THE SAFE-BAYESIAN LASSO

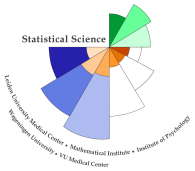
Rianne de Heide

Thesis advisor: Prof. Dr. P.D. Grünwald

MASTER THESIS

Defended on June 2nd, 2016

MATHEMATICAL INSTITUTE



STATISTICAL SCIENCE FOR THE LIFE AND BEHAVIOURAL SCIENCES

Contents

Chapter 1

Introduction

Many problems in statistics involve *estimation*: making an inference about a population based on information obtained from a small number of examples. Suppose we have a set of measurements of these examples $X_1, X_2, \ldots, X_n$ (the *sample*) that are independent and identically distributed (*i.i.d.*) according to a distribution P^*: each variable X_i is from the same probability distribution and they are mutually independent. This underlying, *true* distribution P^* is unknown and we want to estimate it by $\hat{P}$, an element of a set of candidate probability distributions $\mathcal{M}$, which we call the *model*.

We have to make a choice for the model $\mathcal{M}$. We can choose a small, restrictive set of distributions that have a straightforward interpretation and make the estimation mathematically simple. For example, the normal model is widely used. It consists of all normal distributions on the space of possible values of the measurements, the *sample space*, which in this case is $\mathbb{R}$. To obtain the estimate $\hat{P}$, only the mean and variance of the data need to be estimated. However, lifetimes for example are not well represented by a normal distribution. Hence we can also choose a larger model, at the cost of interpretability and mathematical or computational convenience, that might bring us closer to the truth.

But what *is* the truth, the distribution P^*? Do 'true distributions' exist in the sense that they reflect the full reality? By definition a model is an approximation of reality, often a simplified and idealized representation, used for understanding a phenomenon or trying to make predictions. George Box's famous statement "Essentially, all models are wrong, but some are useful" (Box and Draper, 1987) implies that there is no such thing as a true model. For example, in biology, the height of men is assumed to be normally distributed. But the range of the normal distribution is infinite in either direction, and we can be fairly sure from biological assumptions that someone's height cannot be negative or absurdly large. Still $\hat{P}$ is useful for the understanding and prediction of the heights of men.

Whether or not the 'truth' P^* is able to comprise full reality, we can often not assume that P^* is an element of our model. This means that our model is *misspecified*. Does this cause trouble? Often it does not. Applied statisticians use *misspecified* models continually, and the models may be wrong, but useful. Yet the word *often* is disturbing. Can things go horribly wrong?

Outline

In this thesis we empirically investigate the behaviour of the Bayesian lasso of Park and Casella (2008) under model misspecification in simple linear regression problems, analogous to Grünwald and van Ommen (2014) in their paper on model averaging/selection and Bayesian ridge regression. In problems where the model is misspecified, yet useful — it contains a good, not a true predictor — it turns out that the Bayesian lasso can be inconsistent: it does not find this good distribution. To repair the problem we implement and investigate the *Safe Bayesian* method of Grünwald (2012), instantiated to the Bayesian lasso.

This thesis is organized as follows. The first chapter starts with an introduction to the problem that is the main theme of this thesis and an overview of our main conclusions. The rest of the chapter consists of a comprehensive overview of linear regression, frequentist and Bayesian statistics, and a further introduction to the problem of Bayesian inconsistency under model misspecification. Chapter 2 covers the (Safe-)Bayesian lasso and its implementation. In Chapter 3 the results of the simulation studies are presented, and the findings are explained. In the fourth chapter we take a brief look at variable selection and model selection in the (Safe-)Bayesian lasso. Two model selection methods are applied to one of the simulation examples of the preceding chapter. In Chapter 5, we look at the prediction performance of the (Safe-)Bayesian lasso on some 'real-world' data sets, that we found in a search for examples in which the Safe Bayesian lasso outperforms the standard Bayesian lasso. The last chapter provides ideas for future work.

Software

All simulations and applications in this thesis are performed in `R`. The core functions of `R` are extended through *packages*. Packages can be created by any `R`-user, and are available at some repository, such as the Comprehensive R Archive Network (CRAN). One of the strong advantages of `R` for statistical computation is that it supports matrix arithmetic. `R` is however considerably inefficient for iterative, procedural code, such as a Gibbs sampler. Fortunately, `C`, `C++` and `Fortran` code can be linked and called from `R`.

For this thesis we implemented several functions in `R`, that shortly will be made publicly available as an `R`-package, called `SafeBayes`. The core function of this package is a function for the η-generalized Bayesian lasso, described in detail in Chapter 2. Its implementation is heavily based on functions from the `monomvn` package of Gramacy and Pantaleo (2009) and the `BLR` (Bayesian Linear Regression) package of de los Campos and Pérez (2010). Part of the Gibbs sampler is developed in `C` and coupled with `R`. Furthermore, the package provides a function for the generalized Bayesian lasso for models with fixed variance, and the four versions of SafeBayes. One can choose from several priors. In Appendix B, the user manual that will accompany the package is provided, starting with a section explaining how to initialize the functions while the package is not yet available to install directly from the repository.

Our implementation of the Gibbs sampler is relatively fast: 10000 iterations in 206.28 seconds (3.5 minutes) for a 201-dimensional Fourier basis, sample size 100 and non-informative priors, compared to, for example the popular `monomvn` package of Gramacy and Pantaleo (2009) under the same circumstances: 1837.48 seconds (30.6 minutes).

The problem

Let us look at an example to introduce the problem. We sample data $(X_1, Y_1), (X_2, Y_2), \ldots$ i.i.d. from a 'true' distribution P^*. We consider a *regression setting*: we want to find a relationship between the X_i's and the Y_i's. Since the X_i are random as well, this is called *regression with random design*. We can choose what kind of relationship we would like to obtain, and in this thesis we will mainly desire one of a linear combination of sines and cosines. Therefore, we will perform linear regression with a so called *Fourier basis*. This set-up will be explained in detail in Section 1.1. In this thesis we investigate a special form of regression: the *lasso*, which will be described in Chapter 2. Moreover, we will focus on its counterpart in the *Bayesian framework*. The Bayesian framework will be explained in Section 1.3.

We continue with our example. First the X_i are sampled i.i.d. from a uniform distribution on $[-1, 1]$. We set the Y_i to $0 + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$: the Y_i's are 'zero with some Gaussian noise'. Now we toss a fair coin for each (X_i, Y_i). If the coin lands heads, we keep the (X_i, Y_i) as it is, and if the coin lands tails, we put the pair to zero: $(X_i, Y_i) = (0, 0)$. Since our 'true' distribution P^* is 'zero with some noise', these points are *easy*. They lie exactly on the 'true' regression function, devoid of noise. The data (X_i, Y_i) are depicted in Figure 1.1. We expect our Bayesian lasso regression to learn the correct regression function; simply zero.

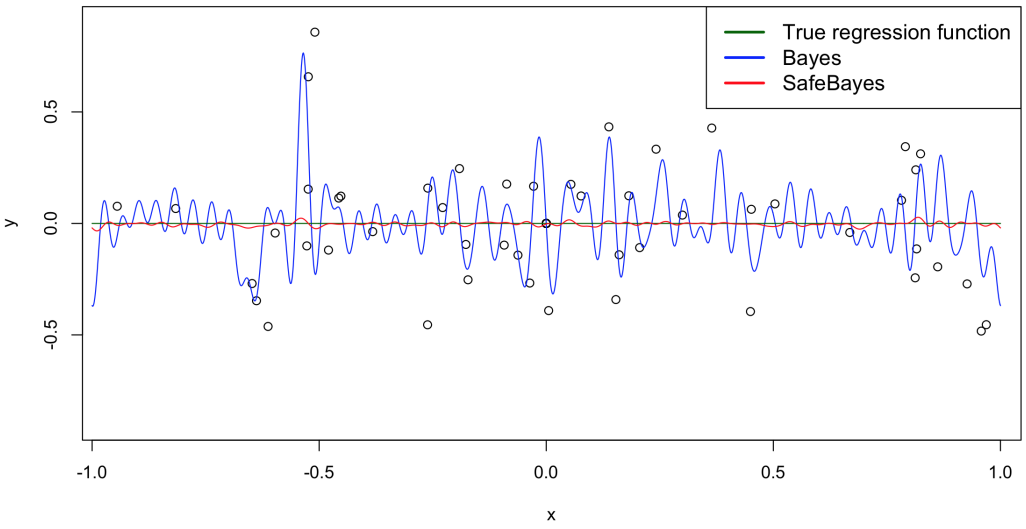


Figure 1.1: The 'true' distribution (green) and predictions of standard Bayes (blue) and SafeBayes (red). Both posteriors are sampled with a 101-dimensional Fourier basis and uninformative priors on 100 data points i.i.d. $\sim P^*$ with approximately half of the data points in $(0, 0)$.

We see in Figure 1.1 what appears to happen: the Bayesian lasso does not learn the underlying distribution, it learns the noise instead. This is called *overfitting*: the curve fits nicely to the data Bayes was trained on. However, if we would make *predictions* for new data, generated according to P^*, Bayes would be far off. Prediction is an important part of statistical inference. Section 1.2 will give a closer examination of prediction, and how to quantify its quality.

An explanation why Bayes overfits, is *model misspecification*: the 'truth' P^* is not an element of the model. In our example, our model uses the assumption of *homoscedasticity*: all the Y_i's should have the same variance. Since we removed the noise of half of our points, our distribution of the data is *heteroscedastic*: the variance of all points with $X_i = 0$ is 0, and all other points have variance σ^2. Model misspecification does not lead automatically to Bayesian trouble, as will be discussed in Section 1.4. In the same section a remedy will be introduced to solve the problem if it occurs: the *generalized posterior*. Bayes' likelihood will be equipped with a *learning rate* η. If η is chosen small enough, Bayes will behave again. It should however not be chosen smaller than necessary, because that makes the procedure needlessly slow in terms of the amount of data needed before a good conclusion can be drawn. A method to learn the 'right' η from the data automatically, is the *Safe Bayesian algorithm* of Grünwald (2012). We see in Figure 1.1 how SafeBayes performs on our example. It seems to solve the problem of overfitting and to learn the correct underlying distribution. An introduction to SafeBayes is given in Section 1.4, and it will be explained in detail with its instantiation to the Bayesian lasso in Chapter 2.

Conclusions

In Chapter 3 we show the results of experiments on simulated data. We perform experiments as described above and some variations on it, and we compare the predictive performance of the standard Bayesian lasso and the Safe-Bayesian lasso. We observe similar phenomena as in Figure 1.1: the standard Bayesian lasso shows considerable overfitting, and the Safe-Bayesian lasso appears to be close to having discovered the true regression function. Fascinatingly, we sometimes observe a weaker version of this happening when the model is *well-specified*. We study the empirical *square-risk*, the expected squared error loss, of Bayes and SafeBayes at different sample sizes in the *misspecified* model set-up. The square-risk of standard Bayes is not only larger than that of SafeBayes, it *grows* with the sample size. Bayes recovers slowly when the sample size becomes larger than the dimension of the basis. We can explain this problematic behaviour by the predictive distribution being a mixture of different 'bad' distributions in the (non-convex) model. As we explain in detail in Section 3.4, Bayes' bad square-risk, while it has good log-risk (Barron, 1998), implies that the posterior is not concentrated. The behaviour of Bayes in the correct-model set-up can be explained by true distribution being expressible as a convex combination of other, bad elements in the model. Safe-Bayes performs excellently in all experiments.

As explained in Section 2.1, the variable selection property of the basic lasso is lost in its fully Bayesian form. Therefore we look at methods for variable selection and model selection in Chapter 4. We investigate the Bayes Factor model selection method and the Deviance Information Criterion (DIC) in the wrong-model experiment with different numbers of basis functions for the standard Bayesian lasso and the generalized Bayesian lasso. Based on the DIC, one would choose the model that gives the worst predictions. The Bayes Factor method yields appropriate results when comparing the standard Bayesian lasso models to the generalized Bayesian lasso models. 'Appropriate' is in the sense of prediction error, since prediction is the focus of this thesis, but in this case it is in the sense of 'close to the true regression function' as well. However, when we compare the standard Bayesian lasso models with different dimensions of the basis mutually, the Bayes Factor method cannot save Bayes.

Bayes' bad predictive performance under model misspecification proves to be to not only a problem in theory, but in practice as well, as we see in Chapter 5. For this chapter, we searched for data sets in which the Safe-Bayesian lasso outperforms the standard Bayesian

lasso, and we present three examples of those. Importantly, in all data sets examined — the examples as well as numerous data sets encountered in our search — the Safe-Bayesian lasso never performed substantially worse than the standard Bayesian lasso. Moreover, SafeBayes sometimes performs substantially better than standard Bayes.

Lastly, we anticipate on future research in Chapter 6 with an interesting variation on the Bayesian lasso. The Bayesian lasso *without* the parameter λ, and *with* a learning rate η performs excellently in our initial experiments, which are presented in this last chapter. Our new method could be a better alternative to the standard Bayesian lasso, in which the specification of the parameter λ proves to be difficult, as described in Section 2.4. Not only does our new method perform comparably to the standard Bayesian lasso, it outperforms the standard Bayesian lasso substantially when the model is misspecified.

1.1 Linear Regression

The following introduction to linear regression is largely based on Chapter 12 of Grünwald (2007), and will follow its notation.

In linear regression, we want to find a relationship between a *regressor* variable $\boldsymbol{u}^n = (u_1, \ldots, u_n) \in \mathcal{U}^n$ and a *regression* variable $\boldsymbol{y}^n = (y_1, \ldots, y_n) \in \mathbb{R}^n$, where $\mathcal{U}$ is some set. We would like to learn a function $g : \mathcal{U} \to \mathbb{R}$ from the data to gain understanding of the relationship between the variables, or to make predictions for new data $(y_{n+1}, \ldots, y_{n+k})$ given the variables $(u_{n+1}, \ldots, u_{n+k})$. Additionally — in Bayesian linear regression — we assume Gaussian noise on $\boldsymbol{y}^n$, that is

$$Y_i = g^*(U_i) + Z_i \,, \tag{1.1}$$

where g^* is the true function we would like to learn, and $Z_i \sim N(0, \sigma^2)$, i.i.d. and independent of U_i. According to (1.1) we can formulate the conditional density of $y_1, \ldots, y_n$ given $u_1, \ldots, u_n$ by

$$f_{g,\sigma^2}(\boldsymbol{y}^n | \boldsymbol{u}^n) = \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right)^n \exp\left(- \frac{\sum_{i=1}^n (y_i - g(u_i))^2}{2\sigma^2} \right) . \tag{1.2}$$

An obvious way to find a function g for our regression problem is to look for it in a p-dimensional space of functions spanned by some basis. In *linear* regression we limit our search to (finite) linear combinations of *basis functions*: $g_\beta(u) = \sum_{i=1}^p \beta_i g_i(u)$ for some $\beta = (\beta_1, \ldots, \beta_p) \in \mathbb{R}^p$.

We first make the concept of a basis for a functional space more precise, then we discuss the basis that we will use throughout this thesis, and continue with the introduction on linear regression thereafter.

Basis functions

The basis for a vector space is a familiar concept from linear algebra: a linearly independent spanning set for that space. We would like to have a similar concept for bases for a function space. Following Rynne and Youngson (2008), we extend the idea of an orthonormal basis to infinite dimensional spaces as follows.

Definition 1 (Def. 3.37 from Rynne and Youngson (2008))**.** Let X be an inner product space. A sequence $\{e_n\} \subset X$ is said to be an *orthonormal sequence* if $\|e_n\| = 1$ for all $n \in \mathbb{N}$, and $\langle e_m, e_n \rangle = 0$ for all $m, n \in \mathbb{N}$ with $m \neq n$.

We want our regression function g to be able to be expressed as a linear combination of an orthonormal sequence of measurable functions $g_1, g_2, \ldots$:

$$g = \sum_{i=1}^{\infty} \langle g, g_i \rangle g_i \,. \tag{1.3}$$

Let us look at the probability space $(\Omega, \mathcal{F}, P)$ with an inner product $\langle g, h \rangle = \int_\Omega gh \,\mathrm{d}P$, from which the L^2 norm on measurable functions g follows: $\|g\|_2 = (\int_\Omega |g|^2 \mathrm{d}P)^{1/2}$. Now the set $L^2(\Omega, \mathcal{F}, P)$, which is the space of equivalence classes of $\mathcal{F}$-measurable functions g with $\|g\|_2 < \infty$, is a Hilbert space (since the metric space $L^p(X)$ is complete for $1 \leq p \leq \infty$). Because $L^2(\Omega, \mathcal{F}, P)$ is a Hilbert space, (1.3) converges for all $g \in L^2(\Omega, \mathcal{F}, P)$ (Corollary 3.44 from Rynne and Youngson (2008)). We conclude by defining an orthonormal basis for a Hilbert space.

Definition 2 (Def. 3.49 from Rynne and Youngson (2008))**.** Let $\mathcal{H}$ be a Hilbert space and let $\{e_n\}$ be an orthonormal sequence in $\mathcal{H}$. Then $\{e_n\}$ is called an *orthonormal basis for* $\mathcal{H}$ if (1.3) holds for all $g \in \mathcal{H}$ (condition (d) from Thrm. 3.47 from Rynne and Youngson (2008)).

An instance of such an orthonormal basis that we will use throughout this thesis, is the *Fourier basis*, for $k \in \mathbb{N}$:

$$\begin{aligned} g_1 &= \frac{1}{\sqrt{2\pi}} \,, \\ g_{2k} &= \frac{1}{\sqrt{\pi}} \cos(k\boldsymbol{u}) \,, \\ g_{2k+1} &= \frac{1}{\sqrt{\pi}} \sin(k\boldsymbol{u}) \,, \end{aligned} \tag{1.4}$$

which is here a basis on $L^2([-\pi, \pi], \mathcal{B}[-\pi, \pi], U[-\pi, \pi])$ where $\mathcal{B}[-\pi, \pi]$ is the Borel σ-algebra and $U[-\pi, \pi]$ the Lebesgue measure on $[-\pi, \pi]$. Of course it can be extended to a general interval $[a, b]$ by a change of variables $x \to \tilde{x} = a + (b - a)x/\pi$.

Linear regression continued

We return to our linear model, for which we briefly go over some notation and definitions. We define the *design matrix* $\boldsymbol{X}$:

$$x_i := \begin{pmatrix} g_1(u_i) \\ \vdots \\ g_p(u_i) \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{X} := \begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix} . \tag{1.5}$$

The *Sum of Squared Errors* (SSE), which we can see as the Euclidean length of the vector of errors with respect to our regression function g_β, is defined as

$$\mathrm{SSE}(\beta, \boldsymbol{y}) := \sum_{i=1}^{n} (y_i - g_\beta(u_i))^2 = \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 = (\boldsymbol{y} - \boldsymbol{X}\beta)^T (\boldsymbol{y} - \boldsymbol{X}\beta)\,, \tag{1.6}$$

and we define the *least squares estimator* $\hat{\beta}$ as

$$\hat{\beta} := \underset{\beta \in \mathbb{R}^p}{\arg\min}\, \mathrm{SSE}(\beta, \boldsymbol{y})\,, \tag{1.7}$$

which corresponds to the *maximum likelihood estimator* (MLE) of (1.2), because we can rewrite (1.2) as

$$f_{\beta,\sigma^2}(\boldsymbol{y}^n | \boldsymbol{u}^n) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{\mathrm{SSE}(\beta, \boldsymbol{y}^n)}{2\sigma^2}\right)\,. \tag{1.8}$$

When $\boldsymbol{X}$ has full column rank, there exists a unique *least squares estimator*

$$\hat{\beta} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}\,, \tag{1.9}$$

and the corresponding estimate of $\boldsymbol{y}$ is

$$\hat{\boldsymbol{y}} := \boldsymbol{X}\hat{\beta} = \boldsymbol{X}(\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}\,. \tag{1.10}$$

We define the *Residual Sum of Squares* (RSS) as the sum of squared errors obtained by the solution $\hat{\beta}$ as

$$\mathrm{RSS}(\boldsymbol{y}) := \mathrm{SSE}(\hat{\beta}, \boldsymbol{y}) = (\boldsymbol{y} - \boldsymbol{X}\hat{\beta})^T (\boldsymbol{y} - \boldsymbol{X}\hat{\beta})\,. \tag{1.11}$$

We introduce two special types of linear models here, because we will use them frequently throughout the next chapters. We will denote by $\mathcal{M}^{(p)} = \{P_\beta | \beta \in \mathbb{R}^p\}$ the family of distributions with densities (1.8) for which the true variance is known, and similarly those for which it is not by $\mathcal{S}^{(p)} = \{P_{\beta,\sigma^2} | \beta \in \mathbb{R}^p, \sigma^2 > 0\}$.

1.2 Prediction and loss

An important part of statistical inference is *prediction*: making a 'forecast' for an unknown observation X_n, given some knowledge about the previous data X^{n-1}. A *prediction method* is a procedure that directly outputs the prediction X_n when given X^{n-1} as input (Grünwald, 2007). We measure the quality of a prediction by a *loss function*, a function that measures the discrepancy between the actual value and the predicted value of the observation. An example that is, not surprisingly, often used for the linear model, is the Euclidean distance between those values: the *squared error loss function* or *square-loss*: $\ell(X_i, Y_i) = (Y_i - X_i\boldsymbol{\beta})^2$.

Besides predicting a single value, we sometimes want to predict a full distribution. A suitable loss function for this type of prediction is the *logarithmic loss* or *log loss*[1], which

[1] Unless otherwise indicated, we refer to the natural logarithm with *log*.

is the negative logarithm of the probability density or mass that is assigned to the true outcome, extended to n outcomes by *adding* the losses. Following Grünwald (2007) again, when p is a distribution on $\mathcal{X}^n$, we can write for every x^n

$$p(x^n) = \prod_{i=1}^{n} \frac{p(x^i)}{p(x^{i-1})} = \prod_{i=1}^{n} p(x_i|x^{i-1}), \tag{1.12}$$

where $p(x_i|x^{i-1})$ is an abbreviation of $p(X_i = \cdot|X^{i-1} = x^{i-1})$. Now the log loss is:

$$-\log p(x^n) = \sum_{i=1}^{n} -\log p(x_i|x^{i-1}). \tag{1.13}$$

If we would predict $\boldsymbol{y}^n$ with a density f_{β,σ^2} from our linear model $\mathcal{S}^X$, we can take the negative logarithm of (1.8) and obtain the log loss:

$$-\log f_{\beta,\sigma^2}(\boldsymbol{y}^n|\boldsymbol{u}^n) = \frac{n}{2}\log 2\pi\sigma^2 + \frac{1}{2\sigma^2}\mathrm{SSE}(\beta, \boldsymbol{y}^n)\,. \tag{1.14}$$

In this special case, we can thus see that the log loss is an affine function of the square-loss.

1.3 Bayesian statistics

In Section 1.1 we looked for parameter values that maximized a likelihood function, so we assumed the parameter to be fixed and the data to be random, in order to obtain a point estimate[2] (with a standard error). In the Bayesian framework the data are treated as fixed (in the sense that we condition on it), the parameters are treated as random variables, and statistical conclusions about parameters or predictions are made in terms of *probability statements* (Gelman et al., 2004). These probability statements are conditional on the observed values x^n. Suppose we have a statistical model $\mathcal{M} = \{p_\theta|\theta \in \Theta\}$, where a parameter θ follows some *prior distribution* $w(\theta)$. Given θ the data x^n are sampled from the distribution p_θ. This renders a joint distribution $p(\theta, x^n) = w(\theta)p_\theta(x^n)$. By Bayes rule we can condition on the observed x^n and obtain the *posterior distribution*

$$p(\theta|x^n) = \frac{p_\theta(x^n)w(\theta)}{\int_\Theta p_\theta(x^n)w(\theta)d\theta}\,. \tag{1.15}$$

Since the denominator does not depend on θ we can treat it as a constant, yielding the *unnormalized posterior* $p(\theta|x^n) \propto p_\theta(x^n)w(\theta)$.

There are several ways to link the posterior distribution (1.15) to point estimation methods, akin to those in Section 1.1. One way to do so is to randomly draw a point from the posterior. This might seem silly, but we will see in the next chapter how it can be of use. Consider a model $\mathcal{M}$. A popular Bayes estimator is the *posterior mean* $\hat{\theta} := \mathbf{E}_{\theta\sim p|x^n}[\theta]$ (Grünwald, 2007). We may not have $\hat{\theta} \in \Theta$, but in all instances in this thesis we have,

[2] Any function that maps data to a parameter value is called a *(point) estimator*.

because in our setting Θ is always equal to $\mathbb{R}^p \cup [0, \infty)$, which is convex. A third Bayes estimator is the *maximum a posteriori* (MAP) estimator, the maximum of the posterior density or posterior *mode*

$$\hat{\theta}_{\text{MAP}} = \arg\max_{\theta \in \Theta} p_\theta(x^n) w(\theta) \,. \tag{1.16}$$

For example, suppose we have a prior distribution $\mu \sim N(\mu_0, \sigma_0^2)$ and data $x = (x_1, \ldots, x_k)$ i.i.d. $\sim N(\mu, \sigma_1^2)$, and we wish to find the estimate $\hat{\mu}_{\text{MAP}}$. Then we have to maximize the following posterior

$$w(\theta) p_\theta(x) = \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left(-\frac{1}{2} \left(\frac{\mu - \mu_0}{\sigma_0} \right)^2 \right) \prod_{i=1}^{k} \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{1}{2} \left(\frac{x_i - \mu}{\sigma_1} \right)^2 \right).$$

The $\hat{\mu}_{\text{MAP}}$ that maximizes the posterior above is given by

$$\hat{\mu}_{\text{MAP}} = \frac{k\sigma_0^2}{k\sigma_0^2 + \sigma_1^2} \left(\frac{1}{k} \sum_{i=1}^{k} x_i \right) + \frac{\sigma_1^2}{k\sigma_0^2 + \sigma_1^2} \mu_0 \,.$$

See for details Gelman et al. (2004).

If we want to make inference or predictions about some unobserved variable x before we have seen the data, we can use the *prior predictive distribution* (1.17). Given a prior distribution $w(\theta)$ the distribution of the unknown x is

$$\overline{p}(x) = \int_{\theta \in \Theta} p_\theta(x) w(\theta) d\theta \,. \tag{1.17}$$

After we have seen data x^{n-1} we can make predictions about X_n by replacing the prior in (1.17) by the posterior, obtaining the *posterior predictive distribution*[3]:

$$\begin{aligned} \overline{p}(X_n = x_n | x^{n-1}) &= \frac{\overline{p}(x^n)}{\overline{p}(x^{n-1})} \\ &= \frac{\int_{\theta \in \Theta} p_\theta(x_n) p_\theta(x^{n-1}) w(\theta) d\theta}{\overline{p}(x^{n-1})} \\ &= \frac{\int_{\theta \in \Theta} p_\theta(x_n) w(\theta | x^{n-1}) \overline{p}(x^{n-1}) d\theta}{\overline{p}(x^{n-1})} \\ &= \int_{\theta \in \Theta} p_\theta(x_n) w(\theta | x^{n-1}) d\theta \\ &= \mathbf{E}_{\theta \sim p|x^n} \left[p_\theta \right](x_n) \,. \end{aligned} \tag{1.18}$$

For a given loss function, we can use the estimator that minimizes the expected posterior predictive loss to make predictions. For a regression model under square-loss, prediction of a future observation $\tilde{y}$ at values $\tilde{X}$ are then made via the mean of the posterior predictive

[3]We will always use bars to denote a predictive distribution.

distribution. Suppose we have a regression model with posterior distribution $p(\beta|y,\sigma^2,\tau)$, where σ^2 and τ are parameters we consider fixed and known for the moment, but we relax this assumption in Chapter 2 where we will look at this model in detail. The posterior predictive distribution is then $p(\tilde{y}|y,\sigma^2,\tau) = \int p(\tilde{y}|y,\beta,\sigma^2,\tau)\, p(\beta|y,\sigma^2,\tau) d\beta$. For this model, the posterior predictive mean turns out to be $\mathbf{E}(\tilde{y}|y,\sigma^2,\tau) = \tilde{X}\,\mathbf{E}(\beta|y,\sigma^2,\tau)$. Thus, we may use the posterior mean $\hat{\beta}$ (a point estimate) to predict $\tilde{y}$ via $\tilde{X}\hat{\beta}$.

The Prior

In the Bayesian paradigm, it is assumed that a statistician can always assign a prior to the parameters in the model $\mathcal{M}$ (Grünwald, 2007). A prior can often be seen as one's belief or knowledge before the observations are made, in which case it can be considered *subjective*. Yet a prior can also be chosen according to some principle to represent a lack of knowledge about the parameter, and then it is called *objective* or *uninformative*. We see from (1.15) that the prior does not need to be a probability distribution, i.e. it can be a measure integrating to ∞. The posterior will still represent a distribution if the sum or integral in the denominator integrates to something finite. A prior distribution is called *proper* if it does not depend on the data and it integrates to 1, and *improper* otherwise.

An uninformative prior that has the desirable property that it is invariant to reparametrisation is *Jeffreys' prior*. Jeffreys' prior is proportional to the square root of the determinant of the Fisher information $I(\theta)$:

$$w_{\text{Jeffreys}}(\theta) = \frac{\sqrt{\det I(\theta)}}{\int_{\theta\in\Theta}\sqrt{I(\theta)}d\theta}\,, \qquad I(\theta) = -\,\mathbf{E}\left[\frac{d^2\log p(y|\theta)}{d\theta^2}\,\middle|\,\theta\right]. \tag{1.19}$$

For example, the Jeffreys' prior for variance $\nu=\sigma^2$ of the Gaussian distribution $f(x|\nu)$ with the mean μ fixed is

$$\begin{aligned} p(\nu) \propto \sqrt{I(\theta)} &= \sqrt{\mathbf{E}\left[\left(\frac{d}{d\nu}\log f(x|\nu)\right)^2\right]} \\ &= \sqrt{\int_{-\infty}^{\infty} f(x|\nu)\left(\frac{(x-\mu)^2-\nu^2}{\nu^3}\right)^2 dx} \\ &= \sqrt{\frac{2}{\nu^2}} \propto \frac{1}{\nu}\,. \end{aligned} \tag{1.20}$$

Jeffreys' prior for the variance $\nu=\sigma^2$ is thus $p(\sigma^2)\propto 1/\sigma^2$. When μ is not fixed, a two-parameter Jeffreys' prior can be computed in different ways. One way is to compute the two-parameter prior $p(\mu,\sigma^2)\propto\sqrt{|I(\mu,\sigma^2)|} = 1/\sigma^3$. Another is to multiply the two single-parameter Jeffreys' prior together: $p(\mu,\sigma^2)\propto\sqrt{|I(\mu)|}\cdot\sqrt{|I(\sigma^2)|} = 1/\sigma^2$. We will use the latter, because that is usually preferred for the normal model.

The *support* of a measure on a measurable topological space X (such as the Borel-σ-algebra we saw in Section 1.1) is defined as the set of all points for which every open

neighbourhood has positive measure. We can view the *support of the prior* as a Bayesian analogue to the choice for the model (Kleijn, 2004), and we can thus in a corresponding way call a model *misspecified* when the true distribution P^* does not have a density p_θ for any θ in the support of the prior.

1.4 Bayesian inconsistency under misspecification

An important concept in statistics is *consistency*. Intuitively, an estimation method is *consistent* if, as more data become available, it chooses distributions that are closer tot the true distribution P^* from which the data are generated, converging to P^* itself eventually. More precisely (from the frequentist point of view): a sequence of estimators $\hat{P}^n$ in a model $\mathcal{M}$, with true distribution $P^* \in \mathcal{M}$ is said to be consistent with respect to a metric d if (Kleijn and van der Vaart, 2006)

$$d(\hat{P}^n, P^*) \xrightarrow{P^*} 0\,. \tag{1.21}$$

Bayesian methods provide (posterior) distributions on $\mathcal{M}$ rather than point estimators. Then *consistency* is roughly taken to mean that the posterior probability concentrates on ever smaller neighbourhoods of the true distribution. Here we consider the nonstandard case with $P^* \notin \mathcal{M}$. Then consistency of a Bayesian method means that the posterior puts its mass on neighbourhoods of $\tilde{P}$: the 'best' approximation in our model. The prior must have significant mass on $\tilde{P}$ for this to take place. How 'close' two distributions with densities p and q are, can be specified in many ways. We will look at three of them: the *Kullback-Leibler (KL) divergence*, the *Hellinger distance* and the *Rényi divergence*.

The *Kullback-Leibler divergence* is defined for probability densities p and q on $\mathcal{X}$ as

$$D(p \parallel q) = \mathbf{E}_p\left[\log \frac{p(X)}{q(X)}\right] = \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} dx\,. \tag{1.22}$$

This is not a metric, because it is not symmetric, but it is nonnegative, with equality if and only if $p = q$. This can be seen from Jensen's inequality, which says that, if f is a convex function and x is a random variable then

$$\mathbf{E}\left[f(x)\right] \geq f(\mathbf{E}\left[x\right])\,. \tag{1.23}$$

The *Hellinger distance* between two probability densities p and q on $\mathcal{X}$ is

$$H(p,q) = \left(\int_{\mathcal{X}} \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 dx\right)^{\frac{1}{2}} = \left(2 - 2\int_{\mathcal{X}} \sqrt{p(x)q(x)} dx\right)^{\frac{1}{2}}. \tag{1.24}$$

Often it is more useful to work with the squared Hellinger distance, which we denote with $H^2(p,q)$. The term $\int_{\mathcal{X}} \sqrt{p(x)q(x)}dx = \mathbf{E}_p\left[\sqrt{\frac{q(x)}{p(x)}}\right]$ is called the *Hellinger affinity*. The Hellinger distance is a metric, and it is 0 when $p = q$ almost surely [4].

[4] Let $(\Omega, \mathcal{F}, P)$ be a probability space. Then an event $E \in \mathcal{F}$ occurs *(P-)almost surely* if $P(E) = 1$.

Let p and q be probability densities on $\mathcal{X}$ again. The *Rényi divergence* of order λ is defined as

$$\overline{d}_\lambda(p \parallel q) = -\frac{1}{1-\lambda} \log \mathbf{E}_q \left[\left(\frac{p(X)}{q(X)} \right)^\lambda \right]. \tag{1.25}$$

We can connect these two divergences and one distance by the following inequalities

$$H^2(p,q) \leq \overline{d}_{1/2}(p \parallel q), \tag{1.26}$$
$$D(p \parallel q) = \lim_{\lambda \uparrow 1} \overline{d}_\lambda(p \parallel q); \quad \overline{d}_\lambda(p \parallel q) \text{ is increasing in } \lambda, \tag{1.27}$$
$$H^2(p,q) \leq D(p \parallel q). \tag{1.28}$$

We see that the KL divergence upperbounds the Rényi divergences. Also, the squared Hellinger distance provides a lower bound for the KL divergence. Thus, convergence in KL divergence implies convergence in Hellinger distance. We can see why (1.26) and (1.28) are true:

$$\begin{aligned}
H^2(p,q) &= \int_{\mathcal{X}} \left(\sqrt{p(x)} - \sqrt{q(x)} \right)^2 dx \\
&= \int_{\mathcal{X}} p(x)dx + \int_{\mathcal{X}} q(x)dx - 2\int_{\mathcal{X}} \sqrt{p(x)q(x)}dx \\
&= 2\left(1 - \int_{\mathcal{X}} \sqrt{p(x)q(x)}dx\right) \\
&\leq -2\log \int_{\mathcal{X}} \sqrt{p(x)q(x)}dx, \quad \text{since } 1-x \leq -\log x \\
&= -2\log \int_{\mathcal{X}} \sqrt{\frac{q(x)}{p(x)}} p(x)dx \\
&= -2\log \mathbf{E}_p \left[\sqrt{\frac{q(X)}{p(X)}} \right] = \overline{d}_{1/2}(p \parallel q) \\
&\leq -2\,\mathbf{E}_p \left[\log \sqrt{\frac{q(X)}{p(X)}} \right], \quad \text{by Jensens inequality (1.23)} \\
&= \mathbf{E}\left[\log \frac{p(X)}{q(X)} \right] = D(p \parallel q).
\end{aligned}$$

For the statements in the coming section, the proofs are in terms of Hellinger distance, but they hold in many cases for KL divergence as well.

It is well-known that when the model is well-specified, the posterior converges fast with high P^*-probability to the true distribution in terms of Hellinger distance, under weak conditions on model and prior (Ghoshal et al. (2000) and Zhang (2006)). This continues to hold, under much stronger conditions, if the model is misspecified. The posterior then concentrates on the distribution $\tilde{P}$ that is closest to P^* in KL divergence. This concentration

around $\tilde{P}$ is sometimes also in KL divergence, sometimes in Hellinger distance, and sometimes in a different metric. However, the posterior does not necessarily concentrate around $\tilde{P}$ if the strong conditions do not hold. Grünwald and Langford (2007) show that Bayesian inference in classification problems can be inconsistent under misspecification: the posterior puts all its mass on distributions that are far away form $\tilde{P}$, both in KL divergence and Hellinger distance at all large sample sizes. Moreover, Grünwald and van Ommen (2014) empirically demonstrate inconsistency under model misspecification for Bayes factor model selection, model averaging and ridge regression. The question arises how we can find out if we are in the 'good' or the 'bad' case of misspecification, in other words, will Bayes concentrate or not on $\tilde{P}$, the best approximation to P^* in the model.

'Good' and 'bad' misspecification

Suppose $\tilde{P}$ is the closest element of the model $\mathcal{M}$ to the true distribution P^* in terms of KL divergence: $\tilde{P} = \arg\min_{P \in \mathcal{M}} D(P^* \parallel P)$. We know that if $P^* \in \mathcal{M}$, when we get more and more data, Bayes will converge to $\tilde{P}$, so the Hellinger distance between the estimate (what we have learnt from the data) and $\tilde{P}$ (the best approximation in our model) almost surely goes to zero, because $\tilde{P} = P^*$. It turns out that this essentially also holds if $P^* \notin \mathcal{M}$ when the model is *convex* (all convex mixtures of densities in the model are in the model as well) (Li, 1999). A requirement that a model needs to be convex for Bayes to converge, seems however too strong. It is sufficient if replacing $\mathcal{M}$ by the convex hull of $\mathcal{M}$ does not decrease $\inf_{P \in \mathcal{M}} D(P^* \parallel P)$. The 'good' and 'possibly bad' cases for a nonconvex model are illustrated in Figure 1.2 (Grünwald and van Ommen, 2014). We come back to this in more detail in Section 3.4.

Assuming we do not know what our true distribution P^* is, how do we know if we are in the 'good' or '(possibly) bad' situation? Obvious but impractical answers are: 'we check if the posterior is not concentrated' or 'we let Bayes learn both on our model, and the convex hull of it, and see if there is a difference'. More practical would be to have a method to deal with the problem automatically. We will encounter such a method in the coming sections.

The generalized posterior

Several authors have brought forward the idea of equipping Bayesian updating with a *learning rate* η, resulting in an *η-generalized posterior* (Vovk (1990), McAllester (2003), Seeger (2002), Catoni (2007), Audibert (2004), Zhang (2004)). Grünwald (2012) suggested its use as a method for dealing with misspecification, and proposed the *Safe Bayesian* algorithm for learning the 'right' η. In the η-generalized posterior (1.30), the likelihood is raised to the power η in order to trade off the relative weight of the likelihood and the prior, where $\eta = 1$ corresponds to standard Bayes.

Grünwald and van Ommen (2014), following Zhang (2004), McAllester (2003) and Catoni (2007), define the *generalized Bayesian posterior with learning rate η relative to loss functions ℓ*, denoted as $\Pi|Z^n, \eta$, formally as follows in Section 3 of their paper. We are given an abstract space of predictors represented by a set Θ, data Z^n, and we have a loss function $\ell : \mathcal{Z} \times \Theta \to \mathbb{R}$, and we will write $\ell_\theta(z) := \ell(z, \theta)$. For any prior Π on Θ with density π relative to some underlying measure ρ, $\Pi|Z^n, \eta$ is the distribution on Θ with density

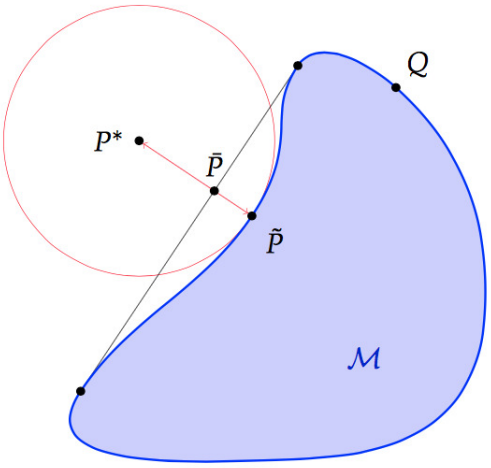


Figure 1.2: $\tilde{P}$ is the best approximation in the model $\mathcal{M}$ to the true distribution P^* in terms of KL divergence. Because the model is not convex, the Bayes predictive distribution $\overline{P}$ might be a mixture of distributions in the model, two in this picture, that ends up outside $\mathcal{M}$. In this case we risk 'bad' misspecification. If Q would be the best approximation in the model instead, the infimum $\inf_{P\in\mathcal{M}} D(P^* \parallel P)$ would not decrease if we would take the convex hull of $\mathcal{M}$, and we are in a 'good' case of misspecification. Picture from Grünwald and van Ommen (2014).

$$\begin{aligned}\pi(\theta|z^n,\eta) :&= \frac{e^{-\eta\sum_{i=1}^n \ell_\theta(z_i)}\pi(\theta)}{\int e^{-\eta\sum_{i=1}^n \ell_\theta(z_i)}\pi(\theta)\rho(d\theta)} \\ &= \frac{e^{-\eta\sum_{i=1}^n \ell_\theta(z_i)}\pi(\theta)}{\mathbf{E}_{\theta\sim\Pi}\, e^{-\eta\sum_{i=1}^n \ell_\theta(z_i)}}\,.\end{aligned} \tag{1.29}$$

As a special case we can take the model $\mathcal{M} = \{P_\theta|\theta \in \Theta\}$ for set Θ, data points $z_i = (x_i, y_i)$, and the log loss (1.13). We then obtain the following definition of the η-generalized posterior:

$$\pi(\theta|z^n,\eta) = \frac{(f(y^n|x^n,\theta))^\eta\pi(\theta)}{\int(f(y^n|x^n,\theta))^\eta\pi(\theta)\rho(d\theta)} = \frac{(f(y^n|x^n,\theta))^\eta\pi(\theta)}{\mathbf{E}_{\theta\sim\Pi}\left[(f(y^n|x^n,\theta))^\eta\right]}\,. \tag{1.30}$$

We can see from (1.30) that if $\eta = 1$, we get the standard posterior. If η gets smaller, the prior becomes more predominant over the data. We will see the instantiation of the generalized posterior to the lasso linear regression model in the next chapter.

Learning η: the Safe-Bayesian

Grünwald (2012) proposed the generalized posterior as a method to deal with misspecification. It is known (Grünwald, 2012) that in our 'bad' case of misspecification, Bayes will 'behave' again when η is chosen small enough. So, we have to choose an appropriate learning rate η. Question is: 'how small should it be?'. Too small an η leads to needlessly slow

convergence rates (with $\eta = 0$ one will not learn at all), but η must still be 'small enough'. It would be a very Bayesian idea to 'learn' η by integrating it out, but Grünwald and Langford (2007) show that this does not solve the problem. Grünwald (2012) and Grünwald and van Ommen (2014) propose the *Safe Bayesian* method to learn η from the data. The four versions of this algorithm will be explicated in the next chapter, Section 2.5, in the context of their implementation to the Bayesian lasso. Grünwald (2012) showed theoretically in several settings that the Safe Bayesian algorithm achieves good convergence rates in terms of KL divergence. Moreover, Grünwald and van Ommen (2014) empirically show that it performs excellently with simulated data in various (non-lasso) regression settings. In the coming chapters of this thesis we will investigate the Safe Bayesian algorithm for Bayesian lasso regression, both in experiments with a set-up similar to Grünwald and van Ommen (2014) and on 'real-world' data sets.

Chapter 2

The Safe Bayesian Lasso

In this chapter we explain the Bayesian lasso of Park and Casella (2008) and its implementation with a Gibbs sampler. Thereafter, we cover the η-generalized Bayesian lasso and several priors for the lasso parameter λ. Lastly we explain the Safe-Bayesian algorithm of Grünwald (2012) with its instantiation to the Bayesian lasso.

2.1 The Bayesian Lasso

Introduction

The Least Absolute Shrinkage and Selection Operator (LASSO) of Tibshirani (1996) is a regularization method that is used in regression problems for shrinkage and selection of features. The lasso is defined by taking the residual sum of squares from ordinary least squares regression, and adding a penalty for complexity. The lasso coefficients minimize the resulting expression. More precisely, they are solutions to:

$$\hat{\boldsymbol{\beta}}_{\text{lasso}} = \arg\min_{\boldsymbol{\beta}} \sum_i (y_i - x_i^T \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \,. \tag{2.1}$$

We can see from (2.1) that we impose an L_1 penalty on the features, additional to the least squares problem. This L_1 norm results in the *variable selection property* of the lasso. The minimum tends to be achieved for $\boldsymbol{\beta}$ with several coefficients set to zero. Consequently, one obtains a sparse solution (Hastie et al., 2009). With n data points and p variables, the lasso can even select at most $\min(n, p)$ non-zero coefficients. Since the lasso tends to set the non-important coefficients to zero, it can be seen as a *model selection method*. One can fit the full model, i.e. all coefficients, and the non-important coefficients will be removed automatically.

When we try to solve (2.1) for $\boldsymbol{\beta}$, we notice that $\hat{\boldsymbol{\beta}}_{\text{lasso}}$ has no closed form solution, except in the special case that the design matrix is orthonormal. The penalty parameter λ determines the amount of shrinkage. The higher the penalty, the more coefficients are shrunk towards and put to zero. The optimal λ is usually determined empirically, e.g. by some form of cross-validation. In the rest of this thesis we call the lasso as described above, with cross-validation to determine λ, the *basic lasso*.

The lasso coefficients have a Bayesian interpretation: they can be interpreted as the posterior mode estimate with independent identical Laplace priors on the coefficients (Park and Casella, 2008). The *Laplace distribution* or *double exponential distribution* is given by the probability density function:

$$f(x|\mu, b) = \frac{1}{2b} e^{-|x-\mu|/b}, \tag{2.2}$$

with mean μ and variance $2b^2$. Because of this Bayesian interpretation of the basic lasso, the past couple of years a variety of Bayesian variations of the lasso have been published. There is an issue with this Bayesian interpretation of the basic lasso: it is based on the posterior *mode*. One of the main reasons for the introduction of the lasso by Tibshirani (1996) was to improve prediction accuracy of regression models. In this thesis we focus on prediction as well. As we saw in Section 1.3, we can use the posterior *mean* for both point estimation and prediction. Predictions under square-loss are made via the mean of the posterior predictive distribution. But unlike the posterior mode, the posterior mean does not have the property of setting features exactly to zero. Still the posterior mean gives the optimal predictions for the square-loss, and therefore, as is standard in the Bayesian lasso literature as well, we will use the posterior mean to make predictions for the Bayesian lasso, although we lose the property of the basic lasso of yielding a sparse solution.

The Bayesian Lasso

In this thesis we concentrate on the Bayesian lasso of Park and Casella (2008), although various other Bayesian lasso's have been proposed recently. We will touch upon several of these versions when we discuss variable selection in Chapter 4. The Bayesian lasso of Park and Casella (2008) is constructed in the following way. Let $\tilde{\boldsymbol{y}}$ be $\boldsymbol{y} - \overline{\boldsymbol{y}}$. As stated in the previous section, with the prior

$$\pi(\boldsymbol{\beta}) = \prod_{j=1}^{p} \frac{\lambda}{2} e^{-\lambda|\beta_j|}, \tag{2.3}$$

and an independent prior $\pi(\sigma^2)$ on σ^2, the lasso estimate is the mode of the posterior distribution (Tibshirani, 1996):

$$\begin{aligned} &\pi(\boldsymbol{\beta}, \sigma^2|\tilde{\boldsymbol{y}}) \propto \\ &\pi(\sigma^2)(\sigma^2)^{-(n-1)/2} \exp\Big\{ \frac{1}{2\sigma^2}(\tilde{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{\beta})^T(\tilde{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{\beta}) - \lambda \sum_{j=1}^{p} |\beta_j| \Big\}. \end{aligned} \tag{2.4}$$

For a fixed variance, the posterior mode estimate coincides with a lasso estimate (2.1). But, from a Bayesian point of view, using the posterior mode is unfavourable: we should make predictions based on the Bayes predictive distribution, which in the regression model has mean of $(Y|X)$ given by the posterior *mean* parameters. As previously mentioned, the variable selection property of the lasso is lost when using the posterior mean, so the Bayesian lasso only performs shrinkage of the features towards zero, not setting them exactly to zero.

If σ^2 is not fixed, a posterior of the form (2.4) can have multiple modes, even if $\pi(\sigma^2)$ is proper (a proof of which is stated in the appendix of Park and Casella (2008)). To solve this problem, Park and Casella use a slightly modified prior with respect to (2.3):

$$\pi(\boldsymbol{\beta}|\sigma^2) = \prod_{j=1}^{p} \frac{\lambda}{2\sqrt{\sigma^2}} e^{-\lambda|\beta_j|/\sqrt{\sigma^2}} \,. \tag{2.5}$$

Conditioning on σ^2 guarantees unimodality of the posterior of β_j.

We will also use a prior on $\boldsymbol{\beta}$ of the form (2.5) together with an improper (Jeffreys') prior on σ^2: $\pi(\sigma^2) = 1/\sigma^2$, though we will allow for other priors on σ^2 later on. We can now formulate a hierarchical model, which we can use to implement this version of the Bayesian lasso with a Gibbs sampler, using a representation of the Laplace distribution as a scale mixture of Gaussians. Such a scale mixture simply modifies the tail of a Gaussian, by 'mixing' or 'averaging' the Gaussian over a mixing distribution. When the mixing distribution is exponential, the resulting distribution is Laplace (Andrews and Mallows, 1974):

$$\frac{a}{2} e^{-a|z|} = \int_0^\infty \frac{1}{\sqrt{2\pi s}} e^{-z^2/(2s)} \frac{a^2}{2} e^{-a^2 s/2} ds, \; a > 0 \,. \tag{2.6}$$

We use a latent parameter τ^2 to rewrite the prior (2.5) as a scale mixture of normals (2.6). A way to view the τ_j's is to think of them as additional parameters to a standard Bayesian regression model, that assign different weights to the columns of the design matrix $\boldsymbol{X}$. When $\tau_j \to 0$, the coefficient of the corresponding column of $\boldsymbol{X}$ is shrunk to zero.

We can now write the hierarchical model formulation of Park and Casella (2008) as follows.

$$\begin{aligned} \boldsymbol{y}|\mu, \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2 &\sim N(\mu + \boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}) \,, \\ \boldsymbol{\beta}|\tau_1^2, \ldots, \tau_p^2, \sigma^2 &\sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{D}_\tau), \quad \boldsymbol{D}_\tau = \mathrm{diag}(\tau_1^2, \ldots, \tau_p^2) \,, \\ \tau_1^2, \ldots, \tau_p^2 &\sim \prod_{j=1}^{p} \frac{\lambda^2}{2} e^{-\lambda^2 \tau_j^2/2} d\tau_j^2, \quad \tau_1^2, \ldots, \tau_p^2 > 0 \,, \\ \sigma^2 &\sim \pi(\sigma^2)\, d\sigma^2 \,. \end{aligned} \tag{2.7}$$

In this model formulation the μ on which the outcome variables $\boldsymbol{y}$ depend, is the overall mean, from which $\boldsymbol{X}\boldsymbol{\beta}$ are deviations. The parameter μ can be given a flat prior and subsequently integrated out, as we do in the coming sections.

2.2 The Gibbs sampler

Now we can implement the model (2.7) with a Gibbs sampler. The Gibbs sampler is a Markov Chain Monte Carlo (MCMC) algorithm that samples from the conditional distributions of a parameter given all other parameters. It does this for all parameters during one iteration. The hierarchical model (2.7) is constructed in such a way that we can formulate the full conditional distributions for each component of the estimate. This provides easy simulation, because the Gibbs sampler merely needs an expression proportional to the

joint distribution. Hence we can avoid having to deal with the normalization constant. The (normalized) posterior distribution is rather complicated and not explicitly known, nor can it be integrated to obtain a marginal distribution.

We will implement the model (2.7) just as Park and Casella (2008) do, and our implementation will also be used for the experiments in the coming chapters. For the time being we assume λ fixed, and we will touch upon the ways to determine or sample λ later, in Section 2.4. We use prior (2.5) on $\boldsymbol{\beta}$, an independent flat prior on μ, and we use the following inverse gamma prior on σ^2:

$$\pi(\sigma^2) = \frac{\gamma^a}{\Gamma(a)}(\sigma^2)^{-a-1}e^{-\gamma/\sigma^2}, \quad \sigma^2 > 0 \quad (a > 0, \gamma > 0)\,.$$

This prior has hyperparameters a, b. The limit of this prior for $a, b \to 0$ is equivalent to Jeffreys' prior: $\sigma \propto \sigma^{-(a+1)}e^{-b/\sigma} \to 1/\sigma$ as $a, b \to 0$.

Now we can write down the full joint density

$$\begin{aligned} f(\boldsymbol{y}|\mu, \boldsymbol{\beta}, \sigma^2)\,\pi(\sigma^2)\,\pi(\mu)\prod_{j=1}^{p}\pi(\boldsymbol{\beta}_j|\tau_j^2, \sigma^2)\,\pi(\tau_j^2) = \qquad (2.8)\\ \frac{1}{(2\pi\sigma^2)^{n/2}}\,e^{\frac{1}{2\sigma^2}(\boldsymbol{y}-\mu\mathbf{1}_n-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\mu\mathbf{1}_n-\boldsymbol{X}\boldsymbol{\beta})}\cdot\\ \frac{\gamma^\alpha}{\Gamma(\alpha)}(\sigma^2)^{-\alpha-1}e^{-\frac{\gamma}{\sigma^2}}\prod_{j=1}^{p}\frac{1}{(2\sigma^2\tau_j^2)^{1/2}}\,e^{-\frac{1}{2\sigma^2\tau_j^2}\boldsymbol{\beta}_j^2}\,\frac{\lambda^2}{2}\,e^{-\lambda^2\tau_j^2/2}\,. \end{aligned}$$

Let $\tilde{\boldsymbol{y}}$ be $\boldsymbol{y} - \overline{\boldsymbol{y}}$ again. We integrate out μ. The joint density marginal over μ is proportional to

$$\frac{1}{(\sigma^2)^{(n-1)/2}}\,e^{-\frac{1}{2\sigma^2}(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})^T(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})}\,(\sigma^2)^{-\alpha-1}\,e^{-\frac{\gamma}{\sigma^2}}\prod_{j=1}^{p}\frac{1}{(\sigma^2\tau_j^2)^{1/2}}\,e^{-\frac{1}{2\sigma^2\tau_j^2}\boldsymbol{\beta}_j^2}\,e^{-\lambda^2\tau_j^2/2}.$$

Since we integrate it out, the parameter μ itself is irrelevant. However, if we would like to include μ in the Gibbs sampler, we can easily see that it is normally distributed with mean $\overline{y}$ and variance σ^2/n, since $(\boldsymbol{y}-\mu\mathbf{1}_n-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\mu\mathbf{1}_n-\boldsymbol{X}\boldsymbol{\beta}) = n(\overline{y}-\mu)^2+(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})^T(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})$.

$\boldsymbol{\beta}$

Let us now construct the full conditional distribution for $\boldsymbol{\beta}$. Since the Gibbs sampler needs no more than an unnormalized posterior, we need nothing but the part of the joint distribution proportional to it that includes $\boldsymbol{\beta}$. When we eliminate the factors not involving $\boldsymbol{\beta}$ from (2.8), we are left with the following exponent terms:

$$
\begin{aligned}
&-\frac{1}{2\sigma^2}(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})^T(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})-\frac{1}{2\sigma^2}\boldsymbol{\beta}^T\boldsymbol{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\beta}\\
=&-\frac{1}{2\sigma^2}\left\{(\boldsymbol{\beta}^T(\boldsymbol{X}^T\boldsymbol{X})\boldsymbol{\beta}-2\tilde{\boldsymbol{y}}\boldsymbol{X}\boldsymbol{\beta}+\tilde{\boldsymbol{y}}^T\tilde{\boldsymbol{y}})+\boldsymbol{\beta}^T\boldsymbol{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\beta}\right\}\\
=&-\frac{1}{2\sigma^2}\left\{(\boldsymbol{\beta}^T(\boldsymbol{X}^T\boldsymbol{X}+\boldsymbol{D}_{\boldsymbol{\tau}}^{-1})\boldsymbol{\beta}-2\tilde{\boldsymbol{y}}\boldsymbol{X}\boldsymbol{\beta}+\tilde{\boldsymbol{y}}^T\tilde{\boldsymbol{y}})\right\}. \qquad (2.9)
\end{aligned}
$$

Now we will write $\boldsymbol{A}=\boldsymbol{X}^T\boldsymbol{X}+\boldsymbol{D}_{\boldsymbol{\tau}}^{-1}$, with which (2.9) reduces to

$$
-\frac{1}{2\sigma^2}\left\{\boldsymbol{\beta}^T\boldsymbol{A}\boldsymbol{\beta}-2\tilde{\boldsymbol{y}}\boldsymbol{X}\boldsymbol{\beta}+\tilde{\boldsymbol{y}}^T\tilde{\boldsymbol{y}}\right\}. \qquad (2.10)
$$

Hereafter, we can use the following square

$$
(\boldsymbol{\beta}-\boldsymbol{A}^{-1}\boldsymbol{X}^T\tilde{\boldsymbol{y}})^T\boldsymbol{A}(\boldsymbol{\beta}-\boldsymbol{A}^{-1}\boldsymbol{X}^T\tilde{\boldsymbol{y}})=\boldsymbol{\beta}\boldsymbol{A}^T\boldsymbol{\beta}-2\tilde{\boldsymbol{y}}\boldsymbol{X}\boldsymbol{\beta}+\tilde{\boldsymbol{y}}^T(\boldsymbol{X}\boldsymbol{A}^{-1}\boldsymbol{X}^T)\tilde{\boldsymbol{y}}.
$$

Accordingly, we can write (2.10) as

$$
-\frac{1}{2\sigma^2}\left\{(\boldsymbol{\beta}-\boldsymbol{A}^{-1}\boldsymbol{X}^T\tilde{\boldsymbol{y}})^T\boldsymbol{A}\,(\boldsymbol{\beta}-\boldsymbol{A}^{-1}\boldsymbol{X}^T\tilde{\boldsymbol{y}})+\tilde{\boldsymbol{y}}^T(\boldsymbol{I}_n-\boldsymbol{X}\boldsymbol{A}^{-1}\boldsymbol{X}^T)\tilde{\boldsymbol{y}}\right\}. \qquad (2.11)
$$

The second part of (2.11) does not rely on $\boldsymbol{\beta}$, hence all parts of the joint distribution (2.8) involving $\boldsymbol{\beta}$ are now reduced to the exponent of

$$
-\frac{1}{2\sigma^2}\left\{(\boldsymbol{\beta}-\boldsymbol{A}^{-1}\boldsymbol{X}^T\tilde{\boldsymbol{y}})^T\boldsymbol{A}\,(\boldsymbol{\beta}-\boldsymbol{A}^{-1}\boldsymbol{X}^T\tilde{\boldsymbol{y}})\right\}.
$$

Recall that the multivariate normal density of $\boldsymbol{X}\sim N(\mu,\boldsymbol{\Sigma})$ is given by

$$
f(\boldsymbol{X};\mu,\boldsymbol{\Sigma})=\frac{e^{-\frac{1}{2}(\boldsymbol{X}-\mu)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{X}-\mu)}}{(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}.
$$

Consequently we can conclude that $\boldsymbol{\beta}$ is distributed multivariate normal with mean $\boldsymbol{A}^{-1}\boldsymbol{X}^T\tilde{\boldsymbol{y}}$ and variance $\sigma^2\boldsymbol{A}^{-1}$.

$\boldsymbol{\sigma^2}$

The second variable to be sampled in the Gibbs sampler is σ^2, for which we will now construct the full conditional. The terms of the joint distribution (2.8) involving σ^2 are

$$
(\sigma^2)^{\{-(n-1)/2-p/2-\alpha-1\}}\exp\Big\{-\frac{1}{2\sigma^2}(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})^T(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})+\frac{1}{2\sigma^2}\boldsymbol{\beta}^T\boldsymbol{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\beta}+\frac{\gamma}{\sigma^2}\Big\}. \qquad (2.12)
$$

The inverse gamma density function is defined for $x > 0$, $x \sim \text{Inv-Gamma}(\alpha, \beta)$ by

$$f(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp\left\{-\frac{\beta}{x}\right\}, \tag{2.13}$$

with shape parameter α and scale parameter β. Thus, from (2.12) and (2.13) we can see straight away that σ^2 is conditionally inverse gamma with shape parameter $(n-1)/2+p/2+\alpha$ and scale parameter $(\tilde{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{\beta})^T(\tilde{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{\beta})/2 + \boldsymbol{\beta}^T \boldsymbol{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\beta}/2 + \gamma$.

$\boldsymbol{\tau_j^2}$

The third and last variable we have to sample is τ_j^2, for each $j = 1 \dots p$. The part of the joint distribution (2.8) including τ_j^2 is

$$(\tau_j^2)^{-1/2} \exp\left\{-\frac{1}{2}\left(\frac{\beta_j^2/\sigma^2}{\tau_j^2} + \lambda^2\tau_j^2\right)\right\}. \tag{2.14}$$

The Inverse-Gaussian (IG) distribution, also known as the Wald distribution, is given by

$$f(x; \mu, \lambda') = \left(\frac{\lambda'}{2\pi x^3}\right)^{1/2} \exp\left\{\frac{-\lambda'(x-\mu)^2}{2\mu^2 x}\right\}, \tag{2.15}$$

for $x > 0$ and with mean parameter $\mu > 0$ and shape parameter $\lambda' > 0$. Section 4.4 from Chhikara and Folks (1989) concerns the distribution of the reciprocal of an inverse Gaussian variable. When a variable is distributed according to an inverse Gaussian distribution: $X \sim IG(\mu, \lambda')$, the distribution of its inverse, f' of $W = X^{-1}$ is given by (equation 4.6 from Chhikara and Folks (1989))

$$f'(w; \mu, \lambda') = \left(\frac{\lambda'}{2\pi w}\right)^{1/2} \exp\left\{\frac{-\lambda'(1-\mu w)^2}{2\mu^2 w}\right\}, \quad w > 0\,. \tag{2.16}$$

Equivalently (equation 4.7 from Chhikara and Folks (1989))

$$f'(w; \mu, \lambda') = \mu w f(w; \mu^{-1}, \lambda'\mu^{-2})\,.$$

To mold (2.14) into the form of the reciprocal of the Inverse-Gaussian distribution (2.16), we can rewrite it as

$$\begin{aligned}
&\left(\frac{1}{\tau_j^2}\right)^{-3/2} \exp\left\{-\frac{1}{2}\left(\frac{\beta_j^2}{\sigma^2}\frac{1}{\tau_j^2} + \frac{\lambda^2}{1/\tau_j^2}\right)\right\} \\
&\propto \left(\frac{1}{\tau_j^2}\right)^{-3/2} \exp\left\{-\frac{\beta_j^2\left((\frac{1}{\tau_j^2}) - \sqrt{\lambda^2\sigma^2/\beta^2}\right)^2}{2\sigma^2(1/\tau_j^2)}\right\}.
\end{aligned}$$

As a result we can see that $1/\tau_j^2$ is distributed according to (2.15), inverse Gaussian, with shape parameter $\lambda' = \lambda^2$ and mean parameter

$$\mu = \sqrt{\frac{\lambda^2\sigma^2}{\beta_j^2}}.$$

Summarizing, we can implement a Gibbs sampler with the following full conditional distributions. In fact we do not need μ, since we marginalized over it, but we can include it nonetheless if we want to.

$$\mu \sim \mathrm{N}(\overline{y},\, \sigma^2/n)\,, \tag{2.17}$$

$$\beta \sim \mathrm{N}\left((\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{D}_{\boldsymbol{\tau}}^{-1})^{-1}\boldsymbol{X}^T\tilde{\boldsymbol{y}},\, \sigma^2(\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{D}_{\boldsymbol{\tau}}^{-1})^{-1}\right)\,, \tag{2.18}$$

$$\sigma^2 \sim \text{Inv-Gamma}\big((n-1)/2 + p/2 + \alpha,\, (\tilde{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{\beta})^T(\tilde{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{\beta})/2 + \boldsymbol{\beta}^T\boldsymbol{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\beta}/2 + \gamma\big)\,, \tag{2.19}$$

$$\frac{1}{\tau_j^2} \sim \mathrm{IG}\left(\sqrt{\frac{\lambda^2\sigma^2}{\beta_j^2}},\, \lambda^2\right)\,. \tag{2.20}$$

2.3 The generalized posterior

The analogue of the Gibbs sampler for an η-generalized Bayesian lasso can be formulated as follows. With the hierarchy of (2.7) the joint density with a likelihood to the power η becomes

$$f(\boldsymbol{y}|\mu,\boldsymbol{\beta},\sigma^2,\eta)\,\pi(\sigma^2)\,\pi(\mu)\prod_{j=1}^{p}\pi(\boldsymbol{\beta}_j|\tau_j^2,\sigma^2)\,\pi(\tau_j^2) = \tag{2.21}$$
$$\left(\frac{1}{(2\pi\sigma^2)^{n/2}}\, e^{\frac{1}{2\sigma^2}(\boldsymbol{y}-\mu\mathbf{1}_n-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\mu\mathbf{1}_n-\boldsymbol{X}\boldsymbol{\beta})}\right)^{\eta}$$
$$\frac{\gamma^\alpha}{\Gamma(\alpha)}(\sigma^2)^{-\alpha-1}e^{-\frac{\gamma}{\sigma^2}}\prod_{j=1}^{p}\frac{1}{(2\sigma^2\tau_j^2)^{1/2}}\, e^{-\frac{1}{2\sigma^2\tau_j^2}\boldsymbol{\beta}_j^2}\frac{\lambda^2}{2}\, e^{-\lambda^2\tau_j^2/2}\,.$$

Let $\tilde{\boldsymbol{y}}$ again be $\boldsymbol{y} - \overline{\boldsymbol{y}}$. If we integrate out μ, the joint density marginal over μ is proportional to

$$\left(\frac{1}{(\sigma^2)^{(n-1)/2}}\right)^{\eta} e^{-\frac{\eta}{2\sigma^2}(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})^T(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})}\,(\sigma^2)^{-\alpha-1}\,e^{-\frac{\gamma}{\sigma^2}}\prod_{j=1}^{p}\frac{1}{(\sigma^2\tau_j^2)^{1/2}}\, e^{-\frac{1}{2\sigma^2\tau_j^2}\boldsymbol{\beta}_j^2}\, e^{-\lambda^2\tau_j^2/2}. \tag{2.22}$$

$\boldsymbol{\beta}$

We can lift out of (2.22) the exponent terms involving $\boldsymbol{\beta}$

$$\begin{aligned}
&-\frac{\eta}{2\sigma^2}(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})^T(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})-\frac{1}{2\sigma^2}\boldsymbol{\beta}^T\boldsymbol{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\beta}\\
=&-\frac{1}{2\sigma^2}\left\{\eta(\boldsymbol{\beta}^T(\boldsymbol{X}^T\boldsymbol{X})\boldsymbol{\beta}-2\tilde{\boldsymbol{y}}\boldsymbol{X}\boldsymbol{\beta}+\tilde{\boldsymbol{y}}^T\tilde{\boldsymbol{y}})+\boldsymbol{\beta}^T\boldsymbol{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\beta}\right\}\\
=&-\frac{1}{2\sigma^2}\left\{(\boldsymbol{\beta}^T(\eta\boldsymbol{X}^T\boldsymbol{X})\boldsymbol{\beta}-2\eta\tilde{\boldsymbol{y}}\boldsymbol{X}\boldsymbol{\beta}+\eta\tilde{\boldsymbol{y}}^T\tilde{\boldsymbol{y}})+\boldsymbol{\beta}^T\boldsymbol{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\beta}\right\}\\
=&-\frac{1}{2\sigma^2}\left\{(\boldsymbol{\beta}^T(\eta\boldsymbol{X}^T\boldsymbol{X}+\boldsymbol{D}_{\boldsymbol{\tau}}^{-1})\boldsymbol{\beta}-2\eta\tilde{\boldsymbol{y}}\boldsymbol{X}\boldsymbol{\beta}+\eta\tilde{\boldsymbol{y}}^T\tilde{\boldsymbol{y}})\right\}.
\end{aligned}$$

If we now write $\boldsymbol{A}=\eta\boldsymbol{X}^T\boldsymbol{X}+\boldsymbol{D}_{\boldsymbol{\tau}}^{-1}$ we arrive at the expression

$$-\frac{1}{2\sigma^2}\left\{(\boldsymbol{\beta}^T(\boldsymbol{A})\boldsymbol{\beta}-2\eta\tilde{\boldsymbol{y}}\boldsymbol{X}\boldsymbol{\beta}+\eta\tilde{\boldsymbol{y}}^T\tilde{\boldsymbol{y}})\right\}.$$

Lastly, we can complete the square just as in (2.11) and achieve

$$-\frac{1}{2\sigma^2}\left\{(\boldsymbol{\beta}-\eta\boldsymbol{A}^{-1}\boldsymbol{X}^T\tilde{\boldsymbol{y}})^T\boldsymbol{A}\,(\boldsymbol{\beta}-\eta\boldsymbol{A}^{-1}\boldsymbol{X}^T\tilde{\boldsymbol{y}})+\tilde{\boldsymbol{y}}^T(\eta\boldsymbol{I}_n-\eta^2\boldsymbol{X}\boldsymbol{A}^{-1}\boldsymbol{X}^T)\tilde{\boldsymbol{y}}\right\}.$$

Accordingly we can see that $\boldsymbol{\beta}$ is conditionally multivariate normal with mean $\eta\boldsymbol{A}^{-1}\boldsymbol{X}^T\tilde{\boldsymbol{y}}$ and variance $\sigma^2\boldsymbol{A}^{-1}$.

$\boldsymbol{\sigma^2}$

The terms from the joint density with the likelihood to the power η marginal over μ that involve σ^2 are:

$$(\sigma^2)^{\{-\eta(n-1)/2-p/2-\alpha-1\}}\cdot \exp\Big\{-\frac{\eta}{2\sigma^2}(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})^T(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})+\frac{1}{2\sigma^2}\boldsymbol{\beta}^T\boldsymbol{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\beta}+\frac{\gamma}{\sigma^2}\Big\}.$$

We can conclude that σ^2 is conditionally inverse gamma with shape parameter $\eta\,\frac{n-1}{2}+\frac{p}{2}+\alpha$ and scale parameter $\frac{\eta}{2}(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})^T(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})+\boldsymbol{\beta}^T\boldsymbol{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\beta}/2+\gamma$.

Since τ_j^2 is not involved in the likelihood, we need not modify the implementation of it. Summarizing, we can implement a Gibbs sampler with the following distributions:

$$\boldsymbol{\beta}\sim \mathrm{N}\left(\eta(\eta\boldsymbol{X}^T\boldsymbol{X}+\boldsymbol{D}_{\boldsymbol{\tau}}^{-1})^{-1}\boldsymbol{X}^T\tilde{\boldsymbol{y}},\,\sigma^2(\eta\boldsymbol{X}^T\boldsymbol{X}+\boldsymbol{D}_{\boldsymbol{\tau}}^{-1})^{-1}\right), \tag{2.23}$$

$$\sigma^2\sim \text{Inv-Gamma}\Big(\frac{\eta}{2}(n-1)+p/2+\alpha,\ \frac{\eta}{2}(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})^T(\tilde{\boldsymbol{y}}-\boldsymbol{X}\boldsymbol{\beta})+\boldsymbol{\beta}^T\boldsymbol{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\beta}/2+\gamma\Big), \tag{2.24}$$

$$\frac{1}{\tau_j^2}\sim \mathrm{IG}\left(\sqrt{\frac{\lambda^2\sigma^2}{\beta_j^2}},\,\lambda^2\right). \tag{2.25}$$

2.4 The lasso parameter λ

There are several ways to determine or sample the shrinkage parameter λ. In the basic version of the lasso, usually some form of cross-validation is performed to determine the parameter from the data empirically. In the Bayesian setting, we could use *empirical Bayes* by marginal maximum likelihood and use an EM algorithm to complement the Gibbs sampler (Park and Casella, 2008). For this thesis, we will focus on a 'pure' Bayesian approach: placing a hyperprior on the parameter without involving the data. We will provide three different ways to do so: a point mass (resulting in a fixed λ), a gamma prior on λ^2 following Park and Casella (2008) and a beta prior following de los Campos et al. (2009), using the `R`-implementation for the Metropolis-Hastings algorithm of the latter. de los Campos et al. (2009) also tested the performance of the Bayesian lasso regression of Park and Casella (2008) with several gamma and beta priors for λ on genome wide marker data. As Park and Casella (2008) note, the hyperprior for λ must be chosen carefully. Where the λ parameter in the basic lasso controls a trade-off between model complexity and goodness of fit, in the Bayesian version it controls the shape of the prior for τ_j^2 as can be seen from (2.25). A proper prior is necessary to avoid mixing problems, and bimodality and non-integrability in the posterior.

The gamma prior

Park and Casella (2008) assign a gamma prior to λ^2, which is more convenient than defining a prior on λ. The gamma distribution for $x > 0$ is given by

$$f(x; k, \theta) = x^{k-1} \frac{\theta^k e^{-\theta x}}{\Gamma(k)}, \qquad \Gamma(k) = (k-1)! \tag{2.26}$$

The factors of the joint density (2.8) together with the gamma prior (2.26) on λ^2 that include λ are

$$\begin{aligned} &\left(\prod_{j=1}^{p} \frac{\lambda^2}{2} e^{-\lambda^2 \tau^2 / 2} \right) \left(\lambda^2\right)^{k-1} e^{-\theta \lambda^2} \\ &= \left(\lambda^2\right)^{p+k-1} \exp\left\{ -\lambda^2 \left(\frac{1}{2} \sum_{j=1}^{p} \tau_j^2 + \theta \right) \right\}. \end{aligned} \tag{2.27}$$

This is again a gamma distribution with shape parameter $p + r$ and rate parameter $\sum_{i=1}^{p} \tau_j^2/2$, which is the full conditional for λ. It can be included in the Gibbs sampler as such, with two notes of caution. To begin with, the parameter θ must be sufficiently large (refer for details to Park and Casella (2008)), and secondly, the prior should be relatively flat to reduce bias and place sufficient mass near the (unknown) maximum likelihood estimate.

The beta prior

To provide more flexibility in choosing a relatively flat prior for λ, de los Campos et al. (2009) considered a beta distribution

$$f(x;\alpha,\beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \quad \alpha,\, \beta > 0\,,$$
$$\text{with} \quad B(x,y) = \int_0^1 t^{x-1}(1-t)^{y-1}dt, \quad re(x),\, re(y) > 0\,.$$

They construct it such a way that with an extra 'maximum' parameter, the prior on λ follows a Beta-distribution

$$\pi(\lambda|\alpha_1,\alpha_2,\max) \propto \text{Beta}\left(\frac{\lambda}{\max}|\alpha_1,\alpha_2\right). \tag{2.28}$$

However, the conditional distribution of λ (combining (2.27) and (2.28)) does not have a closed form, so we cannot sample from the full conditional distribution as we did before with the other parameters. Instead we can use a Metropolis-Hastings algorithm, an algorithm that makes use of an acceptance/rejection rule to make draws from the desired distribution. We can include it in the sampling by alternating the Gibbs sampler for the other variables with a one-dimensional Metropolis-jump for λ.

2.5 Learning η: the Safe-Bayesian

We will implement several algorithms to learn the learning parameter η from the data for the generalized Bayesian lasso. K-fold cross-validation is an obvious option, both with squared error loss and log-loss. However, Grünwald and van Ommen (2014) note that in their model-wrong experiments, to which we return in Chapter 3, leave-one-out cross-validation with square-loss worked fine, as it is similar to the I-square-SafeBayes version, but behaved terribly with log-loss of the Bayes predictive distribution.

Besides k-fold cross-validation, we implement the four versions of SafeBayes from Grünwald and van Ommen (2014): R-square-SafeBayes and I-square-SafeBayes for models $\mathcal{M}^{(p)}$ with fixed variance and R-log-SafeBayes and I-log-SafeBayes for models $\mathcal{S}^{(p)}$ with varying variance.

The R-Safe-Bayesian algorithm

In the *randomized* versions of the Safe Bayesian algorithm, we obtain many random draws from the generalized conditional posterior distributions for β and σ^2, calculated on data points z^i. With these draws from the posterior, a prediction for the next data-point z_{i+1} is made and a loss with respect to it is calculated. The losses for all the random draws are averaged at each i, and these average losses are summed up over each data point. We have now obtained the *cumulative Posterior-Expected Posterior-Randomized log-loss* (2.29). We repeat this for a grid of learning rates η, and choose the $\hat{\eta}$ that minimizes this loss.

$$\sum_{i=1}^{n} \mathbf{E}_{\beta,\sigma^2\sim\Pi|z^{i-1},\eta}\left[-\log f(y_i|x_i,\beta,\sigma^2)\right] \tag{2.29}$$

Minimizing (2.29) comes down to minimizing (2.30) in this setting (see (1.14)). The loss between the brackets is averaged over many draws of $(\beta_{i,\eta}, \sigma^2_{i,\eta})$ from the posterior, where $\beta_{i,\eta}$ (or $\sigma^2_{i,\eta}$) denotes one random draw from the conditional η-generalized posterior based on data points z^i.

$$\sum_{i=1}^{n} \text{AV} \left[\frac{1}{2} \log 2\pi\sigma^2_{i,\eta} + \frac{1}{2} \frac{(y_{i+1} - x_{i+1}\beta_{i,\eta})^2}{\sigma^2_{i,\eta}} \right] \tag{2.30}$$

When we take σ^2 fixed, minimizing (2.30) over η reduces to minimizing the squared loss $(y_{i+1} - x_{i+1}\beta_{i,\eta})^2$. We calculate this loss by randomly drawing many times from the generalized conditional posterior distribution of β and averaging the results. This is called *R-square-SafeBayes*. When we are in the more usual situation with varying σ^2, we calculate (2.30) by randomly drawing β and σ^2 from the same MCMC iteration. This version is called *R-log-SafeBayes*. For a clear overview, Algorithm 1 is copied from Grünwald and van Ommen (2014).

Algorithm 1: The R-Safe-Bayesian algorithm

Input : data $z_1, \ldots z_n$, model $\mathcal{M} = \{f(\cdot|\theta)|\theta \in \Theta\}$, prior Π on Θ, step-size $\mathcal{K}_{\text{STEP}}$, max. exponent $\mathcal{K}_{\text{MAX}}$, loss function $\ell_\theta(z)$

Output: Learning rate $\hat{\eta}$

$\mathcal{S}_n := \{1, 2^{-\mathcal{K}_{\text{STEP}}}, 2^{-2\mathcal{K}_{\text{STEP}}}, 2^{-3\mathcal{K}_{\text{STEP}}}, \ldots, 2^{-\mathcal{K}_{\text{MAX}}}, \}$;

for *all* $\eta \in \mathcal{S}_n$ **do**

  $s_\eta := 0$;

  for $i = 1 \ldots n$ **do**

    Determine generalized posterior $\Pi(\cdot|z^{i-1}, \eta)$ of Bayes with learning rate η.

    Calculate posterior-expected posterior-randomized loss of predicting actual next outcome:

$$r := \ell_{\Pi|z^{i-1},\eta}(z_i) = \mathbf{E}_{\theta \sim \Pi|z^{i-1},\eta} [\ell_\theta(z_i)] \tag{2.31}$$

    $s_\eta := s_\eta + r$;

  end

end

Choose $\hat{\eta} := \arg\min_{\eta \in \mathcal{S}_n} \{s_\eta\}$ (if min achieved for several $\eta \in \mathcal{S}_n$, pick largest) ;

The I-Safe-Bayesian algorithm

Grünwald and van Ommen (2014) define the *η-in-model-log-loss* or *η-I-log-loss* as:

$$\sum_{i=1}^{n} \left[-\log f(y_i|x_i, \mathbf{E}_{\boldsymbol{\beta},\sigma^2 \sim \Pi|z^{i-1},\eta} \left[\boldsymbol{\beta}, \sigma^2\right]) \right] \tag{2.32}$$

The I-log-loss can be used in a variation on the Safe-Bayesian algorithm because of the following. We have for the posterior-expected posterior-randomized log-loss (2.29) for fixed η and for the addition of each data point i:

$$\mathbf{E}_{\boldsymbol{\beta},\sigma^2 \sim \Pi|z^{i-1},\eta} \left[-\log f(y_i|x_i, \boldsymbol{\beta}, \sigma^2) \right] \geq -\log f(y_i|x_i, \mathbf{E}_{\boldsymbol{\beta},\sigma^2 \sim \Pi|z^{i-1},\eta} \left[\boldsymbol{\beta}, \sigma^2\right]) . \tag{2.33}$$

This is true by Jensen's inequality (1.23) because we have:

$$\mathbf{E}_{\boldsymbol{\beta},\sigma^2\sim\Pi|z^{i-1},\eta}\left[-\log f(y_i|x_i,\boldsymbol{\beta},\sigma^2)\right] = \mathbf{E}_{\boldsymbol{\beta},\sigma^2}\left[\frac{1}{2}\log 2\pi\sigma^2 + \frac{(y-\boldsymbol{X}\boldsymbol{\beta})^2}{\sigma^2}\right]$$

and the negative log-likelihood of a linear model (1.14) is convex in its parameters.

In words, in the in-model version of SafeBayes, we calculate the generalized conditional posterior distributions for $\boldsymbol{\beta}$ and σ^2 for data points z^i. Subsequently, we predict the next data point z_{i+1} according to the posterior means obtained and calculate a loss. We sum again the losses for the sequential addition of each data point, repeat this for a grid of learning rates η and choose the $\hat{\eta}$ that minimizes this cumulative loss.

For fixed σ^2 minimizing the in-model-loss $-\log f(y_{i+1}|x_{i+1},\overline{\beta}_{i,\eta},\sigma^2)$ reduces again to minimizing the square-loss $(y_{i+1}-x_{i+1}\overline{\beta}_{i,\eta})^2$, and we call this version therefore *I-square-SafeBayes*. Here $\overline{\beta}_{i,\eta}$ denotes the average of many $\beta_{i,\eta}$, i.e. the posterior mean. For varying σ^2 we calculate the following loss according to the posterior means $\overline{\beta}$ and $\overline{\sigma}^2$:

$$\sum_{i=1}^{n}\left[\frac{1}{2}\log 2\pi\overline{\sigma}^2_{i,\eta} + \frac{1}{2}\frac{(y_{i+1}-x_{i+1}\overline{\beta}_{i,\eta})^2}{\overline{\sigma}^2_{i,\eta}}\right] \tag{2.34}$$

This last version is called *I-log-SafeBayes*.

Chapter 3

Bayesian Inconsistency: Experiments and Explanation

In this chapter we perform several experiments on simulated data. The first section contains an introduction to the set-up of the experiments and the error measure we use to quantify the predictive performance of our models. In Section 3.2 we start with an initial experiment to observe the problem that is the main theme in this thesis: the Bayesian lasso shows considerable overfitting when the model is *misspecified*. On the contrary, the Safe-Bayesian lasso appears to be close to having discovered the true regression function. Surprisingly, we sometimes see a weaker version of this phenomenon occur when the model is *well-specified*. We take a closer look at this phenomenon with an experiment with a different set-up. In Section 3.3 we compare the empirical square-risk of the standard Bayesian lasso and the Safe-Bayesian lasso, both when the model is misspecified and when it is well-specified. In the case that the model is misspecified, we observe that the empirical square-risk of the standard Bayesian lasso is not only larger than that of the Safe-Bayesian lasso, it grows with the sample size. Bayes recovers slowly when the sample size is larger than the dimensionality of the model. SafeBayes performs excellently in all experiments. An explanation for the behaviour of standard Bayes is given in Section 3.4. We find a discrepancy between the good log-risk (Barron's bound (Barron, 1998)) and bad square-risk (our experiments), which can be explained by the predictive distribution being a mixture of different 'bad' distributions in the model. Consequently, the discrepancy between good log-risk and bad square-risk implies that substantially many components of the predictive distribution must be substantially different from every distribution in the model, hence the posterior is not concentrated.

3.1 Preparation

We perform several experiments on simulated data. Since we generate the data ourselves, we know what the *truth*, P^* is. All the experiments are performed on data from a surprisingly simple sampling distribution and some variations on it, similar to, but slightly different from Grünwald and van Ommen (2014).

Correct-Model

In what we call the *correct-model* experiments the X_i are sampled i.i.d. from a uniform distribution on $[-1, 1]$. For every X_i an Y_i is sampled from a normal distribution $N(0, \sigma^2)$

with some fixed variance σ^2. The true conditional distribution $P^*(Y|X)$ is essentially 'zero with some Gaussian noise'.

Wrong-Model

For the *wrong-model* experiments, we first generate correct-model data as before. Then we toss a fair coin for each data point. If the coin lands heads, we use (X_i, Y_i) from the correct-model, but if the coin lands tails, we put $(X_i, Y_i) := (0, 0)$. The true distribution $P^*(Y|X)$ is still 'zero with some Gaussian noise', but approximately half of the points are *easy*: they lie right on the true regression function $\mathbf{E}_{P^*}[Y|X] = 0$ — without noise. When we want to compare the correct-model and wrong-model experiments, we give the non-easy data points in the wrong-model twice the variance of that of the data points in the correct-model, so that the marginal variance for each Y_i is the same and the experiments become comparable.

Risk

Since our focus is on prediction, we need an error measure to quantify the quality of our model's predictions. For linear models in general we are predominantly interested in finding coefficients $\boldsymbol{\beta}$ for which the model predicts well with regard to the *squared error loss function* or *square-loss*: $\ell(X_i, Y_i) = (Y_i - X_i\boldsymbol{\beta})^2$, the Euclidean distance between the observed and predicted values. Therefore we will run simulations to empirically determine the expected square-loss: the *square-risk*. The square-risk of $\tilde{P}$ relative to the underlying distribution P^* is:

$$\mathrm{RISK}^{\mathrm{sq}}(\boldsymbol{\beta}) := \mathbf{E}_{(X,Y)\sim P^*}(Y - \mathbf{E}_{Y\sim\tilde{P}_{(\boldsymbol{\beta},\sigma^2)}|X}[Y])^2 = \mathbf{E}_{(X,Y)\sim P^*}(Y - \sum_{j=0}^{p}\beta_j X_j)^2. \quad (3.1)$$

3.2 The problem: an initial experiment

Initially we observe what happens with the Bayesian lasso on wrong-model data. To this end we sample the posterior of the Bayesian lasso with a 201-dimensional Fourier basis to 100 wrong-model data points with standard improper priors on the variance σ^2 and λ (see Chapter 2). Since the true model P^* is essentially 'zero with some noise', we expect that the Bayesian lasso will 'choose' (put high mass on) values close to zero for every coefficient β_j. However, when we take a look at the resulting plot in Figure 3.1, we see that it chooses a model with considerably large coefficients for a variety of sines and cosines. It appears to overfit worryingly; it learns the noise instead of the signal. Hereafter, on the same data we let SafeBayes learn $\hat{\eta}$ and sample the corresponding generalized Bayesian lasso. SafeBayes picks a model with all coefficients very close to zero[1], and from the plot it seems to be close to having discovered the true regression function.

Our model is *misspecified*: it uses the assumption of homoskedasticity, while the underlying distribution of the data, the wrong-model P^*, is heteroskedastic: the variance σ^2 is not independent from X. The distribution $\tilde{P}$ in our model that would be closest to P^* in KL divergence, would have all $\beta_j = 0$ and $\tilde{\sigma}^2 = \sigma^2/2$. The problem is caused by the *in-liers* $(X_i, Y_i) = (0, 0)$. In the linear model experiments from Grünwald and van Ommen

[1] To prevent numerical problems in the sampling algorithm, a lower bound of $1 \cdot 10^{-9}$ is imposed on each $|\beta_j|$, preventing the algorithm from setting coefficients exactly to zero.

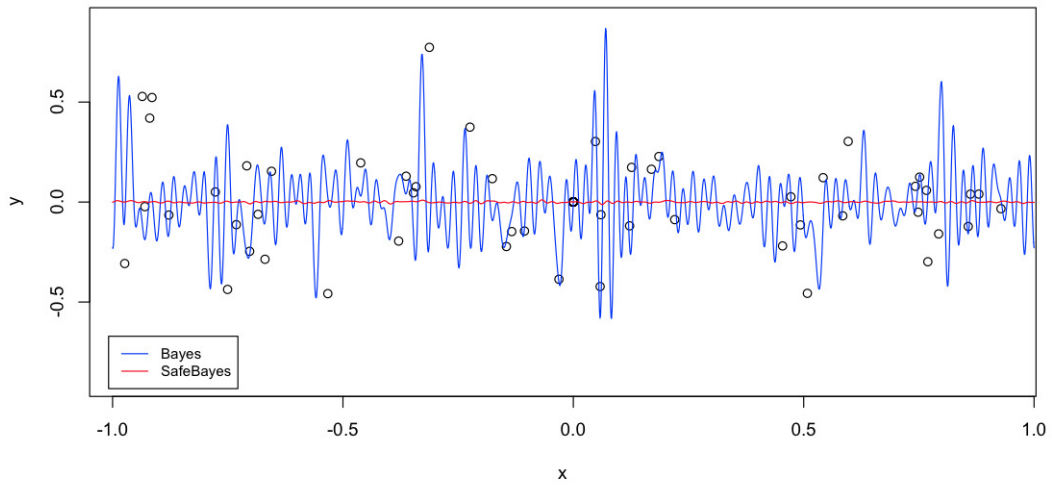


Figure 3.1: Conditional expectations $\mathbf{E}[Y \mid X]$ according to the posteriors of the standard Bayesian lasso (blue) and the (R-log-)Safe-Bayesian lasso (red) with 201 Fourier basis functions in the *wrong-model* experiment, based on 100 data points i.i.d. $\sim P^*$ of which approximately half put to $(0, 0)$. Bayes seems to overfit, whereas SafeBayes (with $\eta = 0.5$) appears to approximate the true regression function.

(2014), the Bayesian lasso behaves excellently again when applied to a somewhat modified set-up in a way that the in-liers are no longer present and homoskedasticity is restored. While Grünwald and van Ommen (2014) get essentially the same results for polynomial basis functions and independent multivariate X's, the Bayesian lasso shows its 'bad' behaviour not with those, but solely with a Fourier basis. We give a possible explanation for this phenomenon in Section 3.4.

The standard Bayesian lasso demonstrates its overfitting for different dimensions of the Fourier basis as well, including considerably low dimensions. We also examined Bayes' performance with priors from de los Campos et al. (2009), and the choice of vague prior did not influence its 'bad' behaviour. SafeBayes performs excellently again in these cases. See Appendix A for additional figures.

First surprise: correct-model misbehaviour

Interestingly, SafeBayes sometimes outperforms standard Bayes as well in the *correct-model* experiments, though on a smaller scale than in the wrong-model experiments. We had not expected this to happen. In Figure 3.2 we see predictions of both Bayes and SafeBayes, with a 201-dimensional Fourier basis and a Beta$(1.4, 1.4)$ prior on λ, $\lambda_{\text{MAX}} = 100$. In Appendix A four figures of the same correct-model set-up are added. Bayes' behaviour in the wrong-model set-up can be explained by variance issues (see Section 1.4 and Section 3.4). In the correct model, Bayes seems to behave worse when the variance of the data looks less evenly spread, i.e. when a weaker version of the misspecification problem occurs.

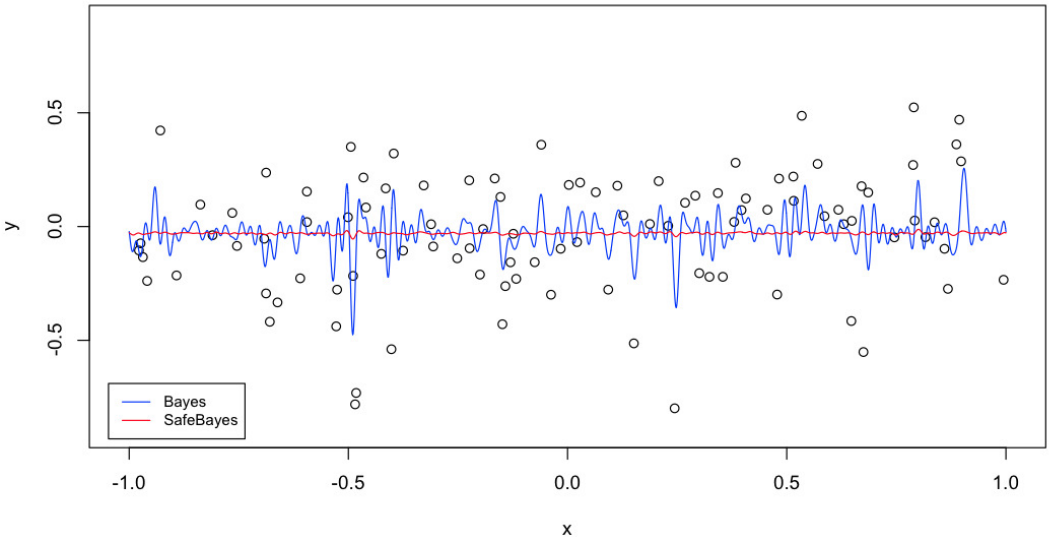


Figure 3.2: Conditional expectations $\mathbf{E}[Y \mid X]$ according to the standard Bayesian lasso (blue) and the (R-log-)Safe-Bayesian lasso (red) posteriors in the *correct-model* experiment, based on 100 data points i.i.d. $\sim P^*$, a Beta$(1.4, 1.4)$ prior on λ, $\lambda_{\text{MAX}} = 100$, with 201 Fourier basis functions.

In Figure 3.3 we see the same phenomenon. Additionally we see the consequence of a problem pointed out earlier: the model selection capacity of the lasso is diminished in its fully Bayesian form (see Section 2.1). The set-up for the experiment in Figure 3.3 is as follows. First we perform a *correct-model* experiment. The 400 X_i are sampled random uniform on $[-2\pi, 2\pi]$. The 'true' conditional distribution is $y_i = \sin(x_i) + \epsilon$, where ϵ is Gaussian noise as before. We sample the posteriors of the standard Bayesian lasso and the generalized Bayesian lasso (with $\eta = 0.2$) with a 201-dimensional Fourier basis and standard improper priors. In the correct model we observe overfitting for both standard and generalized Bayes, however, the overfitting of standard Bayes is worse than that of generalized Bayes. This is confirmed by predictions for new data (see Table 1 in Appendix A). Secondly, we sample 400 data points as before, but with twice as much variance of the Gaussian noise. We select approximately half of the data points randomly, and set them to $(0, 0)$. When we sample the posterior of the generalized Bayesian lasso to this *wrong-model* data, we see only a slight difference with the correct-model fit. The standard Bayesian lasso however, overfits not only worse than the generalized Bayesian lasso, but also to greater extent than its correct-model fit. While the generalized Bayesian lasso performs better in both cases, it keeps overfitting and yielding suboptimal predictions. In the first place, this issue could arise due to the Bayesian lasso's model selection problems. We look at those in Chapter 4. Secondly, we did not use SafeBayes to learn η, and it might be that the predictions of SafeBayes are optimal, although it might not have learnt the true model.

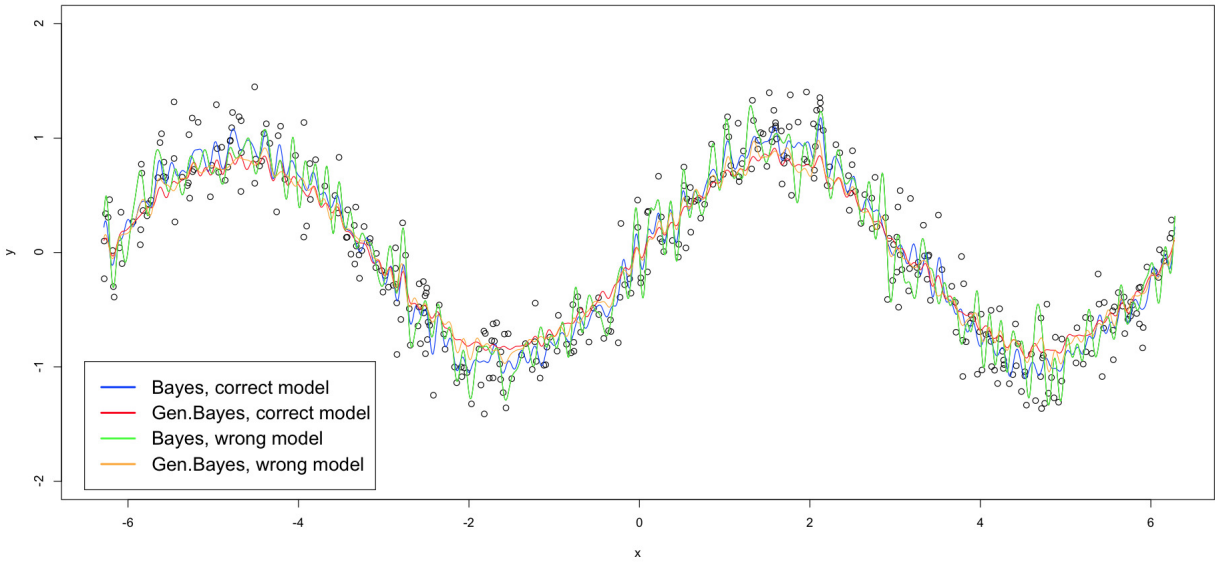


Figure 3.3: Conditional expectations $\mathbf{E}[Y \mid X]$ according to the standard Bayesian lasso and the generalized Bayesian lasso (with $\eta = 0.2$) full posteriors with with a 201-dimensional Fourier basis. In the correct model, X_i are sampled random uniform on $[-2\pi, 2\pi]$, and $y_i = \sin(x_i) + \epsilon$ with $\epsilon \sim N(0, 1/4)$. In the wrong-model, the data are as in the correct model with twice the variance of the Gaussian noise and with approximately half of the points set to $(0, 0)$.

3.3 Main experiments

Here we present the results of the main risk-experiments based on the Bayesian and Safe-Bayesian lasso regression with Fourier bases on data described in Section 3.1. We work with the varying-variance case (a prior on σ^2) and therefore use R-log-SafeBayes and I-log-SafeBayes. For each sample size ($n = 30$ to $n = 200$ in steps of 10), the mean risk of six experiments is taken.

From Figure 3.4 we observe that standard Bayes not only performs worse than both versions of SafeBayes, but moreover that the risk initially grows with the sample size. Meanwhile both versions of SafeBayes perform fine. Bayes appears to recover slowly, comparably to the ridge regression experiments of Grünwald and van Ommen (2014). Because the experiments are computationally intensive, we stopped at sample size 200, but we expect Bayes to recover fully and concentrate on $\tilde{P}$ eventually. In addition we expect Bayes to concentrate even later when the dimensionality p of the model increases, never concentrating when $p \to \infty$.

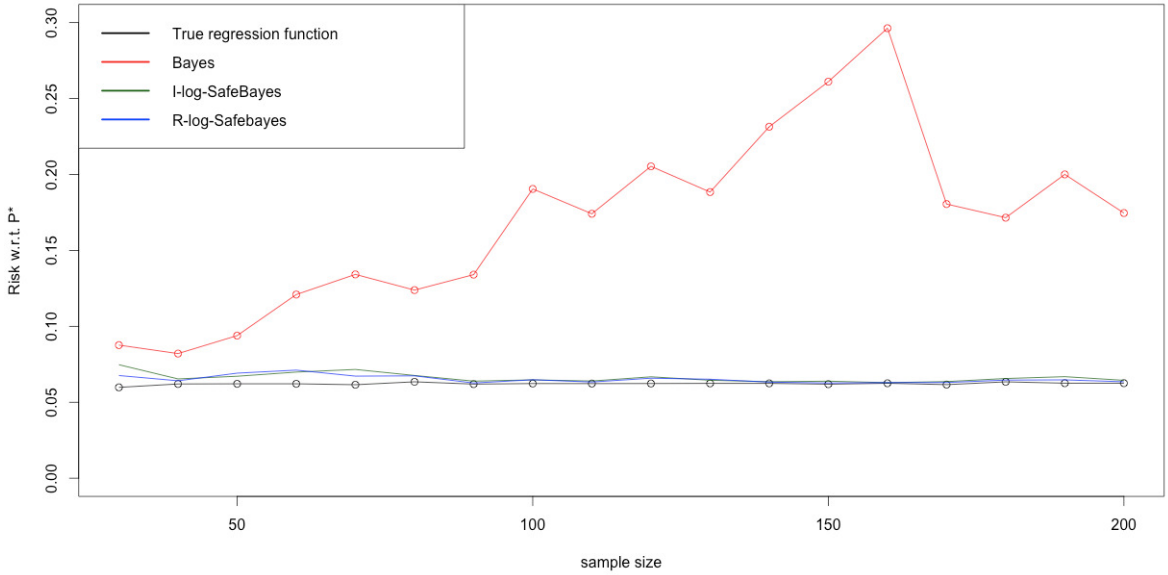


Figure 3.4: The empirical squared error risk with respect to P^* as a function of the sample size for the *wrong-model* experiments as defined in section 3.1 according to the posterior predictive distribution of the standard Bayesian lasso (red), I-log-SafeBayes (green) and R-log-SafeBayes (blue) with standard improper priors and 201 Fourier basis functions.

Correct-model risk

In Section 3.2 we surprisingly discovered that the standard Bayesian lasso showed its 'bad' behaviour in correct-model experiments. To investigate this behaviour further, we examine the empirical square-risk with respect to P^* as a function of the sample size.

We observe from Figure 3.5 and Figure 3.6 that for lower sample sizes Bayes performs badly, but not as bad as in the wrong-model experiments. It recovers much faster than in the wrong-model experiments. In the coming section we consider an explanation for this phenomenon.

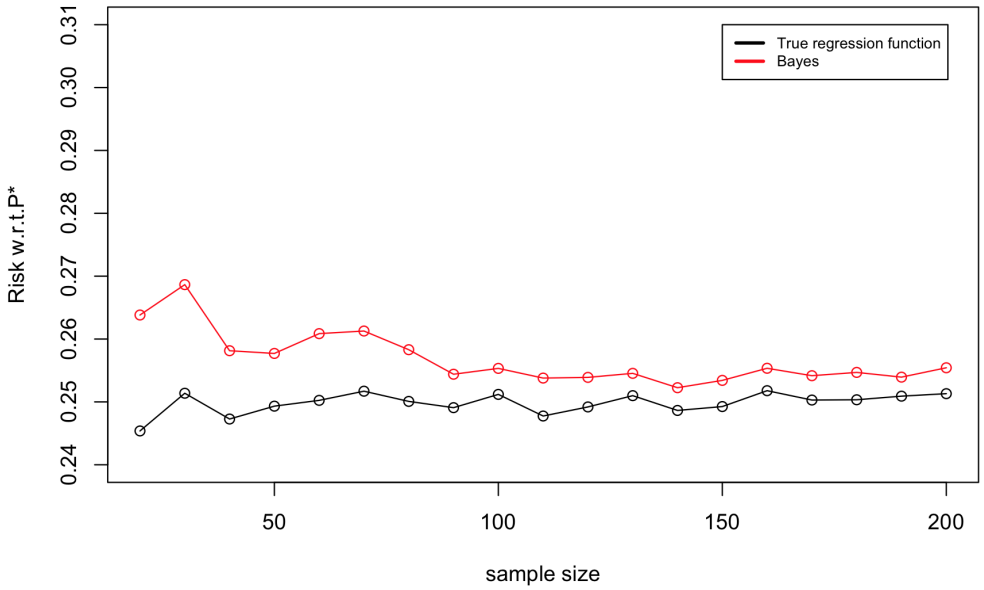


Figure 3.5: The empirical squared error risk with respect to P^* as a function of the sample size for the *correct-model* experiments as defined in section 3.1. The standard Bayesian lasso is equiped with standard improper priors on σ^2 and λ.

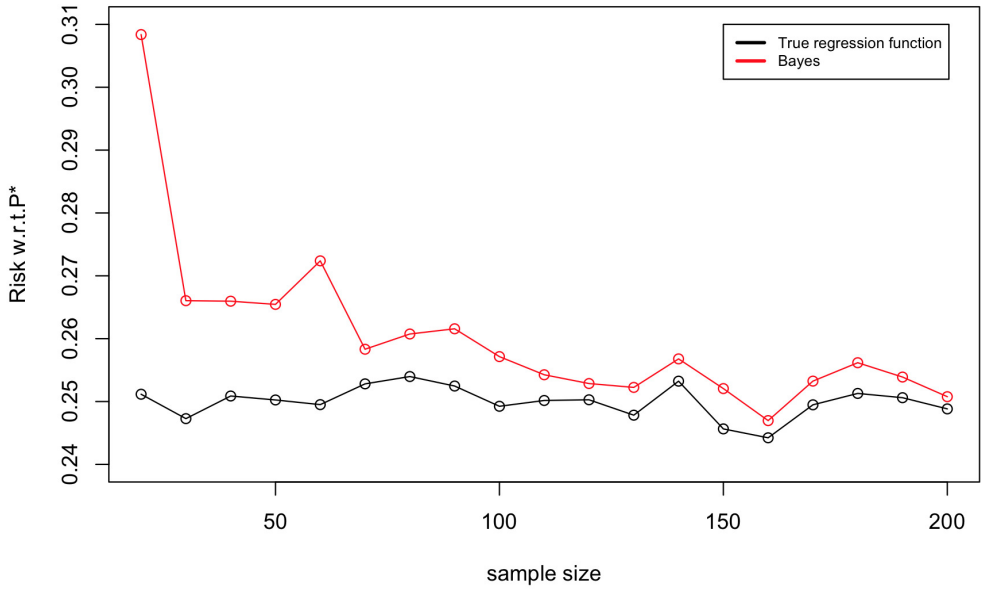


Figure 3.6: The empirical squared error risk with respect to P^* as a function of the sample size for the *correct-model* experiments as defined in section 3.1. The standard Bayesian lasso is equiped with a standard improper prior on σ^2 and a Beta$(1.4, 1.4)$ prior on λ.

3.4 Explanation and Discussion

Bayes' behaviour can be explained if we recall Section 1.4 and look at Figure (1.2). The Bayesian predictive density $\overline{P}(Y_n|X_n, Z^{n-1})$ is a mixture of different distributions P_θ (where $\theta = (\beta, \sigma^2, \tau^2, \lambda)$). When the predictive distribution is considerably different from the distributions *in* the model, 'bad' misspecification can occur (see Section 1.4). In this section we examine this more closely (see also Grünwald and van Ommen (2014)).

The log-risk of the Bayes predictive distribution $\overline{P}$ can be written as (Section 15.2 from Grünwald (2007)):

$$\begin{aligned} & \mathbf{E}_{P^*}\left[-\log \overline{P}(Y_n|X_n, Z^{n-1})\right] \\ = \; & \mathbf{E}_{P^*}\left[-\log \frac{\overline{P}(Y_n|X_n, Z^{n-1})}{P^*(Y_n|X_n)}\right] + \mathbf{E}_{P^*}\left[-\log P^*(Y_n|X_n)\right]. \end{aligned} \tag{3.2}$$

The first term of (3.2) is the KL-divergence (1.22) between the true distribution P^* and the predictive distribution. The second term is defined as the *entropy* of P^*, denoted by $H(P^*)$, which is independent of the sample size n. We thus have:

$$\begin{aligned} \textsc{risk}^{\log}\overline{P}(\,\cdot\,|Z^{i-1}) &= \mathbf{E}_{P^*}\left[-\log \overline{P}(Y_n|X_n, Z^{n-1})\right] \\ &= \mathrm{KL}(P^* \parallel \overline{P}(\,\cdot\,|\,\cdot\,, Z^{n-1})) + H(P^*). \end{aligned} \tag{3.3}$$

The log-risk of the best distribution in our model (the distribution closest to P^* in KL divergence), which we denote by $P_{\tilde{\theta}}$, is

$$\begin{aligned} \textsc{risk}^{\log}P_{\tilde{\theta}} &= \mathbf{E}_{P^*}\left[-\log P_{\tilde{\theta}}(Y_n|X_n)\right] \\ &= \mathrm{KL}(P^* \parallel P_{\tilde{\theta}}) + H(P^*). \end{aligned} \tag{3.4}$$

Now, we want to look at the difference between our predictive distribution and the best distribution in our model (3.4). We subtract the right hand sides of (3.3) from (3.4) and we see that the entropy term $H(P^*)$ cancels. Barron (1998) showed that the cumulative log-risk of sequential Bayesian prediction is bounded:

$$\mathbf{E}_{Z^n \sim P^*}\left[\sum_{i=1}^{n} \textsc{risk}^{\log}\overline{P}(\,\cdot\,|Z^{i-1}) - \textsc{risk}^{\log}P_{\tilde{\theta}}\right] \leq \textsc{red}_n, \tag{3.5}$$

where $\textsc{red}_n$ in regression models of dimension p is of order $\frac{p}{2}\log n$. Since the risks on the left hand side of (3.5) increase linearly, Barron's bound implies that that the difference of the risks must be very close to 0 for most i. Even though we have misspecification, the predictive distribution becomes *good* in log-risk.

We can easily see (from for example Section 2.5) that for our regression model with parameters β and σ^2 we have

$$\textsc{risk}^{\log}(\beta, \sigma^2) = \frac{1}{2\sigma^2}\textsc{risk}^{\mathrm{sq}}(\beta) + \frac{1}{2}\log(2\pi\sigma^2). \tag{3.6}$$

The best distribution in our model with parameters $(\tilde{\beta}, \tilde{\sigma}^2)$ thus minimizes not only the log-risk, but also the square-risk. Furthermore, $\mathbf{E}_{P^*}[Y|X]$ is our *true regression function* and in all our experiments it is contained in the model.

So far it seems that we do not have a problem: if some (β, σ^2) yields a log-risk close to that of $(\tilde{\beta}, \tilde{\sigma}^2)$, its square-risk will be close to that of $(\tilde{\beta}, \tilde{\sigma}^2)$ as well. The catch is, that (3.6) holds for individual (β, σ^2), not for mixtures of them as we have in our predictive distribution. The difference between the good log-risk (Bayes) and the bad square-risk (Sections 3.2 and 3.3) implies that the posterior is not concentrated. The difference implies that substantially many components of the predictive distribution $\overline{P}(\cdot|Z^i)$ must be substantially different from every P_θ in the model for most i.

Correct-model 'misspecification'

Consider our results in Section 3.2, where the standard Bayesian lasso exhibited a weaker version of its 'bad' behaviour in the correct-model experiments: it seems that problems may still arise in the case that the the model is well-specified. We speculate that an explanation for this is that the true distribution can be expressed as a convex combination of distributions that are also in the model but that are far from it in terms of KL divergence. This is possible, for example, if the model is convex and the true distribution lies in its interior (suitable defined) or the true distribution lies at the boundary (suitable defined) of the model in a region where the boundary is convex but not strictly convex (see Figure 3.7). Although the 'true' distribution P^* is in our model, and thus the optimal distribution in our model $\tilde{P}$ is equal to P^*, the predictive distribution might be 'close' to P^* in KL divergence, but still a convex combination of other, 'bad' elements in the model. So we can again have that Barron's theorem holds ($\tilde{P}$ is 'close' to P^*), yet there is non-concentration and bad square-loss performance, as we saw in Section 3.3.

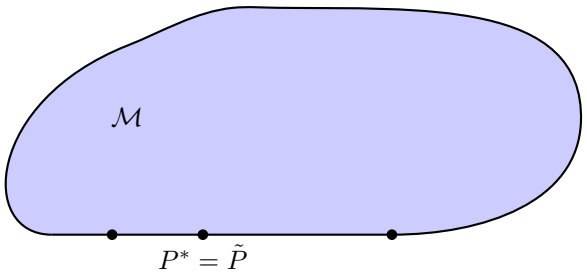


Figure 3.7: The true distribution P^* lies at the boundary of the model $\mathcal{M}$ in a region where the boundary is convex, but not strictly convex. Althoug the 'true' distribution P^* equals the best distribution in our model, $\tilde{P}$, the predictive distribution might still be a convex combination of other, 'bad' distributions in the model.

The Fourier basis

In the wrong-model experiments, Grünwald and van Ommen (2014) get essentially the same results for other basis functions than the Fourier basis, such as polynomials, while

we do not see this with the Bayesian lasso. A possible explanation for this may be the following. The predictive distribution based on the given data may consist of different types of mixtures of other distributions. One possibility is that it consists of many elements, e.g. many polynomial basis functions, with small coefficients. Another option is that the predictive distribution consists of a few distributions with large coefficients. The feature that distinguishes the Bayesian lasso from the methods investigated by Grünwald and van Ommen (2014), is that the Bayesian lasso shrinks the small, non-important coefficients towards zero and leaves the larger coefficients nearly unaffected. It then seems unlikely that the first type of predictive distribution mentioned arises for our data: a mixture of many distributions with small coefficients, as used by Grünwald and van Ommen (2014). The Bayesian lasso will shrink these small coefficients towards zero. In this way, the shrinkage property of the Bayesian lasso may prevent one type of 'bad' misspecification from taking place. The alternative option in which the 'bad' behaviour persists with the Bayesian lasso, is a predictive distribution that is a mixture of a few, large elements. With only a few polynomials it is difficult to construct a graph that passes through many data points as in one of the examples of Section 3.2. With a few Fourier basis functions, this is easily accomplished.

Chapter 4

Model Selection

The basic lasso (with λ determined by cross-validation) performs both shrinkage and variable selection to improve both prediction and interpretability in linear regression models. The variable selection property is one of the main merits of the basic lasso. Coefficients can be set to zero exactly and features or functions are removed from the model automatically, without needing a model selection on top. In the Bayesian lasso, the Laplace prior for the coefficients has a peak at zero. Consequently, values close to zero are given high probability. Because of the relatively thick tails of the prior, coefficients that are further from zero are shrunk less. As described in Section 2.1, the basic lasso estimator coincides with the posterior mode of the Bayesian lasso with Laplace priors on the coefficients. We use posterior means instead of modes as point estimators, and these approach yet do not reach zero exactly. The Bayesian lasso (with a well-specified model) loses the variable selection property, but it keeps good prediction accuracy in most cases. Sometimes however, the prediction accuracy of the standard Bayesian lasso is decreased in the correct-model set-up, as we saw in Section 3.2. The standard Bayesian lasso did not learn the true regression function in Figure 3.2 and Figure 3.3, and it yielded suboptimal predictions as well. In the wrong-model experiments of Chapter 3, the Bayesian lasso did not learn the true regression function, and predicted very poorly. We would like to find out if using a model selection method on top of the standard Bayesian lasso leads to regaining one or both of the desired properties: good prediction and variable selection. The Safe-Bayesian lasso does not perform variable selection either, but in our experiments of Chapter 3 it showed reasonable predictive performance. This is an indication that sufficient regularization takes place. However, we would like to see whether we can improve prediction further by performing model selection on top of the Safe-Bayesian lasso, and whether we, as a side-effect, select variables.

In this chapter we address these questions provisionally. The first two sections provide an overview of recent approaches to either modifying the Bayesian lasso so that it does perform variable selection, or using a model selection method, such as the Bayes factors method (Spiegelhalter et al., 2002), on top of it. Thereafter, we look into two specific model selection approaches: Bayes Factor model selection and the Deviance Information Criterion, and apply them to one of the examples from the previous chapter. It turns out that in the wrong-model experiment, based on the DIC one would strongly prefer the most complex, standard Bayesian lasso model over the simpler standard Bayesian lasso and the generalized Bayesian lasso models. This is far from the true regression function and yields bad predictions on new data. Bayes factor model selection gives the desired results when comparing standard Bayes with SafeBayes, but not when comparing the standard Bayesian

models of different dimensions mutually. Whether prediction of the Safe-Bayesian lasso can be improved by performing model selection on top of it remains inconclusive at this point. Further experiments need to be performed to answer this question.

4.1 Variable selection in the Bayesian lasso

The fully Bayesian lasso does not set coefficients of non-important variables exactly to zero, but it does shrink them towards it. Therefore, one approach could be to set coefficients to zero when zero lies within the posterior credible interval (Fahrmeir et al., 2010). This however does not take into account model uncertainty. It also depends on the construction of the credible intervals and the posterior probability attached to it. Furthermore, this approach does not quantify how important each variable is (Lykou and Ntzoufras, 2013). All following approaches are mainly based on the choice of the prior for the coefficients and the shrinkage parameter λ. They all try to shrink small coefficients as much as possible while leaving the larger coefficients relatively unaffected.

Balakrishnan and Madigan (2010) propose the *demi-Bayesian lasso*. In this version, the Bayesian lasso and sparse Bayesian learning are combined through a mixture of a normal-exponential prior. The mixing parameter controls which variables are excluded from the model. It is estimated by maximizing the marginal data likelihood. Zero values in the mixing parameter indicate that the variable is excluded. The shrinkage parameter λ is determined through cross-validation.

The Bayesian lasso of Hans (2009) uses a Laplace prior similar to Park and Casella (2008), and a gamma prior on the shrinkage parameter λ. Hans (2009) focuses on prediction rather than variable selection. Hans (2010) discusses model uncertainty for this version of the Bayesian lasso. For small models, he computes the marginal posterior probabilities. For large models, he uses a mixture prior of a point mass at zero and a Laplace prior to sample the posterior inclusion probabilities.

Griffin and Brown (2010) adopt a normal-gamma prior to shrink the posterior expectation of the coefficients closely towards zero. This results in a generalization of the Bayesian lasso. Further they suggest a data-dependent prior for the shrinkage parameter λ.

Other Bayesian approaches to variable selection have been tried on the Bayesian lasso as well. One important class of priors are *spike and slab priors* (Mitchell and Beauchamp, 1988). George and McCulloch (1997) use *zero inflated mixture priors* for linear regression. Zhao and Sarkar (2015) applied them to the Bayesian lasso. Yuan and Lin (2005) also use a mixture of a point mass and a Laplace distribution for the linear model, and they prove that the resulting model with the highest posterior probability is the lasso solution. Lykou and Ntzoufras (2013) suggest a comparable mixture prior, but concentrate on the specification of the shrinkage parameter λ. It is determined by Bayes factors, that evaluates the inclusion of each variable.

Other priors for shrinkage and variable selection in linear regression have been proposed, such as the Horseshoe prior (Carvalho et al., 2010) and the double generalized Pareto prior (Armagan et al., 2013).

4.2 Bayes factors

The *Bayes factor method* (Kass and Raftery, 1995) is an approach to model selection in the Bayesian framework (Grünwald, 2007). When we need to compare a hypothesis H_1 to an other H_2 with data y, we can look at the ratio of posterior probabilities (Gelman et al., 2004)

$$\frac{p(H_2|y)}{p(H_1|y)} = \frac{p(H_2)}{p(H_1)} \times \text{Bayes factor}(H_2; H_1), \tag{4.1}$$

where the Bayes factor is defined as

$$\text{Bayes factor}(H_2; H_1) := \frac{p(y|H_2)}{p(y|H_1)} = \frac{\int p(\theta_2|H_2)p(y|\theta_2, H_2)\, d\theta_2}{\int p(\theta_1|H_1)p(y|\theta_1, H_1)\, d\theta_1}. \tag{4.2}$$

The Bayes factor expresses by what factor the ratio of the prior probabilities of the two hypotheses changes due to seeing the data.

For our (Safe-)Bayesian lasso, we have the complication that we only have a Markov Chain Monte Carlo simulation of the posterior distribution. Bayes factor involves computation of the marginal likelihood for each model. In our case this is analytically not possible, and evaluation by numerical integration is not feasible.

Various approximations to the Bayes factor have been suggested. They roughly fall into two categories. The *harmonic mean approximation* (HMA) makes no prior assumptions about the distributions. It is however often inconsistent, typically for models with vague prior distributions, because then the harmonic mean integral is improper. The second category consists of approximations that make strong assumptions about the model or the posterior distribution. For example the Laplace approximation (Kass and Raftery, 1995) works well if the posterior is smooth, unimodal, and well-represented by a multidimensional normal distribution (Weinberg et al., 2013). Several other solutions are very specific and can not be adapted to more general settings (Yoon et al. (2011), Lu et al. (2012)). Weinberg (2012) proposes an algorithm for computing a marginal likelihood from an MCMC simulation. In Weinberg et al. (2013) an extension to the algorithm is made, and it basically comes down to the following idea. The algorithm assigns probability to a tree partition of the sample space. Thereafter, the marginal likelihood integral itself is numerically computed. It is consistent for all proper posterior distributions (Weinberg, 2012). However, it still runs into problems for larger sample sizes and dimensionalities, due to the *curse of dimensionality* (Weinberg et al., 2013). At this moment it seems not feasible to use for our (Safe-)Bayesian lasso.

4.3 Dawid's prequential approach to Bayes factors

Let us look at one of the two factors in (4.2). We see that the marginal distribution of data x given some parameters θ and a prior $w(\theta)$: $\int p(x|\theta)w(\theta)\, d\theta$ matches the prior predictive distribution (1.17). We can rewrite a marginal distribution on a sequence of outcomes x^n as a product of sequential predictions (Dawid, 1984), using the chain rule.

$$P(x^n) = P(x^1)\frac{P(x^2)}{P(x^1)} \cdot \ldots \cdot \frac{P(x^n)}{P(x^{n-1})}$$
$$= \prod_{i=1}^{n} P(x_i|x^{i-1}) \tag{4.3}$$

The terms in the product are the probabilities assigned to the x_i by the predictive probability distributions conditional on the previous data x^{i-1}.

Application to the (Safe-)Bayesian lasso

For our (Safe-)Bayesian lasso setting of Chapter 2 and 3, we have data $z^n = (x^n, y^n)$. For predicting y_i we use the posterior mean $\overline{\theta}$ of the parameters $\theta = (\beta, \sigma^2, \tau^2, \lambda)$ for some model M_1. The product (4.3) then becomes:

$$\begin{aligned} p_1(y^n|x^n) &= \prod_{i=1}^{n} p(y_i|x_i, z^{i-1}, M_1) \\ &= \prod_{i=1}^{n} \int_\theta (y_i|x_i) w(\theta|z^{i-1}) d\theta \\ &\approx \prod_{i=1}^{n} p_{\overline{\theta}_{(z^{i-1})}}(y_i|x_i) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2}(y_i - \overline{\beta}_{(z^{i-1})} x_i)^2 \right\}, \end{aligned} \tag{4.4}$$

which we can readily implement. Since the lasso prior for β is proper, the computation of (4.4) is straightforward for models $\mathcal{M}^{(p)}$ with fixed variance. However, for models $\mathcal{S}^{(p)}$, we are in trouble when using an improper prior on σ^2, including Jeffreys' prior, which we used throughout Chapter 3. In this case, the variance for the first factor is zero. For the second factor we can not sample β for the same reason. This is a *start-up problem* as described in Example 9.1 of Grünwald (2007). We can easily solve it by setting both factors to a default value, the same for both models under comparison (see also p. 431 of Grünwald (2007)).

4.4 Other methods

There are many other methods for Bayesian model selection: AIC, BIC, Bayesian model averaging, etc. An examination of how these methods can be used in practice for (Safe-)Bayesian lasso model selection problems may be included in future work. For this thesis we look into one more method for model selection from a pragmatic point of view: the *Deviance Information Criterion* (DIC) (Spiegelhalter et al., 2002). The DIC is a method that is especially useful in the case that (only) a MCMC simulation of the posterior distribution is available. It is very easy to implement for our (Safe-)Bayesian lasso problems of Chapter 3. As with many methods for model selection the DIC has limitations: for example, it is not consistent and one of its terms is not invariant to reparameterisation. Therefore, variations on it have been proposed since the paper of Spiegelhalter et al. (2002) (see e.g. Spiegelhalter et al. (2014)). As said, appropriate model selection methods for the (Safe-)Bayesian lasso need further investigation. For now we examine the 'original' DIC to

compare substantially different models in the next section.

Spiegelhalter et al. (2002) suggest a measure of complexity, the *effective number of parameters*:

$$p_D = \mathbf{E}_{\theta|X}\left[-2\log\left\{p(X|\theta)\right\}\right] + 2\log\left\{p\left(X \mid \tilde{\theta}(X)\right)\right\}. \tag{4.5}$$

If $\mathbf{E}[\theta|X]$ is taken to be $\tilde{\theta}$, p_D can be calculated from the MCMC simulation of the posterior by taking minus the posterior likelihood minus the likelihood at the posterior mean. The DIC is:

$$\mathrm{DIC} = D(\hat{\theta}) + 2p_D\,, \tag{4.6}$$

where $D(\hat{\theta})$ is the deviance of the expectation of θ. Models with lower DIC should be preferred to models with higher DIC. As a rule of thumb, differences of more than 10 are a strong indication to rule out the model with the higher DIC. Differences between 5 and 10 are substantial, but differences below 5 are not an indication that the models make substantially different inferences.

4.5 Model selection in the wrong-model experiment

We perform the wrong-model experiment as described in Section 3.1 to compare the standard Bayesian lasso and the generalized Bayesian lasso with $\eta = 0.2$ with several numbers of basis functions. In Table 4.1 we see the empirical square-risk for those models and with respect to the true regression function, so that we can compare the models' predictive performance.

# Basis functions	201	101	51	25	3
Sq.Risk true regression function	0.0611	0.0616	0.0602	0.0632	0.0638
Sq.Risk Bayes	0.2802	0.2770	0.1487	0.0946	0.0672
Sq.Risk Gen. Bayes ($\eta = 0.2$)	0.0682	0.0618	0.0625	0.0647	0.0639

Table 4.1: The empirical square-risk with respect to P^* of the true regression function, the Bayesan lasso and the generalized Bayesian lasso with different numbers of basis functions on the wrong-model experiment as described in Section 3.1.

DIC

First we calculate the DIC (4.6) from the MCMC simulations of the posterior. The result can be seen in Table 4.2. If we compare the standard Bayesian lasso to the generalized Bayesian lasso with an equal number of basis functions, we see that the DIC of standard Bayes is constantly lower than the DIC of generalized Bayes. An explanation for this is that the complexity of both models is approximately equal, because the (generalized) Bayesian lasso does not set coefficients exactly to zero. The fit of the standard Bayesian lasso is substantially better: it shows substantial overfitting. Hence the DIC is not a suitable criterion to compare models in which one or both of the models appear to be inconsistent in the way described in Section 3.4. Furthermore, observe the DIC of the standard Bayesian lasso with the different number of basis functions in Table 4.2. On basis of the DIC one would strongly

prefer the most complex model. It seems that the first term of the DIC (4.6), the goodness of fit, strongly dominates the complexity term. This is again intuitively logical, since the inconsistency as described in Section 3.4 manifests itself as severe overfitting. However, we do not see the same phenomenon when we compare the DIC's of the generalized Bayesian lasso with the different sizes of the Fourier basis. For the generalized Bayesian lasso, the simplest model would be preferred weakly. The generalized Bayesian lasso shrinks all the coefficients strongly towards zero for all dimensions of the Fourier basis, so predictions based on those models are comparable, as we see in Table 4.1.

# Basis functions	201	101	51	25	3
DIC Bayes	-166	-95	-81	-78	-79
DIC Gen. Bayes ($\eta = 0.2$)	-65	-71	-70	-69	-73

Table 4.2: DIC of the Bayesan lasso and the generalized Bayesian lasso with different numbers of basis functions on the wrong-model experiment as described in Section 3.1.

Bayes Factors

For the Bayes Factor model comparison as described in Section 4.3 we see the Bayes evidence for the different models in Table 4.3. Contrary to the DIC we see that when we compare the generalized Bayesian lasso to the standard Bayesian lasso with the same number of basis functions, the evidence for the generalized Bayesian lasso model is very strong in all cases. When we compare the Bayes evidence of the standard Bayesian lasso to that of one with a different number of basis functions, we see the same problem as with the DIC: some of the more complex models are strongly preferred. On the contrary, we see that for the generalized Bayesian lasso the simplest model with 3 basis functions is preferred, albeit weakly with respect to the models with 51 and 25 basis functions. This preference is weak for reasons analogous to those with the DIC.

# Basis functions	201	101	51	25	3
BE(logBE) Bayes	$1.21\,\mathrm{e}-7$ (-15.93)	$9.00\,\mathrm{e}2$ (6.80)	$1.41\,\mathrm{e}-78$ (-179.25)	$1.01\,\mathrm{e}-4$ (-9.20)	$1.14\,\mathrm{e}-35$ (-80.45)
BE(logBE) ($\eta = 0.2$) Gen. Bayes	$3.44\,\mathrm{e}8$ (19.66)	$1.13\,\mathrm{e}9$ (20.84)	$1.06\,\mathrm{e}10$ (23.08)	$2.64\,\mathrm{e}10$ (24.00)	$2.96\,\mathrm{e}10$ (24.11)

Table 4.3: Bayes evidence (log Bayes evidence) of the Bayesan lasso and the generalized Bayesian lasso with different numbers of basis functions on the wrong-model experiment as described in Section 3.1.

From our initial experiments it appears that Bayes Factor model selection is an appropriate method to choose the 'best' model when we compare consistent models, and when we compare a model that shows inconsistency as described in Section 3.4 with a model that does not. 'Best' is both in the sense of 'closest to the true regression function' and corresponding to the minimum square-risk for new data (Section 3.2).

Using the DIC or Bayes factor model selection on top of the standard Bayesian lasso does not solve or ease the problem of 'bad' misspecification, i.e. it cannot *save Bayes*. Whether prediction of the Safe-Bayesian lasso is improved by performing model selection on top of it remains inconclusive, based on the single experiment with the generalized Bayesian lasso in this chapter. Further experiments need to be performed to answer this question. Further research into model selection for the (Safe-/generalized) Bayesian lasso may be a subject for future work.

Chapter 5

Real-world data

In this chapter we present the findings of a search for real-world examples in which the standard Bayesian lasso performs inferior to the Safe-Bayesian lasso. From our simulation examples of Chapter 3, we saw that the standard Bayesian lasso showed inconsistency only with a Fourier basis. To avoid going too far afield, we should thus look for data in which the choice of a Fourier basis would not strike one as particularly odd. In this chapter, three such real-world examples are presented. They show that bad misspecification is a problem not only in theory, but also in practice. In the three examples, but also in the numerous data sets we encountered in our search, SafeBayes never performed substantially worse than standard Bayes. Moreover, in the examples presented below, SafeBayes mostly performs substantially better than standard Bayes.

5.1 Temperature in Seattle

The R-package `weatherData` (Narasimhan, 2014) can load weather data online available from `www.wunderground.com`. This website aims to share information with the public from weather stations all over the world. Besides data from many thousands of personal weather stations and government agencies, the website provides access to data from Automated Surface Observation Systems (ASOS) stations located at airports in the US, owned and maintained by the Federal Aviation Administration. Among them is a weather station at Seattle Tacoma International Airport, Washington (WMO ID 72793). From this station we collect the data for first experiment.

We take a look at the maximum temperatures for each day of the year 2011 at Seattle airport (Figure 5.1). We divide the data randomly in a training set (300 measurements) and a test set (65 measurements). First, we sample the posterior of the standard Bayesian lasso with a 201-dimensional Fourier basis and standard improper priors on the training set. Next, we sample the generalized posterior with the learning rate $\hat{\eta}$ learnt by the four versions of the Safe-Bayesian algorithm, with the same model and priors on the same training set. We take a relatively small grid of η's, because of the lengthy computations involved. The two square-Safe-Bayesian algorithms require a fixed variance. For those fixed variances, we estimate the sample variance from the training data, for each addition of a data-point to the training data in the Safe-Bayesian algorithm afresh. We compare the performance of the standard Bayesian lasso (Bayes) and the four Safe-Bayesian versions of the lasso (SB)

	Bayes	I-log SafeBayes	R-log SafeBayes	I-square SafeBayes	R-square SafeBayes
MSE $((°C)^2)$	8.32	7.53	6.93	7.87	7.35
η	1	0.9	0.6	0.9	0.9

Table 5.1: Mean square errors (5.1) for the Bayesian lasso and four versions of the Safe-Bayesian lasso on the Seattle weather data. The folling grid of η's was used: $\{1, 0.9, 0.85, 0.8, 0.75, 0.7, 0.6, 0.5, 0.25\}$.

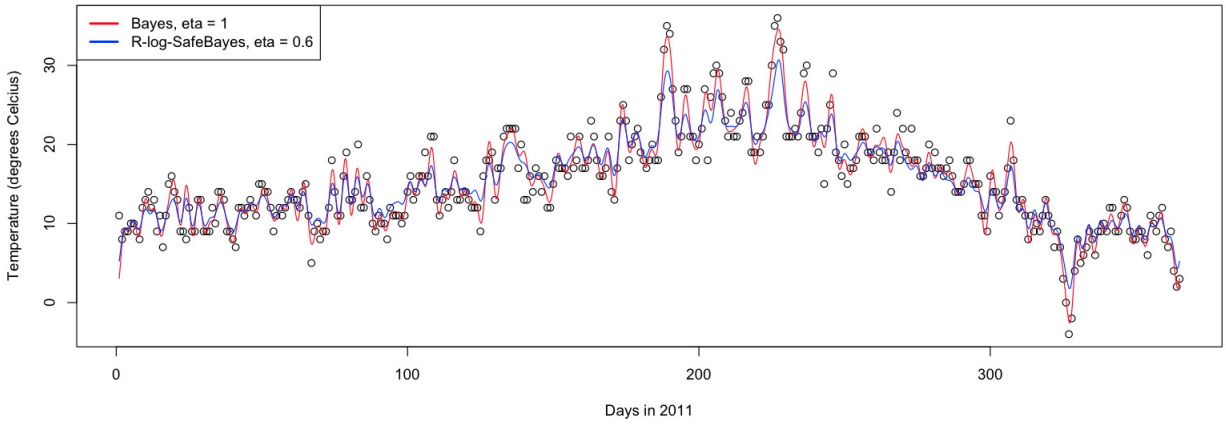


Figure 5.1: Seattle weather data: maximum temperatures for each day of the year 2011, with predictions of the standard Bayesian lasso (red) and the R-log-Safe-Bayesian lasso (blue, $\eta = 0.6$) from the posterior sampled on a training set of 300 data points.

in terms of mean square error

$$\frac{1}{n}\sum_{i=1}^{n}\left(y - \sum_{j=0}^{p}\beta_j X_j\right)^2 \tag{5.1}$$

on predictions for the test set in Table 5.1.

Experiments with different priors for λ yielded similar results. The first observation that stands out, is that the four versions of SafeBayes do not always pick the same η. Which η is chosen by a Safe-Bayesian algorithm depends partly on the specific partition of the training and test set. For some partitions, all versions of SafeBayes pick relatively high η's (> 0.7), for some other partitions, all versions pick relatively low η's (< 0.75). However, in all experiments performed with different partitions, priors and number of iterations, SafeBayes never picked $\eta = 1$. Moreover, whichever learning rate was chosen by SafeBayes, SafeBayes always outperformed standard Bayes (with $\eta = 1$) in an unchanged set-up.

SafeBayes does not always outperform standard Bayes with maximum temperature data. We can see this when we perform exactly the same analysis on weather data from Schiphol airport, Amsterdam (WMO ID 06240). Averaged over 10 different partitions of training and test set, the mean square errors for standard Bayes and R-log-SafeBayes (the grid of

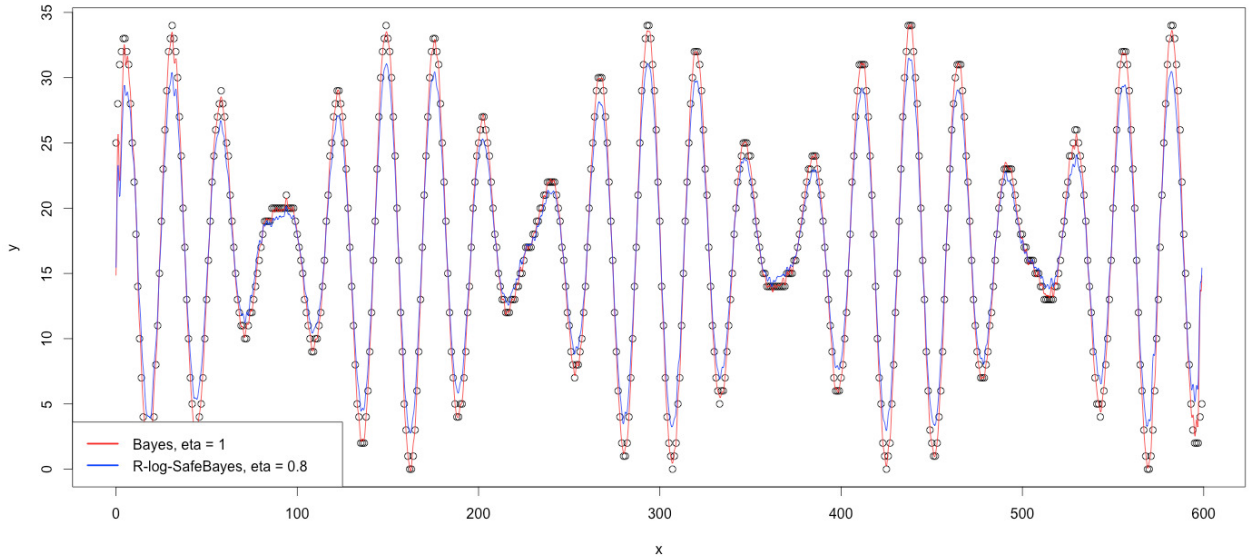


Figure 5.2: Variable star data with predictions of the standard Bayesian lasso (red) and the R-log-Safe-Bayesian lasso (blue, $\eta = 0.8$) from the posterior sampled on a training set of 500 data points.

η's is $\{1, 0.9, 0.8, 0.7\}$) are respectively 4.75 and 4.65. When looking at the individual prediction errors, sometimes SafeBayes predicts slightly better, and sometimes slightly worse than standard Bayes.

We find that the Safe-Bayesian lasso is never substantially worse, and sometimes substantially better than the standard Bayesian lasso.

5.2 Variable star

The second example consists of the *variable star series* from Whittaker and Robinson (1924). A *variable star* is a star whose apparent magnitude as seen from the earth fluctuates. The data give the magnitudes (AU) of the variable star RW Cassiopeiae at midnight on 600 consecutive nights, observed by Whittaker from Dublin. The series is almost exactly the sum of two cosinusoids, which is shown in Chapter 2 of Bloomfield (2000).

We divide the data randomly in a training set (500 measurements) and a test set (100 measurements), and sample the posteriors of the four versions of SafeBayes and standard Bayes with improper priors and a 701-dimensional Fourier basis on the training set. We make predictions for the test set. We look at the mean-square error (Table 5.2).

Does the R-Safe-Bayesian lasso really predict better than the standard Bayesian lasso, or is it coincidence of this particular partition of the training and test set? Or has standard Bayes not yet converged enough? — Because performing SafeBayes with a grid of 8 η's with 1100 sampling iterations took 36 hours, it is not feasible to repeat this several times with substantially more iterations for the Gibbs sampler. Therefore, we will use the learn-

	Bayes	I-log SafeBayes	R-log SafeBayes	I-square SafeBayes	R-square SafeBayes
MSE	6.11	6.10	5.52	6.28	5.66
η	1	1	0.8	1	0.8

Table 5.2: Mean square errors for the Bayesian lasso and four versions of the Safe-Bayesian lasso on the variable star data.

ing rate $\hat{\eta}$ learnt by one run of SafeBayes. We sample the posterior of standard Bayes and the generalized Bayesian lasso with the fixed $\hat{\eta}$ ten times on different partitions of training and test set. For each partition we predict for the test set. Then we average the resulting mean square errors (Table 5.3). We see a fairly large difference between standard Bayes and generalized Bayes. The reason for this, is that for particular partitions of the training and test set, standard Bayes' performance was exceptionally bad compared to generalized Bayes.

Standard Bayes ($\eta = 1$)	Generalized Bayes ($\eta = 0.8$)
12.63	7.90

Table 5.3: Mean square errors for prediction of a test set (100 measurements) with the standard Bayesian lasso and generalized Bayesian lasso on a training set (500 measurements) of the variable star data, averaged over 10 different random partitions of the data in training and test set.

The Safe-Bayesian lasso outperforms the standard Bayesian lasso on average for the variable star data in different set-ups. Contrary to the Seattle weather data, in isolated partitions of training and test set SafeBayes sometimes yields slightly worse predictions than standard Bayes. SafeBayes also sometimes chooses $\eta = 1$. Again we find that the Safe-Bayesian lasso never performs substantially worse, and sometimes substantially better than the standard Bayesian lasso.

5.3 London air pollution

Our third example consists of air pollution data from the Openair Project of the Environmental Research Group of King's College, London (Carslaw and Ropkins (2012) and Carslaw (2015)). The project's aim is to make air pollution data of thousands of monitoring sites available to the public, together with open-source software to perform analyses on this data. For our experiments, we use the data from a monitoring station at Marylebone Road (UK-AIR ID: UKA00315, EU Site ID: GB0682A). Marylebone Road is an important thoroughfare in London's city center. We choose to look at the concentration (in parts per billion) of nitrogen dioxide (NO_2). Nitrogen dioxide is a pollutant that can be found in exhaust gas. The concentration of NO_2 at the Marylebone Road site is highly dependent on the traffic. Therefore it shows a periodic pattern with differences both between day and night, and between weekend and working days. Hence it seems to be an appropriate data set to analyze with a Fourier basis Bayesian lasso.

Opposed to the Seattle weather data and the variable star data, we do not predict for a random test set within the data, but for a wholly different set. As training set we use the following data. We start with the first four weeks of the year 2013, starting at Monday Jan-

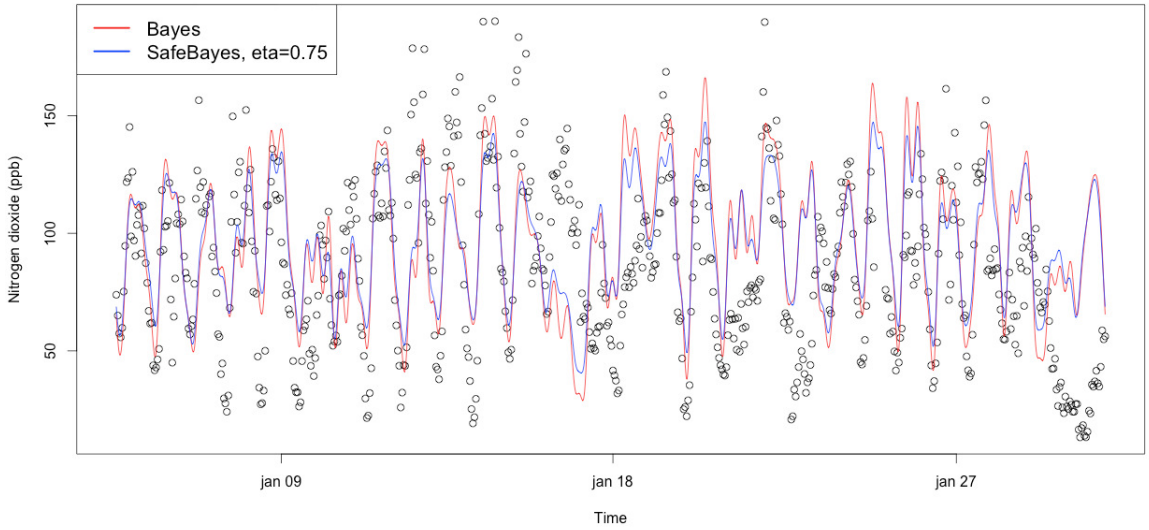


Figure 5.3: Predicions of the standard Bayesian lasso (red) and Safe-Bayesian lasso (blue, $\eta = 0.5$) for the concentration of nitrogen dioxide (ppm) for the first four weeks of 2015. Both models are trained on a random selection of measurements from the first four weeks of the two preceding years, 2013 and 2014.

	Bayes	I-log SafeBayes	R-log SafeBayes	I-square SafeBayes	R-square SafeBayes
MSE ((ppm)^2)	1224	1185	1132	1192	1189
η	1	0.95	0.75	0.95	0.95

Table 5.4: Mean square errors for predictions with the Bayesian lasso and four versions of the Safe-Bayesian lasso on the London air pollution data. For the $\hat{\eta}$ obtained by SafeBayes, the posterior of the generalized Bayesian lasso is sampled 10 times and the prediction errors are averaged.

uary 7 at midnight. We have a measurement for (almost) every hour until Sunday February 3$^{\text{rd}}$, 23.00. We also have data for the first four weeks of 2014, starting at Monday January 6 at midnight, until Sunday February 2$^{\text{nd}}$, 23.00. For each hour in the four weeks we randomly pick a data point from either 2013 or 2014. We remove the missing values. We predict for the same time of year in 2015: starting at Monday January 5 at midnight, until Sunday February 1$^{\text{st}}$ at 23.00. We do this with a (Safe-)Bayesian lasso with a 201-dimensional Fourier basis and standard improper priors. In Figure 5.3 we see the data for 2015 with the predictions for the (Safe-)Bayesian lasso trained on the previously described data. We look at the mean square prediction errors, and average the errors over 10 runs of the generalized Bayesian lasso with the η learnt by SafeBayes. These results are shown in Table 5.4.

Again we find that SafeBayes performs substantially better than standard Bayes.

5.4 Discussion

The experiments in this chapter show that there exist real-world data sets in which the Safe-Bayesian lasso performs substantially better than the standard Bayesian lasso in terms of prediction. Moreover, the Safe-Bayesian lasso never performed substantially worse than the standard Bayesian lasso. This was not only the case in the data examples presented above, but also in the many data sets we encountered in our search for examples in which the Safe-Bayesian lasso would outperform the standard Bayesian lasso.

Since data points are sequentially added in the Safe-Bayesian algorithm, the application of the algorithm to real-world data is computationally demanding, even though our implementation is relatively fast. When the sample size and the grid of η's are sizable, the use of SafeBayes is impractical.

In addition, we observed the following during the execution of the experiments and from the results. As mentioned in the previous sections, the four versions of SafeBayes do not pick the same learning rate η. Moreover, if we consider one of the versions of SafeBayes individually, it picks different η's for different partitions of the training and the test set. In these real-world data sets, the η's chosen by SafeBayes seem relatively high compared to the η's from our simulation experiments (Chapter 3) and those from Grünwald and van Ommen (2014). Presumably the misspecification in the real-world data is less extreme than the misspecification we constructed for the simulation experiments. Likely this is also the cause of the following observation. When we look at the figures, SafeBayes appears to be a 'less extreme version' of standard Bayes on all data sets. We did not apply some form of model selection on top of the (Safe-)Bayesian lasso. We could do further research into this, but at first glance is seems not necessary to do model selection on top of SafeBayes, the Safe-Bayesian methods regularize sufficiently by themselves. Interestingly, as we will see in Chapter 6, on the Seattle weather data, the Safe-Bayesian lasso performs better than the basic lasso (with λ determined by cross-validation) in the same set-up, which does serve as a model selection method. Lastly we observed a few instances in which standard Bayes and SafeBayes did not learn at all on the variable star and London air pollution data, for no apparent reason. A possible explanation for this is that mixing problems occur due to the prior on λ, and the MCMC algorithm explores the posterior poorly. Occurrences of this phenomenon were very few.

Chapter 6

Future work

The results presented in this thesis raise several new questions, which we hope to address in future work. We expound these questions in Section 6.2. The first section of this chapter describes an interesting idea that arose when this thesis was nearly finished.

6.1 The Bayesian Lasso without the parameter λ

In the basic lasso the penalty parameter λ (determined by cross-validation) controls the trade-off between model complexity and goodness of fit. In the Bayesian lasso, the parameter λ determines the shape of τ_j, which we can see as a latent variable to determine the Laplace prior on the coefficients. Many ways have been proposed to deal with the parameter λ, see Section 2.4, and e.g. Park and Casella (2008), de los Campos et al. (2009) and many others. The question arose whether or not the learning rate η could neutralize the effect of the λ parameter, or even replace it. If we set λ at a fixed value, for example 1, the full conditional distribution on τ_j becomes

$$\frac{1}{\tau_j^2} = \mathrm{IG}\left(\sqrt{\frac{\sigma^2}{\beta_j^2}}, 1\right).$$

Above all: no method is needed to determine or sample λ anymore, which proved to be troublesome. The learning rate η replaces the effect of λ. Since it is not a parameter of our model it can be determined by e.g. some form of cross-validation.

For an initial examination of this idea we look at the well-known prostate cancer data set from the Elements of Statistical Learning (Hastie et al., 2009), which can be found on `www.stat.stanford.edu/` $\sim$ `tibs/ElemStatLearn/`. The features are 8 clinical measures, and the response is the log of the prostate specific antigen. As a biological data set, the features are strongly related as can be seen from the correlation matrix in Table 6.1.

First we fit ordinary least squares and the basic lasso on a randomly selected training set, and look at the mean square error of the prediction on the test set. These methods yield a mean square error of 0.5213 and 0.5050 respectively. Next we predict according to the posterior of the Bayesian lasso with standard improper priors, which yields mean square error of 0.5083. The posterior mean of λ is 3.5139. In the following experiment we put a point mass on λ (resulting in a fixed λ), and vary the learning rate η for the generalized

	lcavol	lweight	age	lbph	svi	lcp	gleason	pgg45	lpsa
lcavol	1.000	0.300	0.286	0.063	0.593	0.692	0.426	0.483	0.733
lweight	0.300	1.000	0.317	0.437	0.181	0.157	0.024	0.074	0.485
age	0.286	0.317	1.000	0.287	0.129	0.173	0.366	0.276	0.228
lbph	0.063	0.437	0.287	1.000	-0.139	-0.089	0.033	-0.030	0.263
svi	0.593	0.181	0.129	-0.139	1.000	0.671	0.307	0.481	0.557
lcp	0.692	0.157	0.173	-0.089	0.671	1.000	0.476	0.663	0.489
gleason	0.426	0.024	0.366	0.033	0.307	0.476	1.000	0.757	0.342
pgg45	0.483	0.074	0.276	-0.030	0.481	0.663	0.757	1.000	0.448
lpsa	0.733	0.485	0.228	0.263	0.557	0.489	0.342	0.448	1.000

Table 6.1: Correlation matrix of the prostate cancer data set.

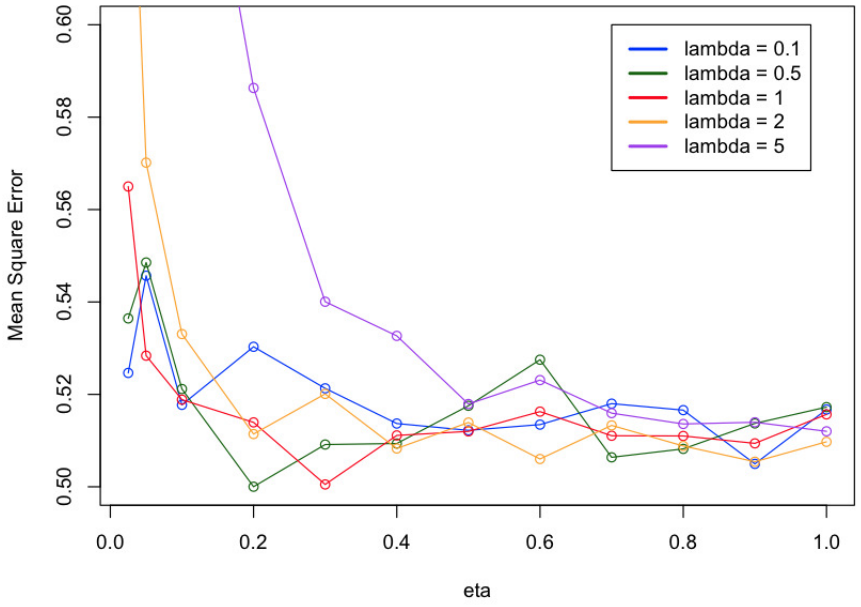


Figure 6.1: The mean square errors for the generalized Bayesian lasso on the prostate cancer data. The parameter λ is kept at a fixed value, while the learning rate η is varied.

Bayesian lasso. The results can be seen in Figure 6.1.

On the right side Figure 6.1, we can see the mean square errors for the standard Bayesian lasso ($\eta = 1$) with the different values of the fixed λ. They perform better than ordinary least squares, and (slightly) worse than the basic lasso. For a clear overview, these mean square errors are in the first column of Table 6.2. What happens when we vary η (second column of Table 6.2) is very interesting.

These results suggest that the generalized Bayesian lasso with a fixed value of λ and a learning parameter η performs at least as good as the standard Bayesian lasso (with no learning parameter) with a vague prior on λ. Furthermore, for some choices of the value of the fixed λ, the generalized Bayesian lasso with fixed λ performs *better* than the standard Bayesian lasso with a vague prior on λ. An intuitive choice for the value of a fixed λ would

Method	MSE with BL: $\eta = 1$	MSE with BL: $\eta = \eta_{\text{MIN}}$
Ordinary Least Squares	0.5213	–
Lasso	0.5050	–
BL with improper priors	0.5083	–
BL with $\lambda = 0.1$	0.5167	0.5050
BL with $\lambda = 0.5$	0.5172	0.5000
BL with $\lambda = 1$	0.5157	0.5005
BL with $\lambda = 2$	0.5097	0.5054
BL with $\lambda = 5$	0.5120	0.5120

Table 6.2: Mean square errors for predictions on the prostate cancer data with the following methods. Ordinary least squares; the basic lasso (with λ determined by cross-validation); the standard Bayesian lasso (BL) with improper priors; the standard Bayesian lasso with several fixed values of λ (first column), and the generalized Bayesian lasso with the optimal η chosen from a grid: $\{0.025, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ (second column).

be the aforementioned 1, which yields excellent results on this data, but we need to investigate this further in future work. We would hope that the learning rate can ensure good performance for arbitrary λ, but this is not the case. As we see from Table 6.2, the choice for $\lambda = 5$ gives suboptimal performance for all choices of η. We will see in the next section on an other data set that for some choices of λ, the performance can be much worse.

Concluding, these first results suggest that the generalized Bayesian lasso with a fixed value of λ and a learning parameter η performs equally good or possibly better than the standard Bayesian lasso with a (vague) prior on λ.

The Bayesian lasso without λ in a misspecified model

Even more interesting would be to see what happens in the case that the model is misspecified; i.e. when the Safe-Bayesian lasso outperforms the standard Bayesian lasso. Fortunately we have such data sets at our disposal from Chapter 5. We perform the same experiments on the Seattle weather data as on the prostate cancer data mentioned above.

Figure 6.2 and Table 6.3 show the results of the experiments. The Bayesian lasso with a standard improper prior on λ yields a posterior mean of $\lambda = 0.3794$. This is smaller than 1, contrary to the prostate cancer experiments, and therefore we see a different pattern in Figure 6.2. From Table 6.3 we see that the standard Bayesian lasso with a prior on λ performs slightly worse than the basic lasso, but substantially better than ordinary least squares and the variations of the Bayesian lasso with a fixed parameter λ. For some choices of this fixed value, the generalized Bayesian lasso with a fixed value of λ and a learning parameter η performs substantially better than the standard Bayesian lasso. Moreover, surprisingly, it performs substantially better than the basic lasso. This could be explained by the fact that the lasso picks the penalty parameter λ through cross-validation (10-fold in our case) that minimizes the cross-validation error. The model with this λ_{MIN} tends to be too complex and slightly overfitting. Another choice of the penalty parameter could be the value of λ that is 1 standard error larger than λ_{MIN}, resulting in a simpler model. However in our case, the mean square error for prediction with a lasso model fitted with λ_{1se} is 8.3279. This is even larger than that of the lasso model with λ_{MIN}.

Method	MSE with BL: $\eta = 1$	MSE with BL: $\eta = \eta_{\text{MIN}}$
Ordinary Least Squares	12.7687	–
Lasso	8.1143	–
BL with improper priors	8.2527	–
SB worst version	7.87	–
BL with $\lambda = 0.1$	9.2723	8.4907
BL with $\lambda = 0.5$	10.6582	6.9947
BL with $\lambda = 1$	20.0700	6.9949
BL with $\lambda = 2$	24.6235	7.6332
BL with $\lambda = 5$	25.4626	11.6823

Table 6.3: Mean square errors for predictions the Seattle weather data with the following methods. Ordinary least squares; the basic lasso (with λ determined by cross-validation); the standard Bayesian lasso (BL) with improper priors; the worst version of SafeBayes (SB) from Section 5.1, the standard Bayesian lasso with several fixed values of λ (first column), and the generalized Bayesian lasso with the optimal η chosen from a grid: $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ (second column).

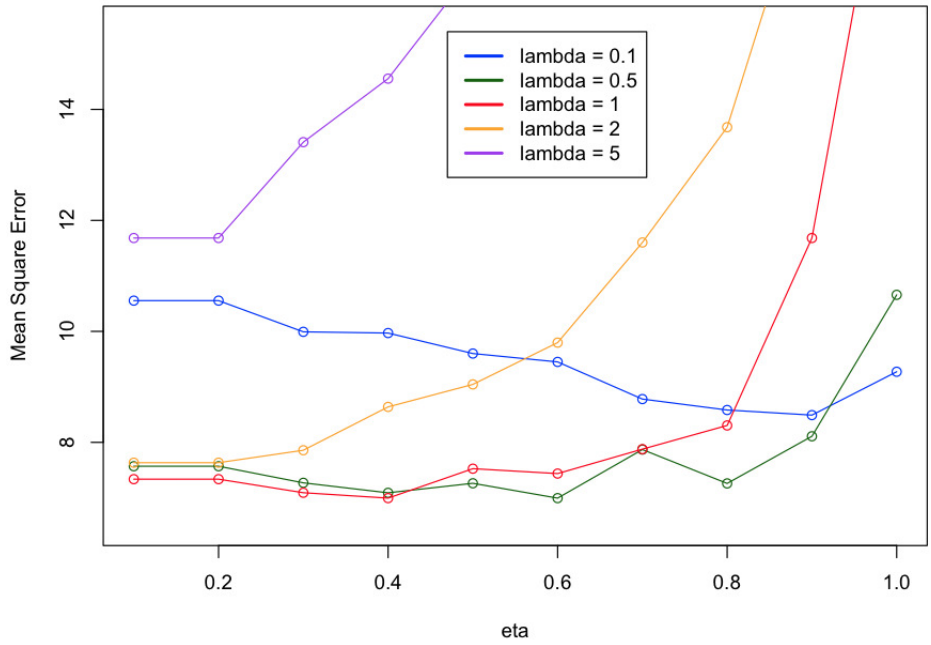


Figure 6.2: The mean square errors for the generalized Bayesian lasso on the Seattle weather data. The parameter λ is kept at a fixed value, while the learning rate η is varied.

Further research needs to be done in this direction. Our preliminary results suggest that the generalized Bayesian lasso with a fixed value of λ and a learning rate η could be a better alternative to the standard Bayesian lasso. Not only does our new method perform comparably to the standard Bayesian lasso (or slightly better), our new method outperforms the standard Bayesian lasso substantially in the case that the model is misspecified. Further research needs to be done to see if the learning rate η may be chosen through some form of cross-validation, which makes the method computationally feasible, and an attractive alternative to the Bayesian lasso in the case that no strong prior information about the λ parameter is available.

6.2 Further work

The Safe-Bayesian algorithm, in combination with the Gibbs sampling, is very time-intensive. Perhaps it might work as well in practice to do a 'quick and dirty SafeBayes', in which one randomly splits the sample in two, and picks the learning rate from a finite grid such that the first sample gives the best randomized or in-model square-loss or log-loss predictions on the second half. We can view this as a kind of two-fold forward (rather than cross-) validation, where we measure error in the same way as in the original SafeBayes method. Results from Grünwald (2012) suggest that this works. We would like to find out whether this — much faster — method performs comparably. The optimal learning rate η is learnt from half the sample. Perhaps the Safe-Bayesian algorithm on the full sample yields a completely different η. Because of this, an interesting question would be whether we should predict according to the generalized posterior of half of the sample (the training set), or to the generalized posterior based on the full sample — with the learning rate η obtained from the training set only.

So far we have only searched for real-world data sets in which the Safe-Bayesian lasso outperforms the standard Bayesian lasso with a Fourier basis. It would be interesting to see if we can find real-world data for which other versions of Safe-Bayesian linear regression perform better than standard Bayes, with different bases as well. Our results with the (Safe-)Bayesian lasso on the data in Chapter 5 show that 'bad misspecification' is a problem not only in theory, but also in practice. Finding examples for Bayesian linear regression in a broader setting would endorse this.

We have seen in Chapter 4 that some Bayesian model selection methods yield results that are highly conflicting with the performance of the models in terms of square-loss when the model is misspecified. It would be interesting to examine the performance of more methods for model selection in these situations. The results from the simulation experiments (Chapter 3), the real-world data (Chapter 5) and from the first experiments with a fixed λ parameter (Section 6.1), show that predictions according to the generalized posterior with the 'right' learning rate η demonstrate excellent results. It might be interesting to see whether an additional model selection criterion on top of the Safe-Bayesian lasso (for example to choose an appropriate basis) can further improve the Safe-Bayesian lasso. The results from Chapter 4 and Section 6.1 suggest that the learning rate η causes additional regularization, such that the resulting model and predictions are very similar for different amounts of basis functions.

The Safe-Bayesian algorithm has an interpretation in terms of sequential prediction with squared error loss with the *exponential weighted forecaster* algorithm. This algorithm does essentially the same as generalized Bayes with fixed σ^2 and is designed for online applica-

tions. Consequently, it would be interesting to see for this algorithm whether it helps to learn the variance from the data (i.e. pick the empirical variance that would have been best in the past), and whether that can have both a beneficial and a negative effect.

Conclusion

In this thesis, we empirically investigated the behaviour of the Bayesian lasso under model misspecification. It turns out that the Bayesian lasso can be inconsistent when the model is misspecified — it contains a 'good', not a 'true' distribution — and Bayes does not find this 'good' distribution. A method to deal with this problem if it occurs is the generalized posterior, with a learning rate η. We investigated the Safe-Bayesian method of Grünwald (2012) to lean the 'right' η.

For this thesis, we implemented several functions that shortly will be made publicly available as the `R`-package `SafeBayes`. All simulations and experiments in this thesis were performed in `R`. The Safe-Bayesian algorithm is, in combination with the Gibbs-sampling, very time-intensive. Further research needs to be done to find a method that is much faster and performs comparably.

In Chapter 3 we performed experiments on simulated data. When we sampled the posterior of the Bayesian lasso with a Fourier basis to wrong-model data, it showed considerable overfitting. SafeBayes on the other hand appeared to be close to having discovered the true regression function. Interestingly, we saw a weaker version of this phenomenon occur in correct-model data, something we had not expected. We examined the empirical square-risk of the standard Bayesian lasso and the Safe-Bayesian lasso. The square-risk of the standard Bayesian lasso in the wrong-model experiments was not only larger than that of the Safe-Bayesian lasso, it grew with the sample size. Bayes recovered slowly after the sample size became larger than the dimension of the basis. When the dimensionality of the model increases, we expect Bayes to concentrate later, and never concentrate when $p \to \infty$. This behaviour can be explained by the predictive distribution being a mixture of different 'bad' distributions in the model. The discrepancy between the good log-risk and the bad square-risk implies that substantially many components of the predictive distribution must be substantially different from every distribution in the model, hence the posterior is not concentrated. SafeBayes performed excellently in all experiments.

In Chapter 4 we looked at variable selection and model selection methods for the Bayesian lasso that have been proposed in the last few years. We examined the Bayes Factor method and the DIC in the wrong-model experiment with different dimensions of the Fourier basis for both standard Bayes and SafeBayes. Interestingly, based on the DIC one would strongly prefer the most complex, standard Bayesian lasso model. We know from the preceding chapter that this is highly problematic. Bayes factor model selection appears to yield the desired results when comparing the Safe-Bayesian models mutually and SafeBayes with standard Bayes, but not when comparing the standard Bayesian models mutually. The Bayes factor method can not save Bayes. Future work may include research into model selection methods in the situation that the model is misspecified, and research to see whether applying an addi-

tional model selection criterion on top of the Safe-Bayesian lasso could improve performance even further, or that the learning rate induces enough regularization, similar to the basic lasso.

Our real-world data in Chapter 5 show that bad misspecification is a problem not only in theory, but also in practice. There exist real-world data sets in which the Safe-Bayesian lasso performs substantially better than the standard Bayesian lasso in terms of prediction. Importantly, in our analyses the Safe-Bayesian lasso never performed substantially worse than the standard Bayesian lasso. This was also the case for all data sets we encountered in our search but which we did not include in this thesis. The analyses of all data sets in Chapter 5 were performed with a Fourier basis, because we saw in Chapter 3 that the standard Bayesian lasso's 'bad' behaviour presented with a Fourier basis only. It would be interesting to see if we can find real-world data for which other versions of Safe-Bayesian linear regression perform better than standard Bayes, with different bases as well.

Future research will include an interesting question raised in Chapter 6. Our preliminary results suggest that a better alternative to the standard Bayesian lasso could be the generalized Bayesian lasso with a fixed value of λ and a learning rate η. Not only does our new method perform comparably to the standard Bayesian lasso (or slightly better) if the model is correct, our new method outperforms the standard Bayesian lasso substantially in the case that the model is misspecified.

Bibliography

Andrews, D. F. and C. L. Mallows
1974. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(1):99–102.

Armagan, A., D. Dunson, and J. Lee
2013. Bayesian generalized double pareto shrinkage. *Statistica Sinica*, 23:119–143.

Audibert, J. Y.
2004. *PAC-Bayesian statistical learning theory.* PhD thesis, Université Paris VI.

Balakrishnan, S. and D. Madigan
2010. Priors on the variance in sparse Bayesian learning: the demi-Bayesian lasso. In *Frontiers of Statistical Decision Making and Bayesian Analysis: In Honor of James O. Berger*, P. Muller, D. Sun, and K. Ye, eds., Pp. 346–359. Springer, Berlin.

Barron, A. R.
1998. Information-theoretic characterization of Bayes performance and the choice of priors in parametric and nonparametric problems. In *Bayesian Statistics*, J. Bernardo, J. Berger, A. Dawid, and A. Smith, eds., volume 6, Pp. 27–52. Oxford University Press.

Bloomfield, P.
2000. *Fourier Analysis of Time Series: An Introduction*, second edition edition. Wiley.

Box, G. E. P. and N. R. Draper
1987. *Empirical Model-Building and Response Surfaces.* Wiley, New York.

Carslaw, D. C.
2015. *The openair manual - open-source tools for analysing air pollution data. Manual for version 1.1-4.* King's College London.

Carslaw, D. C. and K. Ropkins
2012. Openair - an R package for air quality data analysis. *Environmental Modelling & Software*, 27-18:52–61.

Carvalho, C., N. Polson, and J. Scott
2010. The horseshoe estimator for sparse signal. *Biometrika*, 97:465–480.

Catoni, O.
2007. *PAC-Bayesian Supervised Classification.* Lecture Notes-Monograph Series. IMS.

Chhikara, R. S. and J. L. Folks
1989. *The Inverse Gaussian Distribution: Theory: Methodology, and Applications.* Marcel Dekker, Inc.

Dawid, A. P.
1984. Present position and potential developments: Some personal views, statistical theory, the prequential approach. *Journal of the Royal Statistical Society, Series A*, 147(2):278–292.

de los Campos, G., D. Naya, J. Gianola, J. Crossa, A. Legarra, E. Manfredi, K. Weigel, and J. Cotes
2009. Predicting quantitative traits with regression models for dense molecular markers and pedigree. *Genetics*, 182:375–385.

de los Campos, G. and P. Pérez
2010. BLR: Bayesian linear regression. *R package version*, 1.

Fahrmeir, L., T. Kneib, and S. Konrath
2010. Bayesian regularisation in structured additive regression: a unifying perspective on shrinkage, smoothing and predictor selection. *Statistics and Computing*, 20:203–219.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin
2004. *Bayesian Data Analysis*. Chapman and Hall/CRC.

George, E. and R. McCulloch
1997. Approaches for Bayesian variable selection. *Statistica Sinica*, 7:339–374.

Ghoshal, S., J. K. Ghosh, and A. W. van der Vaart
2000. Convergence rates of posterior distributions. *Annals of Statistics*, 28(2):500–531.

Gramacy, R. B. and E. Pantaleo
2009. Shrinkage regression for multivariate inference with missing data, and an application to portfolio balancing. *Bayesian Analysis*, 5(2):237–262.

Griffin, J. E. and P. J. Brown
2010. Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis*, 5:171–188.

Grünwald, P. D.
2007. *The Minimum Description Length Principle*. The MIT Press.

Grünwald, P. D.
2012. *Algorithmic Learning Theory: 23rd International Conference, ALT 2012, Lyon, France, October 29-31, 2012. Proceedings*, chapter The Safe Bayesian, Pp. 169–183. Berlin, Heidelberg: Springer Berlin Heidelberg.

Grünwald, P. D. and J. Langford
2007. Suboptimal behavior of Bayes and MDL in classification under misspecification. *Machine learning*, 66(2-3):119–149.

Grünwald, P. D. and T. van Ommen
2014. *Inconsistency of Bayesian Inference for Misspecified Linear Models, and a Proposal for Repairing It*. eprint arXiv:1412.3730.

Hans, C.
2009. Bayesian lasso regression. *Biometrika*, 96:835–845.

Hans, C.
2010. Model uncertainty and variable selection in Bayesian lasso regression. *Statistics and Computing*, 20:221–229.

Hastie, T., R. Tibshirani, and J. Friedman
2009. *The Elements of Statistical Learning*, second edition edition. Springer.

Kass, R. E. and A. E. Raftery
1995. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795.

Kleijn, B. J. K.
2004. *Bayesian asymptotics under misspecification*. PhD thesis, Vrije Universiteit Amsterdam.

Kleijn, B. J. K. and A. W. van der Vaart
2006. Misspecification in infinite-dimensional Bayesian statistics. *The Annals of Statistics*, 34(2):837–877.

Li, J. Q.
1999. *Estimation of Mixture Models*. PhD thesis, Yale University, New Haven, CT.

Lu, Y., H. J. Mo, N. Katz, and M. D. Weinberg
2012. Bayesian inference of galaxy formation from the k-band luminosity function of galaxies: tensions between theory and observation. *Monthly Notices of the Royal Astronomical Society*, 421:1779–1796.

Lykou, A. and I. Ntzoufras
2013. On Bayesian lasso variable selection and the specification of the shrinkage parameter. *Statistics and Computing*, 23(3):361–390.

McAllester, D.
2003. PAC-Bayesian stochastic model selection. *Machine Learning*, 51(1):5–21.

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

Narasimhan, R.
2014. Package weatherdata, get weather data from the web. URL: http://ram-n.github.io/weatherData/.

Park, T. and G. Casella
2008. The Bayesian lasso. *Journal of the American Statistical Association*, 103:369–412.

Rynne, B. P. and M. A. Youngson
2008. *Linear Functional Analysis*, second edition edition. Springer.

Seeger, M.
2002. PAC-Bayesian generalization error bounds for gaussian process classification. *Journal of Machine Learning Research*, 3:233–269.

Spiegelhalter, D. J., N. G. Best, B. P. Carlin, and A. van der Linde
2002. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society, Series B*, 64(4):583–639.

Spiegelhalter, D. J., N. G. Best, B. P. Carlin, and A. van der Linde
2014. The deviance information criterion: 12 years on. *Journal of the Royal Statistical Society, Series B*, 76(3):485–493.

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

Vovk, V. G.
1990. Aggregating strategies. In *Proc. COLT 90*, Pp. 371–383.

Weinberg, M. D.
2012. Computing the Bayes factor from a markov chain monte carlo simulation of the posterior distribution. *Bayesian Analysis*, 7(3):737–770.

Weinberg, M. D., I. Yoon, and N. Katz
2013. *A remarkably simple and accurate method for computing the Bayes Factor from a Markov chain Monte Carlo Simulation of the Posterior Distribution in high dimension.* eprint arXiv:1301.315.

Whittaker, E. T. and G. Robinson
1924. *The Calculus of Observations*. English Universities Press, London.

Yoon, I., M. D. Weinberg, and N. Katz
2011. New insights into galaxy structure from GALPHAT-I. motivation, methodology and benchmarks for sersic models. *Monthly Notices of the Royal Astronomical Society*, 414:1625–1655.

Yuan, M. and Y. Lin
2005. Efficient empirical Bayes variable selection and estimation in linear models. *Journal of the American Statistical Association*, 100(472):1215–1225.

Zhang, T.
2004. Learning bounds for a generalized family of Bayesian posterior distributions. In *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, eds., Pp. 1149–1156. MIT Press.

Zhang, T.
2006. From ε-entropy to KL entropy: analysis of minimum information complexity density estimation. *Annals of Statistics*, 34(5):2180–2210.

Zhao, Z. and S. Sarkar
2015. On the credible interval under the zero-inflated mixture prior in high dimension inference. *Statistica Sinica*, 25(2):725–742.

Appendix A

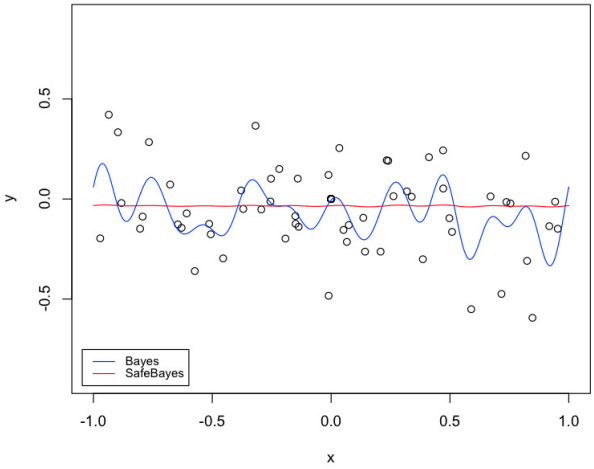


Figure 1: Predictions of standard Bayes (blue) and SafeBayes (red, $\eta = 0.5$) for the *wrong-model* experiment as described in section 3.1, with 100 data points i.i.d. $\sim P^*$ with a 25-dimensional Fourier basis.

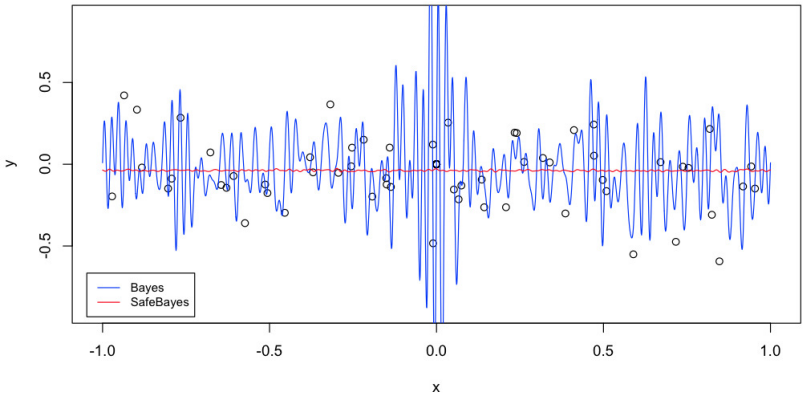
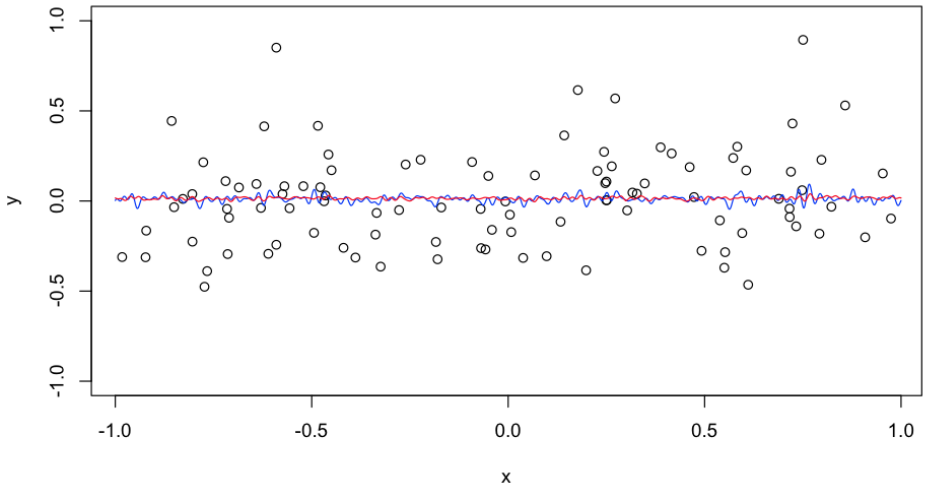
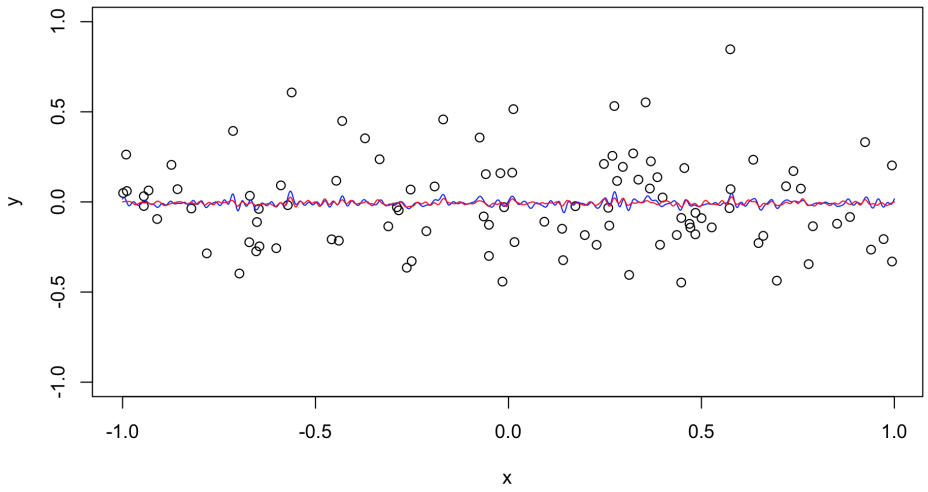
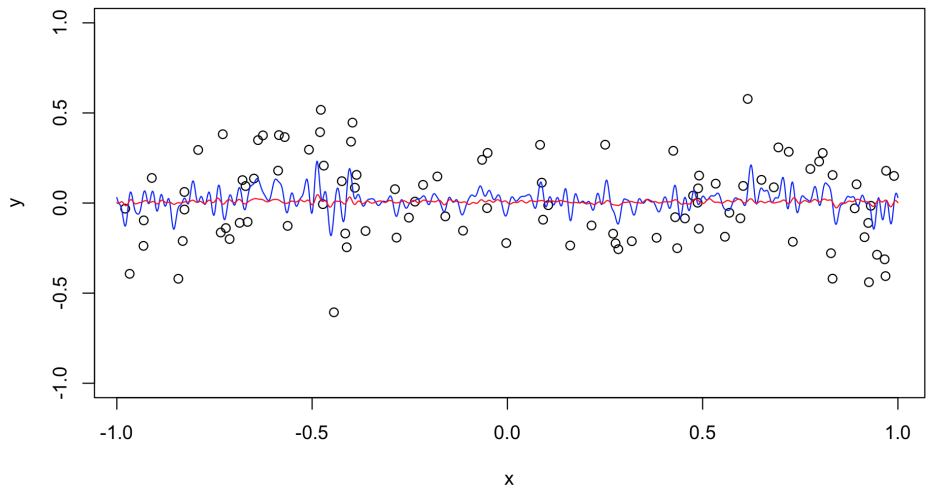


Figure 2: Predictions of standard Bayes (blue) and SafeBayes (red, $\eta = 0.5$) for the *wrong-model* experiment as described in section 3.1, 100 data points i.i.d. $\sim P^*$ with a 201-dimensional Fourier basis and a Beta$(1.4, 1.4)$ prior on λ.


y
1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.5
0.0
0.5
1.0
x


y
1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.5
0.0
0.5
1.0
x


y
1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.5
0.0
0.5
1.0
x

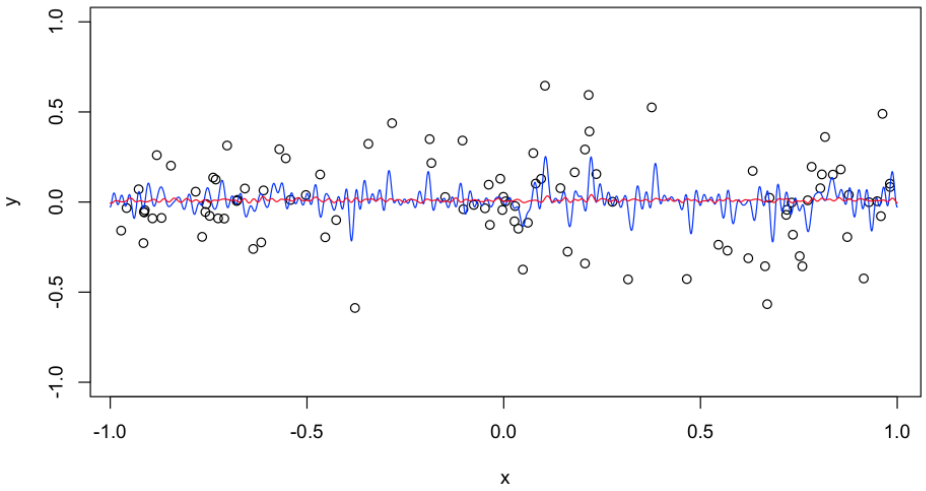


Figure 3: In these four figures *correct-model* experiments are performed as described in Section 3.2. The X_i are sampeled i.i.d. from a uniform distribution on $[-1, 1]$. For every x_i, an y_i is sampled from normal distribution $N(0, 1/4)$. Predictions of the Bayesian lasso are depicted in blue, and of the generalized Bayesian lasso ($\eta = 0.25$) in red. In the first two figures, Bayes and generalized Bayes seem well-matched. In the latter two figures Bayes shows considerably more overfitting.

Bayes, correct model	Gen. Bayes, correct model	Bayes, wrong model	Gen. Bayes, wrong model
0.1022766	0.0966986	0.2962349	0.1653797

Table 1: Empirical square-risk of the experiment in Section 3.2, Figure 3.3. In the correct-model experiment, 400 X_i are sampled random uniform on $[-2\pi, 2\pi]$ and $y_i = \sin(x_i) + \epsilon$ where $\epsilon \sim N(0, 1/4)$. We compare the risk of the standard Bayesian lasso and the generalized Bayesian lasso with $\eta = 0.2$. In the wrong-model, (X, Y) are sampled as before, but with $\epsilon \sim N(0, 1/2)$ and approximately half of the points are set to $(0, 0)$.

Appendix B
SafeBayes user manual

For this thesis several functions are implemented in R, linked to C-code for a relatively fast sampling procedure. The implementation of the Gibbs sampler is partly based on functions from the monomvn package of Gramacy and Pantaleo (2009) and the BLR (Bayesian Linear Regression) package of de los Campos and Pérez (2010). The functions to apply the η-generalized Bayesian lasso and the four Safe-Bayesian algorithms will be made available as an R-package shortly. During the time that the package is not yet available to install directly from an online repository, the first section of this appendix provides instructions for initialization and for loading the provided compiled C-code in R. The subsequent sections provide a description of the different functions. These will be included in the package manual.

Contents

Initialization

Put all files in the same directory, and change the working directory in R by:

```
setwd("<path>")
```

Load the compiled C-code safe_sample_betas.c in R under Windows by:

```
dyn.load("safe_saple_betas.dll")
```

Or load the compiled C-code safe_sample_betas.c in R for all other operating systems by:

```
dyn.load("safe_sample_betas.so")
```

Load the required functions, for example `GBLasso.R` by:

```
source("GBLasso.R")
```

Install and load the `SuppDists` package (needed for sampling from an Inverse Gauss distribution)

```
install.packages("SuppDists")
library(SuppDists)
```

`GBLasso` and `GBLassoFV`

Description

The two functions `GBLasso` (Generalized Bayesian Lasso) and `GBLassoFV` (Generalized Bayesian Lasso with Fixed Variance) provide a Gibbs sampler to fit generalized Bayesian lasso regression models with learning rate `eta`.

Usage

```
GBLasso(y, X, eta, prior, nIter, burnIn, thin,
                            minAbsBeta, weights, pIter)
GBLassoFV(y, X, sigma2, eta, prior, nIter, burnIn
                            thin, minAbsBeta, weights, pIter)
```

Arguments

y Vector of outcome variables, numeric, NA allowed, length n.

X Design matrix, numeric, dimension $n \times p$, $n \geq 2$. Must be of class `matrix`.

sigma2 for use in `GBLassoFV`: (fixed) variance parameter σ^2, numeric. Default `NULL`, in which case the variance will be estimated from the data.

eta Learning rate η, numeric, $0 \leq \eta \leq 1$. Default 1.

prior List containing the following elements

- For use in `GBLasso`: `prior$varE` prior for the variance parameter σ^2 with parameters `$df` and `$S` for respectively degrees of freedom and scale parameters for an inverse-χ^2-distribution. Default $(0, 0)$.
- `prior$lambda` Prior for the penalty parameter λ with three items.
  - `$value` Initial value for λ. Default 50
  - `$type` Can be "`fixed`": initial value is used as fixed penalty parameter or "`random`", in which case a prior for λ is specified. Default "`random`"

  - For a Gamma prior on λ^2: `$shape` for shape parameter and `$rate` for the rate parameter; for a Beta prior on λ: `$shape1`, `$shape2` and `$max` for $\lambda \propto \text{Beta}(\frac{\lambda}{\text{max}}, \text{shape1}, \text{shape2})$. Default $\text{Gamma}(0, 0)$.

nIter Number of iterations, integer. Default 1100.

burnIn Number of iterations for burn-in, integer. Default 100.

thin Number of iterations for thinning, integer. Default 10.

minAbsBeta Minimum absolute value of sampled coefficients $\boldsymbol{\beta}$ to avoid numerical problems, numeric. Default 10^{-9}.

weights Vector of weights, numeric, length n. Default `NULL`, in which case all weights are set to 1.

pIter Print iterations, logical. Default `TRUE`.

Details

See Chapter 2 for details on the implementation of the generalized Bayesian lasso.

Value

`$y`	Vector of original outcome variables.
`$weights`	Vector of weights.
`$mu`	Posterior mean of the intercept.
`$varE`	Posterior mean of of the variance.
`$yHat`	Posterior mean of $\mu + \boldsymbol{X}\boldsymbol{\beta} + \epsilon$.
`$SD.yHat`	Corresponding standard deviation.
`$whichNa`	Vector with indices of missing values of $\boldsymbol{y}$.
`$fit$pD`	Estimated number of effective parameters (Spiegelhalter et al., 2002).
`$fit$DIC`	DIC (Spiegelhalter et al., 2002).
`$lambda`	Posterior mean of λ.
`$bL`	Posterior mean of $\boldsymbol{\beta}$.
`$SD.bL`	Corresponding standard deviation.
`$tau2`	Posterior mean of τ^2.
`$prior`	List containing the priors used.
`$nIter`	Number of iterations.
`$burnIn`	Number of iterations for burn-in.
`$thin`	Number of iterations for thinning.
`$eta`	Learning rate η.

`SBLassoIlog` and `SBLassoRlog`

Description

The two functions `SBLassoIlog` (I-log-Safe-Bayesian lasso) and `SBLassoRlog` (R-log-Safe-Bayesian lasso) provide functions for learning the learning rate η for a Bayesian lasso regression model via the Safe-Bayesian algorithm as described in Section 2.5.

Usage

```
SBLassoIlog(y, X, etaseq, prior, nIter, burnIn, thin,
                                                minAbsBeta, pIter)
SBLassoRlog(y, X, etaseq, prior, nIter, burnIn, thin,
                                                minAbsBeta, pIter)
```

Arguments

y Vector of outcome variables, numeric, NA allowed, length n.

X Design matrix, numeric, dimension $n \times p$, $n \geq 2$. Must be of class `matrix`.

etaseq Vector of learning rates η, numeric, $0 \leq \eta \leq 1$. Default 1.

prior List containing the following elements

- `prior$varE` prior for the variance parameter σ^2 with parameters `$df` and `$S` for respectively degrees of freedom and scale parameters for an inverse-χ^2-distribution. Default $(0, 0)$.
- `prior$lambda` Prior for the penalty parameter λ with three items.
  - `$value` Initial value for λ. Default 50
  - `$type` Can be "`fixed`": initial value is used as fixed penalty parameter or "`random`", in which case a prior for λ is specified. Default "`random`"
  - for a Gamma prior on λ^2: `$shape` for shape parameter and `$rate` for the rate parameter; for a Beta prior on λ: `$shape1`, `$shape2` and `$max` for $\lambda \propto \text{Beta}(\frac{\lambda}{\max}, \text{shape1}, \text{shape2})$. Default $\text{Gamma}(0, 0)$.

nIter Number of iterations, integer. Default 1100.

burnIn Number of iterations for burn-in, integer. Default 100.

thin Number of iterations for thinning, integer. Default 10.

minAbsBeta Minimum absolute value of sampled coefficients $\boldsymbol{\beta}$ to avoid numerical problems, numeric. Default 10^{-9}.

pIter Print iterations, logical. Default `TRUE`.

Details

See Chapter 2 for details on the implementation of Safe-Bayesian lasso.

Value

`$y`	Vector of original outcome variables.
`$weights`	Vector of weights.
`$mu`	Posterior mean of the intercept.
`$varE`	Posterior mean of of the variance.
`$yHat`	Posterior mean of $\mu + \boldsymbol{X}\boldsymbol{\beta} + \epsilon$.
`$SD.yHat`	Corresponding standard deviation.
`$whichNa`	Vector with indices of missing values of $\boldsymbol{y}$.
`$fit$pD`	Estimated number of effective parameters (Spiegelhalter et al., 2002).
`$fit$DIC`	DIC (Spiegelhalter et al., 2002).
`$lambda`	Posterior mean of λ.
`$bL`	Posterior mean of $\boldsymbol{\beta}$.
`$SD.bL`	Corresponding standard deviation.
`$tau2`	Posterior mean of τ^2.
`$prior`	List containing the priors used.
`$nIter`	Number of iterations.
`$burnIn`	Number of iterations for burn-in.
`$thin`	Number of iterations for thinning.
`$CEallen`	For `SBLassoIlog`: list of cumulative η-in-model-log-loss per η.
`$CMRlogEallen`	For `SBLassoRlog`: list of cumulative posterior-expected posterior-randomized log-loss per η.
`$eta.min`	Learning rate η minimizing the cumulative η-in-model-log-loss (I-log-SafeBayes) or cumulative posterior-expected posterior-randomized log-loss (R-log-SafeBayes).

SBLassoISq and SBLassoRSq

Description

The two functions `SBLassoISq` (I-square-Safe-Bayesian lasso) and `SBLassoRSq` (R-square-Safe-Bayesian lasso) provide functions for learning the learning rate η for a Bayesian lasso regression model with *fixed variance* via the Safe-Bayesian algorithm as described in Section 2.5.

Usage

```
SBLassoIlog(y, X, sigma2, etaseq, prior, nIter, burnIn, thin,
                                                 minAbsBeta, pIter)
SBLassoRlog(y, X, sigma2, etaseq, prior, nIter, burnIn, thin,
                                                 minAbsBeta, pIter)
```

Arguments

y Vector of outcome variables, numeric, NA allowed, length n.

X Design matrix, numeric, dimension $n \times p$, $n \geq 2$. Must be of class `matrix`.

sigma2 Fixed variance parameter σ^2, numeric. Default `NULL`, in which case the variance will be estimated from the data *per addition of new data point* in the Safe-Bayesian algorithm.

etaseq Vector of learning rates η, numeric, $0 \leq \eta \leq 1$. Default 1.

prior List containing the following elements

- `prior$lambda` Prior for the penalty parameter λ with three items.
  - `$value` Initial value for λ. Default 50
  - `$type` Can be "`fixed`": initial value is used as fixed penalty parameter or "`random`", in which case a prior for λ is specified. Default "`random`"
  - for a Gamma prior on λ^2: `$shape` for shape parameter and `$rate` for the rate parameter; for a Beta prior on λ: `$shape1`, `$shape2` and `$max` for $\lambda \propto \text{Beta}(\frac{\lambda}{\text{max}}, \text{shape1}, \text{shape2})$. Default $\text{Gamma}(0, 0)$.

nIter Number of iterations, integer. Default 1100.

burnIn Number of iterations for burn-in, integer. Default 100.

thin Number of iterations for thinning, integer. Default 10.

minAbsBeta Minimum absolute value of sampled coefficients $\boldsymbol{\beta}$ to avoid numerical problems, numeric. Default 10^{-9}.

pIter Print iterations, logical. Default `TRUE`.

Details

See Chapter 2 for details on the implementation of Safe-Bayesian lasso.

Value

`$y`	Vector of original outcome variables.
`$weights`	Vector of weights.
`$mu`	Posterior mean of the intercept.
`$varE`	Posterior mean of of the variance.
`$yHat`	Posterior mean of $\mu + \boldsymbol{X}\boldsymbol{\beta} + \epsilon$.
`$SD.yHat`	Corresponding standard deviation.
`$whichNa`	Vector with indices of missing values of $\boldsymbol{y}$.
`$fit$pD`	Estimated number of effective parameters (Spiegelhalter et al., 2002).
`$fit$DIC`	DIC (Spiegelhalter et al., 2002).
`$lambda`	Posterior mean of λ.
`$bL`	Posterior mean of $\boldsymbol{\beta}$.
`$SD.bL`	Corresponding standard deviation.
`$tau2`	Posterior mean of τ^2.
`$prior`	List containing the priors used.
`$nIter`	Number of iterations.
`$burnIn`	Number of iterations for burn-in.
`$thin`	Number of iterations for thinning.
`$CEallen`	For `SBLassoISq`: list of cumulative η-in-model-square-loss per η.
`$CMRSEallen`	For `SBLassoRSq`: list of cumulative posterior-expected posterior-randomized square-loss per η.
`$eta.min`	Learning rate η minimizing the cumulative η-in-model-square-loss (I-square-SafeBayes) or cumulative posterior-expected posterior-randomized square-loss (R-square-SafeBayes).

Examples

```
# Package needed for creating a Fourier Basis
library(fda)

# Simulate data
x <- runif(100, -1, 1) # 100 random uniform x's between -1 and 1
y <- NULL

# for each x, a y that is 0 + Gaussian noise
for (i in 1:100) {
  y[i] <-  0 + rnorm(1, mean=0, sd=1/4)
  }

# Now sample 100 zero's and ones (coin toss)
cointoss <- sample(0:1, 100, replace=TRUE)
# indices of the ones
indices <- which(cointoss==1)

# we replace x and y with (0,0) for the indices the cointoss
# landed tail (1)
x[indices] <- 0
y[indices] <- 0

plot(x,y)

# Create a design matrix with a 51-dimensinal Fourier basis
fourierx <- as.matrix(fourier(x, nbasis=51))

# Determine the generalized posterior for eta = 0.25

obj <- GBLasso(y, fourierx, eta=0.25)

# posterior means of the coefficients beta and intercept mu
betafour <- obj$bL
mufour <- obj$mu

# Plot of the predictive distribution
xgrid <- seq(-1,1, by=0.001)
xfour <- fourier(xgrid, nbasis=51)
ypred <- NULL
ynew <- NULL
for (j in 1:dim(xfour)[1]) {
for (i in 1:dim(xfour)[2]) {
  ynew[i] <- betafour[i]*(xfour[j,i])
}
ypred[j] <- mufour + sum(ynew)
}

plot(x, y, ylim=c(-1, 1))
```

```
lines(xgrid, ypred, col="blue")

# Specify a Beta(lambda/100, 1.4, 1.4) prior for lambda
prior <- list()
prior$lambda$value <- 50
prior$lambda$type <- "random"
prior$lambda$shape1 <- 1.4
prior$lambda$shape2 <- 1.4
prior$lambda$max <- 100

# Determine the standard Bayesian lasso posterior with this prior
obj2 <- GBLasso(y, fourierx, prior=prior)

# posterior means of the coefficients beta and intercept mu
betafour2 <- obj2$bL
mufour2 <- obj2$mu

# Add predictive distribution to the plot
ypred2 <- NULL
ynew2 <- NULL
for (j in 1:dim(xfour)[1]) {
for (i in 1:dim(xfour)[2]) {
  ynew2[i] <- betafour2[i]*(xfour[j,i])
}
ypred2[j] <- mufour2 + sum(ynew2)
}

lines(xgrid, ypred2, col="red")

# Let R-log-SafeBayes learn the learning rate (note: this might
# be computationally intensive)
sbobj <- SBLassoRlog(y, fourierx, prior=prior, etaseq=c(1, 0.5, 0.25))

# eta
sbobj$eta.min

# Determine the generalized posterior for the eta determined by
# R-log-SafeBayes
obj3 <- GBLasso(y, fourierx, eta=sbobj$eta.min, prior=prior)
```