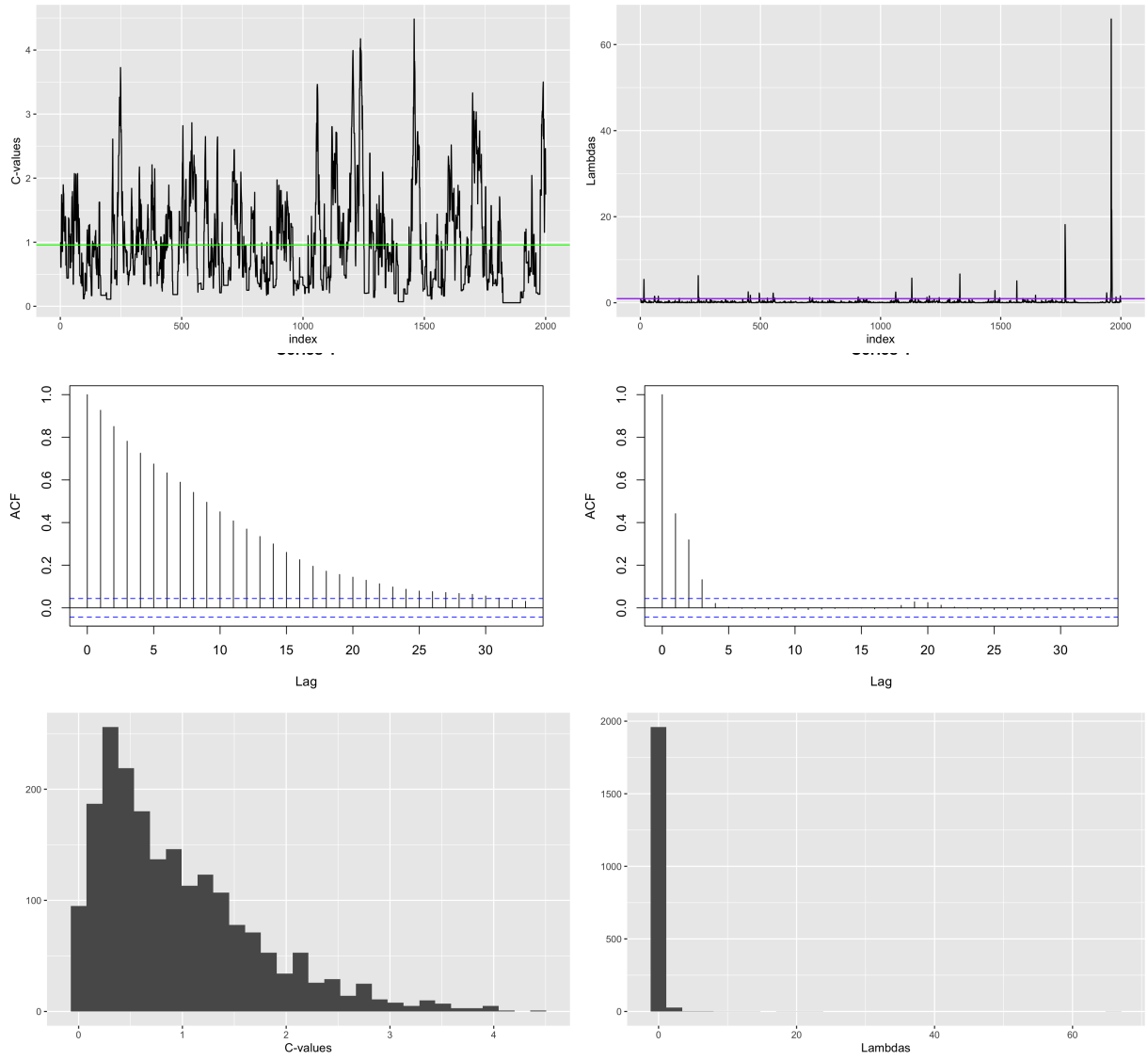


Figure 3.19: Convergence graph, auto-correlation plot and histograms of c (left) and λ (right) for example 3.

Table 3.6: Parameter estimates and marginal inclusion probabilities (MIP)

β_T	MIP_{SLM}	β_{SLM}	MIP_{NLM}	β_{NLM}	MIP_{BVE}	β_{BVE}
3	1.00	2.88 (2.34, 3.41)	1.00	2.83 (1.86, 3.81)	1.00	2.50 (2.07, 2.99)
2	0.99	1.89 (1.34, 2.44)	0.98	1.95 (0.96, 2.91)	1.00	1.98 (1.59, 2.42)
-1	0.93	-0.91 (-1.46, -0.36)	0.75	-0.78 (-1.75, 0.03)	1.00	-1.12 (-1.56, -0.70)
0	0.45	-0.01 (-0.44, 0.44)	0.53	0.006 (-0.82, 0.81)	0.001	-0.00 (-0.00, 0.00)
1.5	0.98	1.43 (0.89, 1.98)	0.90	1.35 (0.35, 2.35)	1.00	1.48 (1.12, 1.87)
1	0.90	0.79 (0.28, 1.35)	0.68	0.54 (-0.26, 1.48)	1.00	1.00 (0.59, 1.42)
0	0.43	-0.005 (-0.44, 0.42)	0.53	-0.05 (-0.85, 0.73)	0.001	-0.00 (-0.00, 0.00)
-4	1.00	-3.89 (-4.43, -3.33)	1.00	-3.75 (-4.74, -2.74)	1.00	-3.63 (-4.07, -3.24)
-1.5	0.99	-1.54 (-2.08, -0.98)	0.92	-1.43 (-2.41, -0.41)	1.00	-1.43 (-1.80, -1.06)
0	0.42	0.008 (-0.43, 0.43)	0.54	-0.12 (-0.93, 0.64)	0.006	-0.00 (-0.00, 0.00)

The results in parentheses represent 95% credible intervals. We also note that β_T stands for the true parameters. BVE inference is not able to trap the correct value of $\beta_1 = 3$ and barely does so for $\beta_8 = -4$ which are two of the most extreme values. This behavior is not very surprising since its specialty is solely on variable selection. However, the credible intervals are smaller in BVE than in the other two methods for most of the parameters.

3.7.5 Application: a study of diabetes data

We will study a dataset that contains diabetes related information. The data consists of 19 variables on 403 subjects who were interviewed in a study to understand the prevalence of obesity, diabetes, and other cardiovascular risk factors in central Virginia for African Americans (Schmidt et al., 1992). According to Dr John Hong, Diabetes Mellitus Type II (adult onset diabetes, DM II) is associated most strongly with obesity. The waist/hip ratio may be a predictor in diabetes and heart disease. DM II is also associated with hypertension - they may both be part of "Syndrome X". The 403 subjects were the ones who were actually screened for diabetes. Glycosolated hemoglobin > 7.0 is usually taken as a positive diagnosis of diabetes.

The variables we will use for our model are given below:

1. chol: Total Cholesterol
2. stab.glu: Stabilized Glucose
3. hdl: High Density Lipoprotein
4. glyhb: Glycosolated Hemoglobin
5. age: age (years)
6. gender: male or female

7. height: height (inches)
8. weight: weight (pounds)
9. bps: Systolic Blood Pressure
10. bpd: Diastolic Blood Pressure
11. waist: waist in inches
12. hip: hip in inches
13. time.ppn : Postprandial Time when Labs were Drawn in minutes

We will contruct the following variables:

1. whr: waist to hip ratio
2. bmi: Body mass index from height and weight
3. ppt: time.ppn < 120 (from Dunson's)
4. obe: Obesity (bmi > 30)
5. ow: Overweight ($25 <$ bmi < 30)
6. > 55: patient is more than 55 years old

Let's look at the distribution of the target variable Glycosolated Hemoglobin (in Figure 3.20). It looks skewed with the mode centered around 5. We run the Walasso model (with same priors and hyper-paramters as defined earlier) on the whole dataset and obtained the resulting selection in Figure 3.21.

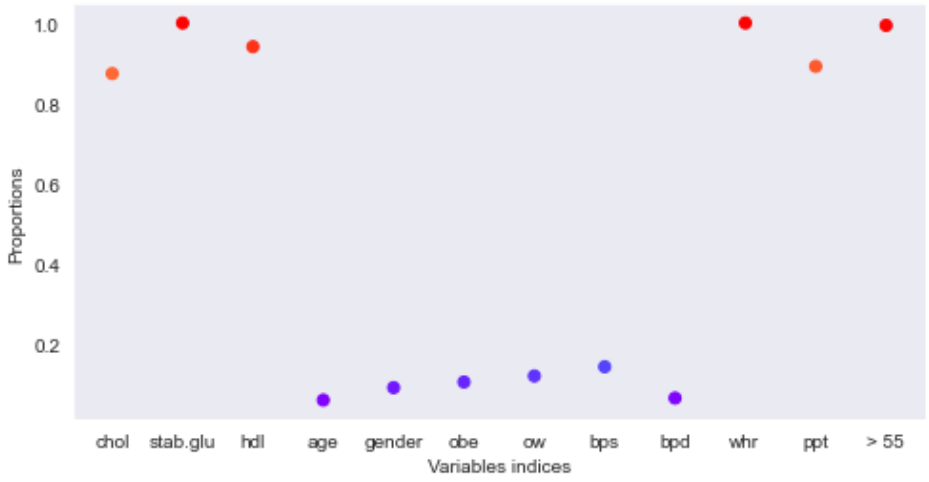


Figure 3.21: MPIs found using the model Walasso

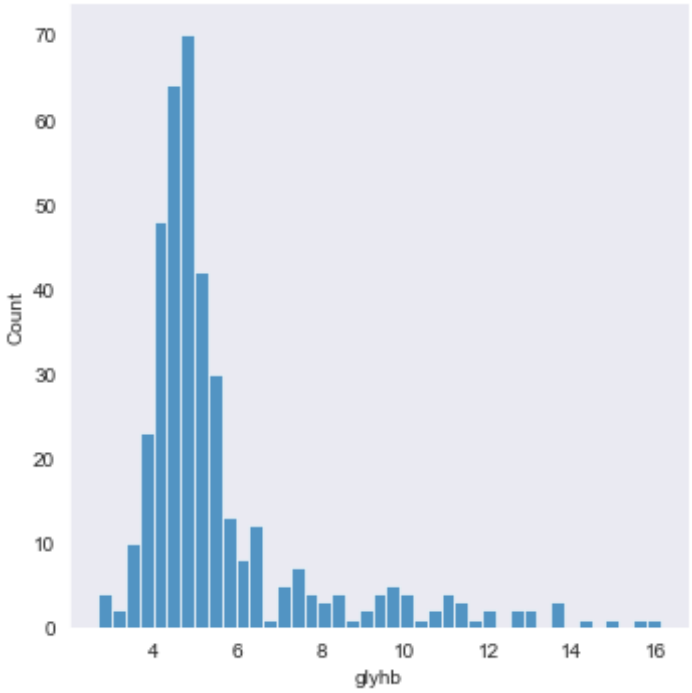
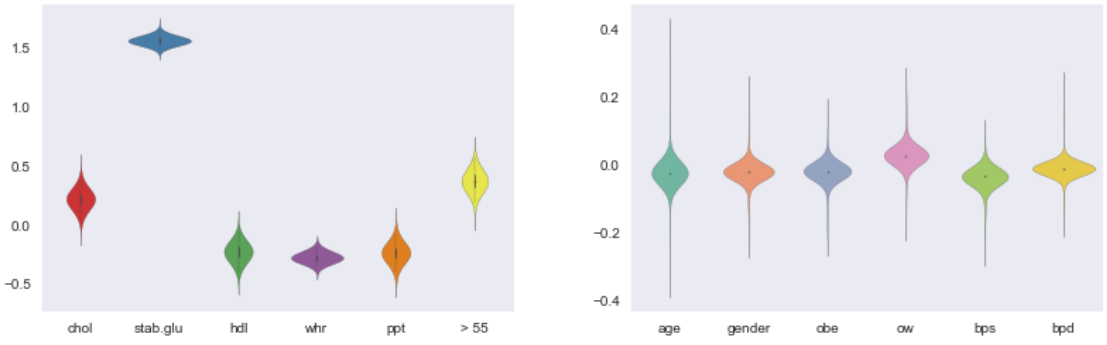


Figure 3.20: Distribution of the target variable Glycosolated Hemoglobin

The graph shows the marginal inclusion probabilities for each variable. There is a clear and unambiguous grouping of important and irrelevant variables here. So there is no need for a selection algorithm. Let's take a look at the estimated parameters (Figure 3.22)



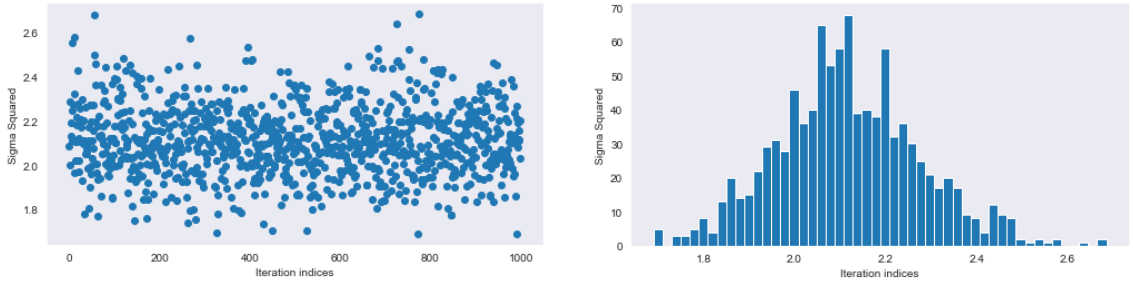
(a) Selected variables

(b) Excluded variables

Figure 3.22: Distribution of the variables with 95 % credible intervals

On the left, we have estimated values for the selected variables. We observe that there is not much deviation from the mean. On the right side, we see the excluded variables. Their estimated means may vary but they are supposed to be zero. The graph also shows that some relatively extreme values (deviating from zero) have been tried. This is a sign of good exploration of the parameters space.

Let's look at some results for the estimated variance (Figure 3.23)



(a) Exploration of σ^2 over iterations

(b) Distribution of σ^2

Figure 3.23: Plots showing information about the variance σ^2

The left figure shows the generated values over iterations. The right one shows the distribution of the variance. It looks more normal than a gamma shape. This is a side effect of the confidence of the model in selecting the appropriate variables. In other words, areas that would have yielded very high variances have pretty much flat densities. And yet, the model is not stuck due to high correlation of samples. To support that hypothesis, let's look at the autocorrelation plots (Figure 3.24) We see that the correlations die off pretty quickly.

Next we turn to prediction performances using cross validation sets to get better information about results. We will compare the Walasso with another new modeling framework known as NGBoost which implements the idea of natural gradient boosting in order to get probabilistic predictions (Ng and et al, 2020). So it outputs a full probability distribution conditioned on the regressors. It works for any base learner in principle. For this simulation study, we use the default learner with the Lognormal distribution option (which is appropriate for this dataset and works better than the default Normal dist option as we tested both and picked the best one). Let's take a look at results in Table 3.7.

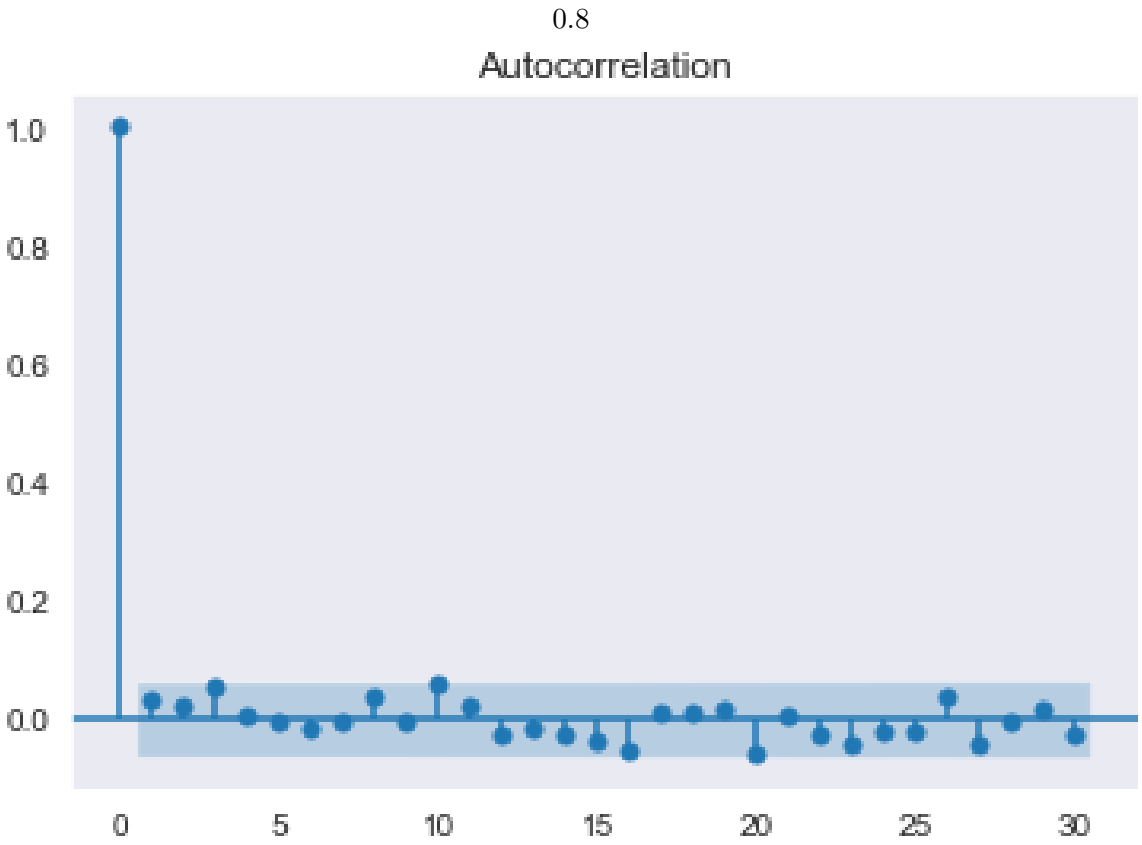
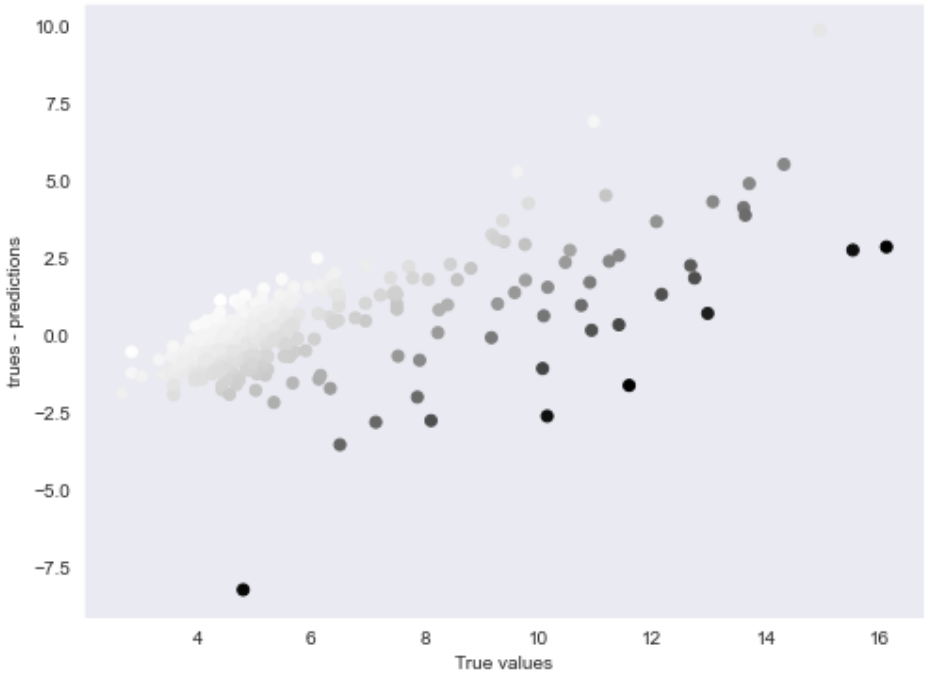


Figure 3.24: Autocorrealtion plot for the variance chain

Table 3.7: Table of cross-validation results. MSE: mean squared error, COV: coverage accuracy, CIW: confidence interval width.

	Walasso			NGBoost		
Folds	**MSE**	**COV**	**CIW**	**MSE**	**COV**	**CIW**
Fold 1	1.7157	0.9744	6.069	1.5149	0.7179	2.5105
Fold 2	1.2611	1.0	5.3084	1.9434	0.7436	2.6955
Fold 3	0.8781	1.0	6.0133	4.5154	0.641	2.6976
Fold 4	1.2824	1.0	5.6669	4.3707	0.7179	2.3703
Fold 5	3.8303	0.9744	6.0293	1.1396	0.8974	3.0518
Fold 6	1.4116	0.9744	6.0392	2.7402	0.7436	2.5669
Fold 7	1.8173	0.9487	5.3442	2.6767	0.7436	2.4802
Fold 8	1.882	0.9744	5.6052	0.9686	0.8974	2.6298
Fold 9	1.6701	0.9744	5.3951	1.095	0.8462	2.5446
Fold 10	1.2555	1.0	5.8861	2.5618	0.8718	2.8032
Averages	1.70041	0.98207	5.73567	2.35263	0.78204	2.63504

The Walasso yields a smaller mean squared error on average. Even though the confidence intervals of NGBoost are smaller, they only capture less than 80 % of the true values compared



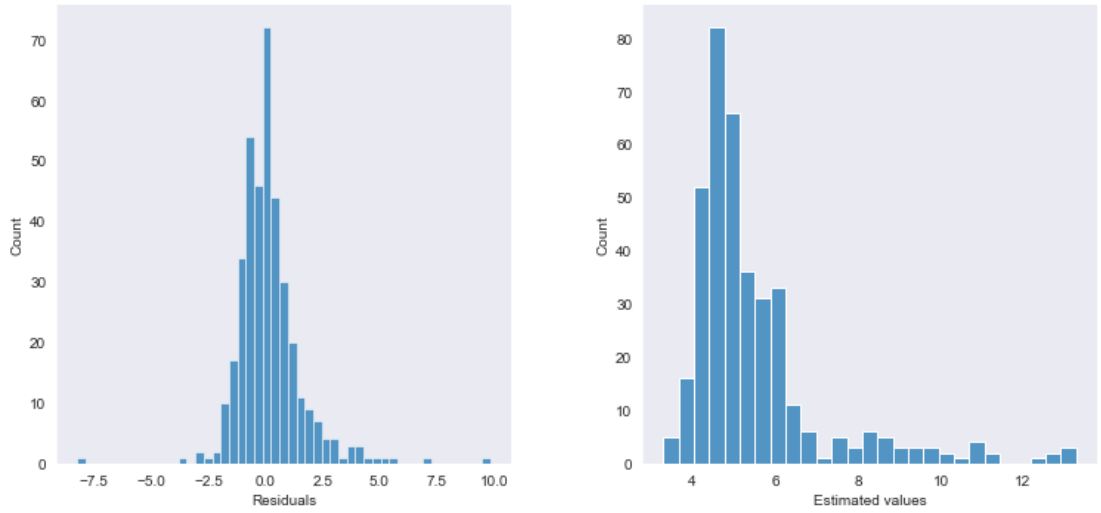
.8

Figure 3.25: Distribution of error terms over true values. Color map represents predicted values from low (white) to high (black)

to 98+ % for Walasso. When we inflate the NGBosst intervals to get similar level of accuracy, the CIW increases to around 7.9 (corresponds to 97.948 %). From a Bayesian perspective, we note that a semi-parametric model developed by Kundu and Dunson (2014) yields a mean squared error of on average 1.6, a coverage accuracy of 98 %, and a mean coverage interval width of about 5.80 (a little more). In fact, we are using our own confidence interval optimizer which, for a given desired average accuracy, produces intervals that have a corresponding CIW. We chose 98 % because that is the level achieved by the semi-parametric linear model above. For some folds, Walasso misses exactly one point. Those points are extreme values that the model ignored in favor of better inference of the clustered points. They might be considered as outliers.

Let's look at the plot of residuals in Figure 3.25. The coloring shows how the predictions relate to the actual values. It varies from low values colored toward the white spectrum and high values tending to black colors. We see that some errors seem to break off the symmetry of the error cluster and are probably outliers. But we chose not to exclude them, since we did not test for that hypothesis.

We also have plots of the residuals and the predictions distributions in Figure 3.26



(a) Residuals

(b) Distribution of predictions

Figure 3.26: Plots showing information about predictions

Conclusion

This thesis tackles the problem of variable selection in settings with highly correlated regressors and less data points than variables ($n < p$). We propose a modeling framework designed to mitigate these relevant issues, as well as algorithms that process the variable search output (sampled from MCMC methods) by computing the posterior inclusion probabilities and use them in order to partition the variables into two sets: the selected variables and the irrelevant ones.

First, we give a brief introduction of the methods that we use to develop all models and tools needed for variable selection algorithms. We also introduced key Bayesian techniques and calculation tricks useful for deriving complex distributions in a relatively easy way. Then, we lay out the derivation of every component for the two models developed. In conjunction with the latter, we provide an intuition for the new post-model selection algorithms that we proposed and used. Lastly, we test the resulting full models on simulated data as well as on real datasets and compare our performance to other well known algorithms such as the Bayesian adaptive lasso, the adaptive lasso and the regular lasso.

With this work, we hope to have participated in the literature on variable selection and Bayesian inference of linear models. We specially think that developing better post-inference selection algorithms, which has been crucial to unlocking the full the potential of our models, is a fertile ground for future research.

Appendix A

Machine learning models

Support Vector Regression

The SVR is a model that minimizes the norm of the regression weights and applies constraints that determine how much error is permissible. To account for the fact that some predictions may lie outside the set margin, it adds those margin deviations to the loss function.

$$\begin{aligned} Minimize \quad & \frac{1}{2}\|w\|^2 + c\sum_{1}^{n}|\xi_i|, \\ & subject \quad to \\ & |y_i - w_i x_i| \leq \epsilon + |\xi_i|. \end{aligned}$$

where c is a regularization constant. The other parameters are defined in figure A.1.

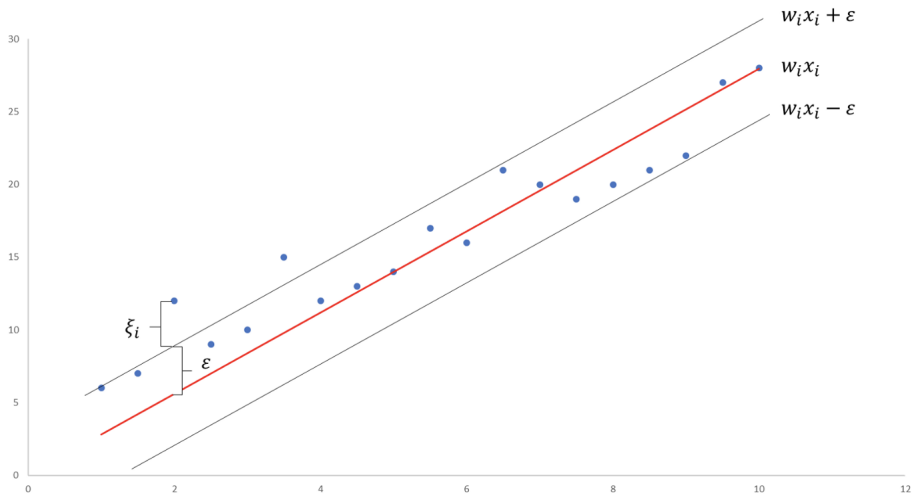


Figure A.1: Illustration of the SVR minimization constraints

Kernel Ridge

It is similar in spirit to both ridge regression and support vector regression. The difference with the first is that it replaces the design matrix dot products with other kernels.

$$\begin{gathered} Minimize \quad \sum_{1}^{n} \xi_i^2 + c\|w\|^2, \\ subject \quad to \\ y_i - w_i x_i = \xi_i, \end{gathered}$$

where w represents the regression parameters. In some fast implementations, the constraints are included in the loss equation which turns into a Lagrangian.

ElasticNet

It is a combination of ridge and lasso regression. It tends to select more variables than the regular Lasso and has better predictive power because of the ridge effect:

$$\hat{\beta} = \arg\min_{\beta} \|y - X\beta\|^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\beta\|^2.$$

Gradient Boosting Regression

Like other boosting methods, it combines many weak (with easy inference mostly) learners into a pipeline in order to fit the data. In order to find a function such that $\hat{y} = F(x)$, which minimizes the mean squared error between predictions and true values, a series of functions h_m are found such that each one is fitted on the residual of the previous one. The final function can be seen as a series of functions F_m such that:

$$F_m(x) = F_{m-1}(x) + h_{m-1}(x) = \hat{y}.$$

For the function used in this benchmark, the weak learner is a decision tree. We note that this model has many hyper-parameters to be chosen.

Orthogonal Matching Pursuit

It is a forward feature selection method that approximates the linear least squares fit with the added constraint on the number of non-zero coefficients (l_0 norm):

$$\begin{gathered} Minimize \quad \|y - X\beta\|^2, \\ subject \quad to \\ \|\beta\|_0 \leq p. \end{gathered}$$

It can also be formulated as follows:

$$\begin{aligned} Minimize \quad & \|\beta\|_0, \\ subject \quad & to \\ \|y - X\beta\|^2 \leq \xi, & \end{aligned}$$

where ξ is an error threshold.

Least Angle Regression

It is a computationally fast kind of forward step-wise regression method developed by Bradley Efron, Trevor Hastie, Iain Johnstone and Robert Tibshirani. It is efficient and numerically stable for high dimensional data where $n < p$. Because of its flexibility, some implementations of Lasso use a modified version of this method. As opposed to introducing variables at each step, the parameters are modified to move in a direction proportional to the correlation of the corresponding regressor with the residual.

Description of case experiments

We use the popular python package pycaret to build the models, train them on half of the data, and evaluate them on the remaining portions. All models described above are created and tuned (optimized) using a 5 fold cross-validation. For each example, we generate n data points for training and n other data points for testing purposes. σ is the standard deviation of the error terms.

- Case 1: Example1 ($n = 30$, $\sigma = 3$),
- Case 2: Example1 ($n = 100$, $\sigma = 5$),
- Case 3: Example2 ($n = 30$, $\sigma = 5$),
- Case 4: Example3 ($n = 100$, $\sigma = 1$),
- Case 5: Example3 ($n = 200$, $\sigma = 1$),
- Case 6: Example3 ($n = 1000$, $\sigma = 5$).

Table A.1: Performance of ML models on first three cases. MSE: mean squared error, Corr: Correlation between predictions and true values

	Case 1		Case 2		Case 3	
Models	**MSE**	**Corr**	**MSE**	**Corr**	**MSE**	**Corr**
LassoLars	14.5339	0.7777	28.5074	0.6514	44.4894	nan
SVR	21.0633	0.6558	33.4886	0.5559	56.8619	0.5395
ElasticNet	14.2964	0.7973	28.556	0.6545	42.7291	nan
GradientBoostingRegressor	20.9413	0.6386	34.1588	0.5656	61.5879	nan
OrthogonalMatchingPursuit	14.7276	0.7584	28.4993	0.6394	36.5252	0.7263
KernelRidge	12.8278	0.782	28.0454	0.6424	32.4948	0.7579

Table A.2: Performance of ML models on last three cases. MSE: mean squared error, Corr: Correlation between predictions and true values

	Case 4		Case 5		Case 6	
Models	**MSE**	**Corr**	**MSE**	**Corr**	**MSE**	**Corr**
LassoLars	9.26	0.9871	7.962	0.9893	31.4614	0.9084
SVR	97.8155	0.6864	71.6656	0.8026	48.469	0.8566
ElasticNet	25.4175	0.952	16.6224	0.9769	45.4834	0.8952
GradientBoostingRegressor	67.5455	0.7034	34.2235	0.8758	38.5695	0.8693
OrthogonalMatchingPursuit	1.7592	0.9932	1.1243	0.9955	26.6224	0.9079
KernelRidge	29.2315	0.8779	3.4309	0.9863	29.0975	0.8993

Bayesian models try to estimate the systemic error that is inherent in the data. Underestimating it is as bad or maybe worse than overestimating it. For these machine learning models, we observed that the error metrics in the training set were sometimes much smaller than the true variance. However, we can see in the test metrics that they don't always generalize well. Only the Orthogonal Matching Pursuit yields results that are sometimes better than certain Bayesian models such as the aLasso or BaLasso.

Bibliography

C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.

M. Baragatti and D. Pommeret. A study of variable selection using g-prior distribution with ridge parameter. *Computational Statistics and Data Analysis*, 56.

M. M. Barbieri and J. O. Berger. Optimal predictive model selection. *The Annals of Statistics*, 32:870–897, 2004.

C. Bishop. *Pattern recognition and machine learning*. Springer: New York, 2006.

P. Brémaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation and Queues*. Springer: Texts in Applied Mathematics, 2020.

G. Casella and E. Moreno. Objective Bayesian variable selection. *Journal of the American Statistical Association*, 101(473):157–167, 2014.

George Casella and Edward I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

Siddhartha Chib and Edward Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

D. Gamerman and H. F. Lopes. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Chapman Hall/CRC, 2006.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6): 721–741, 1984.

S. Kundu and David B. Dunson. Bayes variable selection in semiparametric linear models. *Journal of the American Statistical Association*, 109:437–447, 2014.

C. Leng, Minh N. Tran, and D. Nott. Bayesian adaptive Lasso. *Ann Inst Stat Math*, 66: 221–244, 2014.

David Mackay. *Information Theory, Inference, and Learning Algorithms*. Number 4. Cambridge University Press, 2005.

J. M. Marin and C. P. Robert. *Bayesian Core: A Practical Approach to Computational Bayesian Statistics*. Springer-Verlag New York, 2007.

N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6): 1087–1092, 1953.

T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

I. Murray and Z. Ghahramani. *A note on the evidence and Bayesian Occam's razor*. Gatsby Unit Technical Report GCNU-TR 2005-003, 2005.

Andrew Ng and et al. Ngboost: Natural gradient boosting for probabilistic prediction. *International Conference on Machine Learning*, 2020.

R.B. O'Hara and M. J. Sillanpaa. A review of Bayesian variable selection methods: What, how and which. *Bayesian Analysis*, 4(1):85–118, 2009.

M. Plummer, N. Best, K. Cowles, and K. Vines. Coda: convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11, 2006.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer (2nd ed.), 2004.

C. P. Robert and G. Casella. *Introducing Monte Carlo Methods with R*. Springer, 2009.

M. I. Schmidt, B. B. Duncan, L. H. Canani, C. Karohl, and L. Chambless. *Association of Waist-Hip Ratio With Diabetes Mellitus. Strength and Possible Modifiers*. Number 15. Diabetes Care, 1992.

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

A. Zellner. *On Assessing Prior Distributions and Bayesian Regression Analysis with g Prior Distributions". In Goel, P.; Zellner, A. (eds.). Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti. Studies in Bayesian Econometrics and Statistics*. Number 6. New York: Elsevier, 1986.

H. Zou. The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.