Lennard Jansen

Robust Bayesian inference under model misspecification

Master thesis, defended on July 25, 2013

Thesis advisor: Prof. dr. Peter Grünwald

Specialization: Statistical Science

Mathematical Institute Leiden

Abstract

Bayesian inference is considered one of the best statistical methods available when the model is correctly specified. On the other hand, when this is not the case, and model assumptions do not hold, it can lead to suboptimal results. Equipping the likelihood with a learning rate parameter protects against this. In this thesis the performances of various more robust Bayesian approaches, that differ in the way the learning rate parameter is chosen, are compared to standard Bayes in a variety of situations. Results for various classification problems (with simulated data) and Lasso-type regression problems (with real-world data) indicate that the robust forms of Bayes outperform standard Bayes when the model is incorrect, and don't perform much worse when the model is correct. Especially the robust Bayesian method with learning rate parameter estimated by k-fold cross-validation achieves good results.

Contents

1 Introduction

One of the central goals of statistics is to detect functional relationships based on a finite set of data, such as in regression and classification. However, opinions differ widely on the best means to achieve this. There are two major schools of statistical inference: Bayesian and frequentist. A major attraction of Bayesian inference stems from classical decision theory (Ferguson, 1967). A central result, the so-called complete class theorem, implies that in many situations, optimal frequentist procedures must be equivalent to Bayesian procedures with a certain prior. Thus, even if one is a 'frequentist', one is led to the Bayesian paradigm as the optimal mode of inference. On the other hand, in contrast to frequentist procedures, Bayesian procedures depend on this prior distribution. The problem is then that it is sometimes hard to determine which prior to use. Moreover, in order to behave properly, Bayesian inference generally requires a correctly specified model (Müller, 2009). That is, the model assumptions must hold. For instance, the data that are used being independent and identically distributed according to the unknown true distribution and the residuals distributed as specified by the model.

1.1 Model misspecification

It is known that Bayes performs well in a wide variety of contexts in which the model is correct, that is to say, even in small sample sizes the posterior concentrates around the correct distribution in terms of Kullback-Leibler (KL) divergence (Ghosal, Ghosh and Van der Vaart, 2000). In practice, however, it is not always the case that these assumptions hold, as sometimes the true distribution is not included in the model, $P^* \notin \mathcal{M}$ (notation explained in Figure 1). Scientists in a variety of fields model, for example, nonlinear relationships as linear, model the errors as 'homoskedastic' while they are not (examples of the latter will be used in this thesis), or model data as independent in time while they are not (in e.g. language models in language learning). If this is the case, the model has been misspecified and can be referred to as wrong, even thought predictions based on the model may still yield reasonable results.

Apart from purely ignorance on the part of the above mentioned researchers, there are a variety of reasons why one would choose a misspecified model deliberately. Usually, the model choice is made on the basis of interpretability of the parameterization, in relation to the specific quantities that play a role in the background of the problem. Another reason for the use of simple but misspecified models is math-

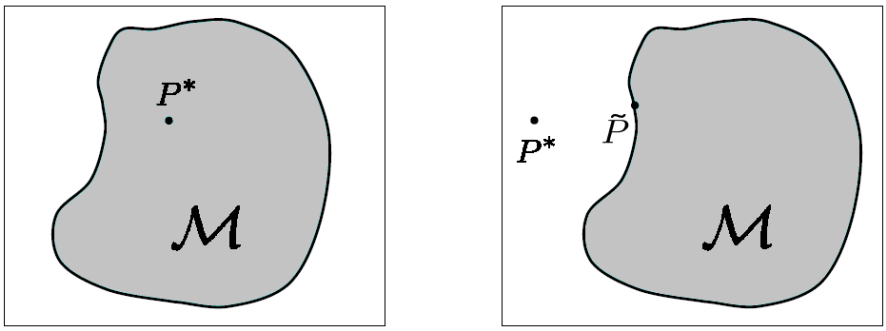

Figure 1: Idealized graphical representation of a model space $\mathcal{M}$, a true distribution P^* and its closest approximation within $\mathcal{M}$, $\tilde{P}$. When the model is true (left), $P^* = \tilde{P} \in \mathcal{M}$ and given enough data, the posterior concentrates around distributions close to $\tilde{P}$. In case $P^* \notin \mathcal{M}$ the model is considered wrong (right). Pictures from Grünwald (2011).

ematical convenience. A small collection of candidate distributions often makes the estimation problem less complicated. Also, computational tractability could be a motive to specify a simple but incorrect model. In practice this results in a trade off. On the one hand, a small, restrictive model leads to interpretability, mathematical simplicity and fast computation, on the other hand, a large model leads to answers that are closer to the truth (Kleijn, 2003).

From a pragmatic point of view the violation of the model assumptions doesn't have to be problematic due to the fact that the predictions these models yield are most of the time reasonably okay. Moreover, Kleijn and Van der Vaart (2006) show theoretically that in certain cases Bayes can perform well even though the model is wrong. That is, it is shown that in such a case the posterior will still concentrate near a point in the support of the prior that is closest to P^* in terms of KL divergence.

There is, however, no guarantee that this is always the case. In fact, other research indicates that Bayes in some settings does go wrong when the model is misspecified. For example, it turns out that forms of Bayesian inference that are often applied to classification problems can be inconsistent under misspecification of the model: there exists a learning problem such that for all amounts of data the generalization error of Bayes remains bounded away from the smallest achievable generalization error. More precisely, a case is exhibited where given a distribution P^*, a model $\mathcal{M}$ with $P^* \notin \mathcal{M}$ and a prior on $\mathcal{M}$ such that the prior puts significant mass on $\tilde{P}$, the best approximation to P^* within the set $\mathcal{M}$, for all large samples the Bayesian posterior puts its mass on a subset of $\mathcal{M}$ that only contains bad approximations to P^* (Grünwald and Langford, 2007). Here 'bad' is both in terms of KL divergence and in terms of classification performance. The posterior distribution in this case fails to

concentrate on the best approximation $\tilde{P}$ of P^*, and instead puts substantial posterior weight on various 'bad' distributions, even in the limit of infinite sample size - the bad distributions that receive substantial posterior mass change as the sample size decreases, but for any given large sample size, with high probability, nearly all mass goes to such bad distributions (Grünwald 2012). This is illustrated in Figure 2. As a consequence, the posterior predictive distribution can yield suboptimal results. In some cases, for large enough samples, predictions of the next outcome based on the Bayesian predictive distribution become worse than predictions based on purely random guessing (Grünwald, 2006). This is of course unsatisfactory and undesirable.

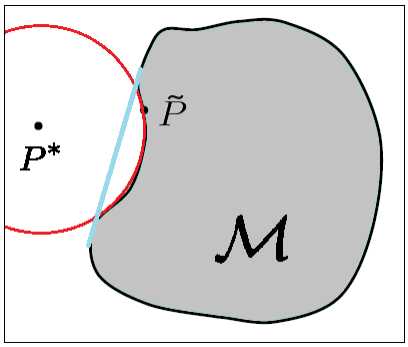

Figure 2: The phenomenon underlying the inconsistency of Bayesian inference under misspecfcation: while one can show that, for large samples, the Bayesian *predictive distribution* (which mixes the distributions in $\mathcal{M}$ according to the posterior) is always at least as close to P^* as $\tilde{P}$ in terms of KL divergence, it may very well be that this predictive distribution is arrived at by a posterior that puts nearly all its weight on distributions that are all much further away from P^* than $\tilde{P}$. Grünwald and Langford (2007) show that for some problems, this happens with high probability at all large samples. Predictions based on such a posterior for standard loss functions used in regression and classification can become quite bad.

One might wonder why it is a problem that Bayesian inference can go wrong when the model is misspecified, given that Bayesian inference has never been designed to work under these conditions (Grünwald and Langford, 2007). The answer to this question is that in practice, Bayesian inference is applied almost all the time under misspecification. It is very hard to avoid misspecification, since the modeler often has no idea about the noise-generating process (Grünwald and Langford, 2007). Knowing this, the next obvious question could be whether it is possible to avoid misspecification altogether.

Misspecification in Bayesian statistics can be caused by a choice for the prior probability that does not contain P^* which results in the fact that there will be no support for a distribution $\tilde{P}$. In other words, the model is misspecified in the Bayesian sense

if there exists a neighbourhood of P^* with prior mass zero (Kleijn, 2003). Examples are when for all P in the model $\mathcal{M}$, the data are i.i.d., whereas under the true P^* this is not the case; or when under all P in the model $\mathcal{M}$, any noise in the observations is independent of the covariates X, whereas in reality, under P^*, it does depend on X. Therefore, the only way to proceed seems to design a prior on all possible distributions. In the case of regression and classification, this would amount to a prior on all possible functions from X to Y and all possible noise rates functions. Now the misspecification problem is solved, because the model is guaranteed to contain the true distribution P^*. However, the cost may be enormous: the model space $\mathcal{M}$ is now much larger and a lot more data may be needed before a reasonable approximation of P^* is learned (Grünwald and Langford, 2007).

1.2 Goal and contents of this thesis

Because the standard Bayesian framework isn't able to give sufficient protection against misspecification in practice, Grünwald (2012) proposed a more robust way of Bayesian inference by equipping the Bayesian likelihood with a learning rate parameter η, in order for the posterior to concentrate at a distribution near P^* at a fast rate, even if the model is wrong. With standard Bayes, when the model is wrong, the Bayesian posterior may fail to concentrate altogether, or may concentrate too slowly, as we will see in later chapters. In other words, less observations should be needed in order for the posterior to put its weight on good approximations of P^* than standard Bayes in the same situation. Moreover, instead of putting a prior distribution on η and integrating it out, as most Bayesians would, the η is determined by minimizing a cumulative loss. This approach, called the Safe Bayesian, has theoretically been proven to work well in the limit of infinite sample size. It is therefore interesting to see how it works in practice and whether it actually outperforms standard Bayesian inference in a variety of settings, including, but not exclusively, settings in which the model has been misspecified.

The main goal of this thesis is to assess and describe some of the situations in which Bayes behaves optimally and, maybe more interestingly, suboptimally in order to examine whether or not the Safe Bayesian algorithm performs better. In order to avoid a biased view towards Safe Bayes, not only situations in which Safe Bayes should theoretically be more accurate than standard Bayes will be explored. Also in this thesis, the theoretical background of the Safe Bayesian algorithm will be explained. As mentioned above, the main aspects of this robust Bayesian approach are:

1. The use of the generalized Bayesian posterior with a learning rate parameter η on the likelihood. This η is used to give more weight to the prior distribution relative to the likelihood if necessary.

2. η is chosen by minimization of a cumulative loss.

Even though determining the optimal η by means of (2) is proven to work asymptotically by Grünwald (2012), it is possible that there are more standard ways to estimate the optimal η successfully. One of these methods is k-fold cross-validation. Therefore, in this thesis, interest is also in the performance of this approach. In order to assess the performances of these robust Bayesian methods, standard Bayes, Safe Bayes and cross-validated Safe Bayes will be tested on several 'toy' classification and 'real data' Lasso regression problems. The classification problems that are applied in this thesis are inspired by (Grünwald and Langford, 2007). They show that in large samples and with infinite sets of classifiers, it is possible that Bayes performs suboptimally. The central question is whether these problems with Bayes also occur in practice where one specifies finite sets of classifiers and has small sample sizes, and, even more interestingly, whether the robust forms of Bayes offer a solution. In the coming chapters we will show that these questions can be answered with a clear 'yes'. In the Lasso regression problems, the robust Bayesian methods are applied to estimate the optimal penalty parameter. It will be shown that the learning rate parameter η in the Safe Bayesian algorithm is related to the penalty parameter in the Lasso, and that standard Bayesian ways of estimating it can yield suboptimal results under model misspecification.

This thesis is composed of five chapters. The current, first chapter, provides an introduction to the subject. The second chapter covers methods of making Bayesian inference more robust against misspecification of the model. In this chapter the Safe Bayes algorithm will be introduced and the general philosophy behind it discussed in detail. Chapter 3 covers the application of standard Bayes and the various robust Bayesian inference[1] methods on a variety of classification experiments. In Section 3.1 it is shown how models consisting of classifiers can be interpreted as probability distributions. The subsequent sections go into detail with respect to various methods of predicting and loss measures. Also in these sections, the performance of these methods on different simulated classification tasks will be exhibited. The fourth chapter is devoted to Lasso regression. In Section 4.1 the relationship between the

[1]We note that 'robust Bayesian inference' is often used for an extension of Bayesian inference in which sets of priors rather than a single prior are used (Berger, 1985). Here we use it in the somewhat different sense of 'general modifications of Bayesian inference' (still using a single prior distribution) that make it more robust under misspecification.

likelihood with learning parameter η, as in Safe Bayes, and the penalty parameter in the Lasso is shown. The subsequent sections cover the differences between the frequentist and Bayesian Lasso. In the final sections of the chapter results will be shown for the frequentist Lasso, Bayesian Lasso, Safe Bayesian Lasso and the cross-validated Bayesian Lasso when applied on various 'real-world' data sets. In the final chapter, a summary of the results and suggestions for further research will be given.

1.3 Technical preliminaries

Before proceeding with the subsequent chapters, we first formally introduce the notation and most important terms used in this thesis.

Let $z^n = (z_1, ..., z_n)$ denote n observations, with each z_i a realization of random variable Z_i, taking values in a sample space Z. With the exception of Section 3.3.3 and 3.4.3 in which the Z_i are dependent, throughout this theses we always assume that the Z_i are i.i.d. according to some distribution P^*.

Probability models and priors. A *model* $\mathcal{M}$ is a set of probability distributions on $\mathcal{Z}$. In this thesis we only consider models such that, according to each $P \in \mathcal{M}$, the Z_i are i.i.d. A model is often written as $\mathcal{M} = \{P_\theta \mid \theta \in \Theta\}$, where Θ is the set of *parameters* corresponding to the model. We identify P_θ with their mass functions p_θ (if $\mathcal{Z}$ is discrete), or their densities, also denoted as p_θ (if $\mathcal{Z}$ is continuous). Note that $p_\theta(z^n) = \prod_{i=1}^n p_\theta(z_i)$, since the data are i.i.d. according to P_θ.

Bayesian inference starts with assuming a *prior distribution* on Θ; we shall denote such a distribution by Π. In case Θ is finite (as in all experiments in Chapter 3), we denote the mass function of Π as π. If Θ is infinite, we shall always assume that the prior has a density, also denoted as π. $\pi(\theta)$ denotes the probability mass/density assigned to θ in advance of any empirical evidence.

Posterior distribution. Given data z^n consisting of $z_1, \ldots, z_n$, the posterior distribution $\pi(\theta \mid z^n)$ summarizes the current state of knowledge about all the uncertain quantities θ in Bayesian inference. Analytically, the posterior density is proportional to the product of the prior density $\pi(\theta)$ and the likelihood $p(z^n \mid \theta)$, such that

$$\pi(\theta \mid z^n) = \frac{p(z^n \mid \theta)\, \pi(\theta)}{\int_\Theta p(z^n \mid \theta)\, \pi(\theta)\, d\theta}. \tag{1.1}$$

Posterior predictive distribution. Suppose there is some unobserved data z_{n+1} that has to be predicted based on z^n. The posterior predictive distribution does

exactly that, as it describes the distribution of z_{n+1} conditional on z^n, such that

$$p(z_{n+1} \mid z^n) = \int_\Theta p(z_{n+1}, \theta \mid z^n)\, d\theta \tag{1.2}$$

$$= \int_\Theta p(z_{n+1} \mid z^n, \theta)\, \pi(\theta \mid z^n)\, d\theta. \tag{1.3}$$

Since we invariably assume that data are i.i.d. under all p_θ with $\theta \in \Theta$, the z^n and z_{n+1} are conditional independent, so the posterior predictive distribution can be expressed as

$$p(z_{n+1} \mid z^n) = \int_\Theta p(z_{n+1} \mid \theta)\, \pi(\theta \mid z^n)\, d\theta. \tag{1.4}$$

In this thesis we often work with conditional models, in which each z_i can be written as (x_i, y_i). Here each $x_i \in \mathcal{X}$ is a 'covariate' or 'input vector', and y_i is the variable to be predicted; in classification problems, y_i takes values in a finite set (we only consider binary classification, in which it takes values in $\{0, 1\}$); in regression problems, y_i takes values in $\mathbb{R}$. The distributions in $\mathcal{M}$ are then just conditional distributions, which only specify distributions of y_i given x_i. All formulas given above can be trivially adjusted to this case; we omit the details. Note however, that the data generating process P^* is always viewed as a distribution under which $Z_i = (X_i, Y_i)$ are i.i.d.; hence the X-values are also sampled from P^*, and not set by the experimenter, corresponding to what is called a 'random design' in statistics.

Randomized vs. mixed prediction. In practical problems such as regression and classification, the goal is to use the available data to make good predictions of future data from the same source. After having observed some data $z^n = (x_1, y_1), \ldots, (x_n, y_n)$, one would like to use the Bayesian posterior to make predictions of a new Y given a new X. Here the quality of the predictions is measured using some loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. Here $\ell(y, \delta(x))$ denotes the loss when the actual outcome is y and the prediction as a function of x is $\delta(x)$. In classification we have $\mathcal{Y} = \{0, 1\}$ and we will use the 0/1-loss, $\ell(y, \delta(x)) = 1$ if $y \neq \delta(x)$ and 0 otherwise; in regression we have $\mathcal{Y} = \mathbb{R}$ and we use the squared loss, $\ell(y, \delta(x)) = (y - \delta(x))^2$.

The standard way of arriving at a prediction based on a probability density p is to use the so-called *Bayes act* for p, which is the action that minimizes expected loss according to p. If $p(y \mid x)$ is a conditional density for y given x, this Bayes act is

given by

$$\delta_p(x) = \arg\min_{a \in \mathcal{Y}} \mathrm{E}_{y \sim p|x} \left[\ell\left(y, \delta\left(x\right)\right) \right]. \tag{1.5}$$

If the prediction is made based on the posterior predictive distribution as given by (1.4) above, we can rewrite this as

$$\begin{aligned} \delta_p(x) &= \arg\min_{a \in \mathcal{Y}} \mathrm{E}_{Y \sim p|X} \left[\ell\left(Y, \delta\left(X\right)\right) \right] && (1.6) \\ &= \arg\min_{a \in \mathcal{Y}} \mathrm{E}_{\theta \sim \Pi|z^n} \mathrm{E}_{Y \sim p_{\cdot|X,\theta}} \left[\ell\left(Y, \delta\left(X\right)\right) \right]. && (1.7) \end{aligned}$$

This is the standard Bayesian way of doing prediction. We will refer to it as *mixed* prediction from now on, reflecting that it is the optimal prediction according to a *mixture* of the p_θ. This is in contrast to a second, less common way to use the Bayesian posterior to arrive at a prediction, so-called *randomized* prediction, which we encounter in a moment. Note that the *actual* loss incurred if mixed prediction is used on a new data point (x_{n+1}, y_{n+1}) is given by

$$\text{mix loss}_{n+1} = \ell\left(y_{n+1}, \delta_p\left(x_{n+1}\right)\right), \tag{1.8}$$

where p is the posterior predictive distribution as above.

Whereas in mixed prediction, we take a mixture and then take the optimal action for that mixture, in *randomized prediction* (a method mostly used within the so-called PAC-Bayesian literature (McAllester, 2003)), we sample ("randomize") from the posterior, and then take the optimal prediction relative to the distribution we sampled. We measure the loss as the expected loss (according to our own randomization) of this procedure. Thus we get:

$$\text{randomized loss}_{n+1} = \mathrm{E}_{\theta \sim \Pi|z^n} \, \ell\left(y_{n+1}, \delta_{p_\theta}\left(x_{n+1}\right)\right). \tag{1.9}$$

Let us consider a simple example for the case of classification. Suppose we have a posterior on just two distributions, p_1 and p_2 and suppose that $\Pi\left(1 \mid z^n\right) = 0.51$, $\Pi\left(2 \mid z^n\right) = 0.49$, so that the posteriors are almost equal. Suppose that given a new x value, $P_1\left(Y = 1 \mid X = x\right) = p_1\left(1 \mid x\right) = 1$, $p_2\left(1 \mid x\right) = 0$. Then according to the predictive distribution, the probability of 1 given x is larger than the probability of 0, and according to the (standard Bayesian) mixed prediction, we should predict 1. The mix loss will then be 1 if the actual outcome is 0 and 0 if it is 1. If we predict randomized, then we measure our loss by the expected error we make if we draw from the posterior and use the drawn distribution to make a prediction. Thus, if the actual outcome is 0 the randomized loss will be 0.51, if it is 1 the randomized loss is 0.49.

Kullback-Leibler divergence. When comparing two probability distributions, identified by their densities or mass functions, say p_θ and $p_{\theta'}$, the similarity or dissimilarity between them has to be quantified. In this thesis the Kullback-Leibler (KL) divergence is used, because of its intimate connection to the frequentist analysis of Bayesian procedures: for any θ, θ', if the data are drawn i.i.d. from p_θ, then the log of the posterior probability ratio $\pi(\theta \mid z^n) / \pi(\theta' \mid z^n)$ divided by n, converges, with P^*-probability 1, to the KL divergence between p_θ and $p_{\theta'}$. The KL divergence from probability distribution θ to θ' is now given by

$$D_{\mathrm{KL}}(p_\theta \| p_{\theta'}) = \mathrm{E}_{Z_1 \sim p_\theta}\left[-\log \frac{p_\theta(Z_1)}{p_{\theta'}(Z_1)}\right]. \tag{1.10}$$

Because it is inconvenient to repeatedly explicitly state that a certain probability distribution is close to, or distant from, some other probability distribution in terms of KL divergence, every time such a statement is made without notion of this measure, it should still be regarded in terms of the KL divergence.

2 The Safe Bayesian

In this chapter the Safe Bayes algorithm will be introduced together with its theoretical background. First, in Section 2.1, related work will be discussed. Section 2.2 covers the theory behind the Safe Bayesian algorithm, whereas in Section 2.3 the algorithm itself is presented.

2.1 Related work

Equipping the Bayesian likelihood with a learning rate parameter in order to increase the robustness of the posterior is not an entirely new idea. The following papers all introduce learning rates that are similar to Grünwald's in that they are single scalar parameters that control the relative weight of the data and a prior regularization term.

1. L1 and L2 regularization: in particular the Lasso and Ridge Regression can be thought of this way (Tibshirani, 1996).
2. Sequential prediction: Vovk's (1990) aggregating algorithm and Freund and Shapire's (1997) Hedge algorithm are widely used tools in sequential on-line prediction problems. Both algorithms closely resemble Bayesian prediction, and both of these involve a learning rate which regulates the importance of the prior (Cesa-Bianchi and Lugosi, 2006).
3. The generalized Bayesian posterior as introduced by Zhang (2004), following related developments by Walker and Hjorth (2001). Early ideas in this direction can be found in Barron and Cover (1991). Grünwald (2012) can be seen as a direct extension of Zhang (2004).

First we will go somewhat deeper into an example of 3., and then an example of 2. is given. Detailed information about the Lasso can be found in Chapter 4.

Zhang's (and Grünwald's) Generalized Bayesian Posterior
Zhang came up with a generalized form of the Bayesian posterior in such a way that convergence of the posterior near the true distribution when the sample size is large only relies on the existence of some prior mass in a small neighborhood around the true distribution. This is in contrast to the behavior of the standard Bayes posterior, namely that even though one puts a positive prior mass around the true distribution, the posterior may concentrate arbitrarily slowly when there exist undesirable prior structures far away from the true distribution (Zhang, 2004).

His solution to guarantee convergence near P^* is the so called α-Bayesian method, and revolves around a generalized Bayesian posterior identical to the one used in Grünwalds Safe Bayesian algorithm. Since in this paper we denote the learning rate by η rather than α, we will henceforth refer to it as the η-Bayesian method. The generalized Bayesian posterior is then of the form $\pi(\cdot \mid z^n, \eta)$ with $\eta \in (0, 1]$, such that

$$\pi(\theta \mid z^n, \eta) = \frac{p^\eta(z^n \mid \theta)\,\pi(\theta)}{\int_\Theta p^\eta(z^n \mid \theta)\,\pi(\theta)\,d(\theta)}, \tag{2.1}$$

with $\theta \in \Theta$, z^n representing the data and $p^\eta(z^n \mid \theta)$ denoting the likelihood. The η-Bayesian posterior equals the standard Bayesian posterior when $\eta = 1$. Zhang shows that, if the model is correct, then, under standard conditions on the prior, the η-Bayesian method will concentrate in close neighborhoods of the true distribution at fast rates for every $\eta < 1$, but for $\eta = 1$ may be arbitrarily slow. Related observations were made by Barron and Cover (1991).

In despite of the promising properties of the η-Bayesian method, how to choose the optimal η for a given sample is not clear. Of course, the optimal choice for η depends on the true distribution P^*. However, P^* is unknown, and predicting by marginalizing out η, as well as picking the learning rate that maximizes the Bayesian marginal likelihood of the data, is not a realistic solution if the model is misspecified (Grünwald and Langford, 2007). This is because both marginalizing out and picking the most likely learning rate require that the model is correct and can give misleading results if it isn't, as both involve the probability of the data under the assumptions of the model. This is of course a little ironic given the purpose of the generalized Bayesian posterior.

On-line sequential prediction

Apart from the work of Zhang, there is more research that relates to the Safe Bayesian approach. Devaine, Gaillard, Goude and Stoltz (2013) use a method to choose a learning rate that closely resembles the way in which the learning rate for the generalized Bayesian posterior in the Safe Bayesian framework is chosen. In their setting, the paradigm of *prediction with expert advice*, one has to make predictions *on-line* by using information that is provided by a number of *experts*. This means that the predicting proceeds in rounds in such a way that per round the separate predictions of the experts are announced first, after which one has to make a prediction. The discrepancy between a prediction and an outcome is measured by a loss function, and losses add up between rounds. Finally, the goal is to minimize the regret, which is the difference between one's cumulative loss and the cumulative loss of the expert that, after T rounds, turns out to be the best so far (Cesa-Bianchi

and Lugosi, 2006). In order to minimize the regret, Devaine et al. use a weighting algorithm in which the weights depend on the past performance of the experts and a learning rate η, with the tuning of the latter based on the cumulative loss at round T_{i-1}. More specifically, η is chosen from a finite set $\{\eta_1, \ldots, \eta_p\} \in \mathcal{S}$, for each of which the algorithm is run in parallel and the learning rate is picked such that $\hat{\eta} = \arg\min_{\eta \in \mathcal{S}} \{\text{regret}_{T_{i-1}}(\eta)\}$. In order to test the algorithm in practice it has been applied to a real-world problem of predicting with expert advice, namely the one-day-ahead forecasting of electricity consumption in France and Slovakia based on a number of experts. The results indicated that algorithm performed very well in both countries.

As will be seen in Section 2.3, this way of estimating the optimal η is analogous to the the way the Safe Bayesian algorithm chooses η. One important difference between the two methods is, however, that Devaine et al. did not prove the optimality of their method, while Grünwald proved the effect of the Safe Bayesian in the limit of infinite sample size. The second important difference is the fact that the work of Devaine et al. is only applicable to *prediction with expert advice* settings, whereas the Safe Bayesian can be applied to problems in various contexts including the more common statistical environments.

2.2 Mixability gap

The solution proposed by Grünwald (2012) to last section's problem of selecting the optimal learning rate can be expressed in terms of what is called the *mixability gap*. Because this concept is rather fundamental to the Safe Bayesian approach, its formal definition as well as the definitions of concepts that are related will be presented next.

We first need to make a little detour and introduce the basic idea of 'statistical learning', where the goal is to learn from the data a function f, taken from some set $\mathcal{F}$, to predict a random variable Y given another random variable X, as in classification and regression problems. Since the functions f are not probability distributions, such an approach is rather different from Bayesian inference, but we reconnect it to Bayesian inference later.

Statistical learning

Given a random sample z^n consisting of $(x_1, y_1), \ldots, (x_n, y_n)$ i.i.d. according to some distribution P^*. By default, the symbol E denotes expectation under P^*; if the expectation is over some other distribution (such as a posterior) we shall

explicitly denote this in the subscript of E. Suppose there is a learning algorithm that given z^n selects a function f from a set of functions $\mathcal{F}$. The function selected by the algorithm will be referred to as $\hat{f}$. Now the quality of $\hat{f}$ can be measured by its excess risk. Given a loss function $\ell \in L : \mathcal{Y} \times \hat{\mathcal{Y}} \to \mathbb{R}$, the excess risk is the expectation of the loss $\ell(Y, \hat{f}(X))$ minus the expected loss of the best prediction function $f^* \in \mathcal{F}$. More formally,

$$\mathrm{E}_{\hat{f}} \left[\mathrm{E}_{X,Y}[\ell(Y, \hat{f}(X))] - \mathrm{E}_{X,Y}[\ell(Y, f^*(X))] \right], \tag{2.2}$$

where f^* is given by $\arg\min_{f \in \mathcal{F}} \mathrm{E}_{X,Y \sim P^*} \ell(Y, f(X))$. Note that the outer expectation is over the sample z^n of size n, which determines the chosen $\hat{f}$; the inner expectation is over a single 'test' example. A typical example of a loss function is the squared loss, $\ell(Y, f(X)) = (Y - f(X))^2$.

As a statistician when facing a problem in which one has to predict a new y_{n+1} given x_{n+1}, the best one can do is picking an $\hat{f}$ based on z^n such that the excess risk is minimal. Even though there are no guarantees that an $\hat{f}$ near this optimal f^* will be chosen, there are usually guarantees about the performance of $\hat{f}$ in the worst case over all P^* under which the data are i.i.d., assuming that $\hat{f}$ is chosen by a clever algorithm (Hastie, Tibshirani and Friedman, 2009).

Mix loss, randomized loss and mixability gap

Now we return to our generalization of Bayesian inference and introduce a few fundamental concepts; we will connect these to statistical learning underneath Equation (2.9). These concepts, that are related to the excess risk, are *excess mix loss*, *excess randomized loss* and the *mixability gap*, which will be presented based on their definitions in Grünwald (2012). Now consider a model $\mathcal{M} = \{p_\theta \mid \theta \in \Theta\}$ of probability distributions, and assume $Z_1, Z_2, \ldots$ are i.i.d. under all $P \in \mathcal{M}$ and also under the 'true' distribution P^*. Let $q = p_{\tilde{\theta}}$ be the density of the best approximating probability distribution of P^*, i.e. $\tilde{\theta}$ achieves $\min_{\theta \in \Theta} D(P^* \| P_\theta)$.

Let the generalized Bayesian posterior be as in (2.1). The generalized Bayesian marginal distribution can now be expressed as

$$p_{\text{Bayes}}(z^n \mid \eta) = \mathrm{E}_{\theta \sim \Pi}[\, p^\eta(z^n \mid \theta)] \tag{2.3}$$

whereas the generalized Bayesian predictive distribution is given by

$$p_{\text{Bayes}}\left(z_i \mid z^{i-1}, \eta\right) = \frac{p_{\text{Bayes}}(z_i \mid \eta)}{p_{\text{Bayes}}(z^{i-1} \mid \eta)} = \mathrm{E}_{\theta \sim \Pi \mid z^{i-1}, \eta}\left[\, p^\eta(z_i \mid \theta)\right]. \tag{2.4}$$

It is now possible to define the expected loss (in this case in terms of logarithmic loss) that is made when predicting with the generalized Bayesian posterior as a function of η at observation n relative to the expected loss that would have been made when predicting with q. Because this *excess loss* (in excess of q) is obtained by mixing the posterior (which is the standard way of Bayesian inference), it can be called *excess mix loss* (EML). At observation i, the EML is given by

$$\mathrm{EML}_i(\eta) = -\frac{1}{\eta} \log \frac{p_{\mathrm{Bayes}}\left(z_i \mid z^{i-1}, \eta\right)}{q^\eta(z_i)} \tag{2.5}$$

$$= -\frac{1}{\eta} \log \mathrm{E}_{\theta \sim \Pi \mid z^{i-1}, \eta} \frac{p^\eta(z_i, \theta, \eta)}{q^\eta(z_i)} \tag{2.6}$$

such that the cumulative excess mix loss (CEML) can be expressed as

$$\mathrm{CEML}_n(\eta) = -\frac{1}{\eta} \log \frac{p_{\mathrm{Bayes}}(z^n \mid \eta)}{q^\eta(z^n)} = \frac{1}{\eta} \sum_{i=1}^{n} -\log \frac{p_{\mathrm{Bayes}}\left(z_i \mid z^{i-1}, \eta\right)}{q^\eta(z_i)}. \tag{2.7}$$

For η is 1, the CEML equals the standard Bayesian loss when predicting with p_{Bayes} relative to the optimal distribution q.

Now the fundamental idea behind the Safe Bayesian approach stems from the fact that there exist very general convergence theorems about Bayesian prediction with log-loss that hold even if the model is completely wrong. These bounds, going back to Barron (1998) and Grünwald (2007), say that the expectation over the cumulative expected excess mix log-loss (i.e. cumulative expected excess mix log-loss *risk*) for standard Bayes is bounded by a sublinear term, the size of which depends on how much mass the prior gives to distributions close to the best approximation of P^* within the model $\mathcal{M}$. For example, if $\mathcal{M}$ is countably infinite and the prior gives positive mass $\pi(\tilde{\theta})$ to the best approximation $\tilde{\theta}$, then the term is of order $-\log \pi(\tilde{\theta})$. This implies that for most individual outcomes $z_1, \ldots, z_n$, the logarithmic expected excess mix risk must be smaller than $-\frac{1}{n} \log \pi(\tilde{\theta})$. As a consequence, at most outcomes the excess mix risk is bounded by $\mathrm{O}(\frac{1}{n})$, which is quite small.

These convergence theorems continue to hold for $\eta < 1$, getting slightly weaker as η gets smaller. For example, in the countable example above, the bound becomes $-\frac{1}{\eta} \log \pi(\tilde{\theta})$, which comes down to $-\frac{1}{n\eta} \log \pi(\tilde{\theta})$ per outcome for $\eta \in (0, 1]$. While these bounds hold even if the model is wrong, there are, however, two serious problems with them.

1. The bounds only hold for the predictive distribution, not for the MAP distribution. It may very well be the case that the predictive distribution works

well for log-loss by mixing bad distributions together into a good prediction, as shown earlier in Chapter 1 (Figure 2).

2. The bounds only hold for the log-loss function. Intuitively, this is because Bayes can be interpreted as trying to find the distribution that has the smallest expected log-loss under the true distribution.

To demonstrate this, an example will be given where predictions are made based on a loss function other than the log-loss. As will be shown in Chapter 4, the Lasso and Ridge Regression can interpreted as Bayesian MAP prediction with a probability model in which the errors are normally distributed, and the priors are Laplacian and Gaussian, respectively. This suggests that for such models, the likelihood can be defined as

$$p^{\eta}\left(z^{n} \mid \theta\right) \propto \mathrm{e}^{-\eta \sum_{i=1}^{n} \ell\left(y_{i}, f_{\theta}\left(x_{i}\right)\right)} \tag{2.8}$$

such that the generalized posterior becomes

$$\pi\left(\theta \mid z^{n}, \eta\right) \propto \frac{\mathrm{e}^{-\eta \sum_{i=1}^{n} \ell\left(y_{i}, f_{\theta}\left(x_{i}\right)\right)} \pi\left(\theta\right)}{\int_{\Theta} \mathrm{e}^{-\eta \sum_{i=1}^{n} \ell\left(y_{i}, f_{\theta}\left(x_{i}\right)\right)} \pi\left(\theta\right) d\left(\theta\right)}. \tag{2.9}$$

For the Lasso and ridge regression, ℓ is the squared loss, that is, $\ell\left(y_{i}, f_{\theta}\left(x_{i}\right)\right) = \left(y_{i} - f_{\theta}\left(x_{i}\right)\right)^{2}$. But for other settings (e.g. classification), ℓ will be something different. With such definitions, the log-loss for any individual $p^{\eta}\left(z^{n} \mid \theta\right)$ will be equal to η times the loss as measured by ℓ of the corresponding function f_{θ} on the same sample. Therefore the generalized Bayesian posterior with $\eta = 1$ as above, which gives small log-loss, should also give small ℓ-loss. However, the predictive distribution p_{Bayes} as defined above in (2.4) is a mixture of $p^{\eta}\left(z^{n} \mid \theta\right)$ for different θ's and not a single $p^{\eta}\left(z^{n} \mid \theta\right)$. More precisely, it will be of the form $\int_{\Theta} e^{-\eta\, \ell(y, f_{\theta}(x))}\, w(\theta)\, d\theta$ for some w depending on the past data. Therefore, it is not clear how this mixture could be used to make predictions for the original loss function ℓ. Again, it might very well be that the mixture puts most of its weight on two very bad predictors, which then together make a good predictor for mix loss (Grünwald and Langford (2007) show that this actually can happen). But the important question is: does this mixture also make a good predictor for the loss of interest ℓ?

It seems that $p_{\text{Bayes}}\left(z_{i} \mid z^{i-1}\right)$ has to be turned into a prediction for the loss function of interest. This can be done in the standard Bayesian way (taking the prediction with minimal posterior-expected loss) but Grünwald and Langford show that this also can lead to very bad losses if the model is wrong. If the loss function has a

special property called 'mixability' (Vovk 1990), then by taking a small enough η, we know that, no matter what the posterior is, there exists a predictor for ℓ which has loss that is smaller than the mix loss. Then we can simply use that predictor, and any bound on the mix loss transfers again to our loss function of interest ℓ. But the 0/1-loss is not mixable, and the squared loss is only mixable if we know a priori that the range of the y-values that we will observe falls in a bounded interval. In both cases, we would like to do something else. The idea behind the safe Bayesian algorithm is that we can look at the randomized instead of the mix loss.

In the same manner as was done previously in the case of the mix loss, the *excess randomized loss* will now be defined. This is the expected loss that is made when predicting 'randomized' according to the generalized Bayesian posterior relative to the expected loss that would have been made when predicting with q. This excess randomized loss (ERL) at observation i is given by

$$\mathrm{ERL}_i(\eta) = \mathrm{E}_{\theta\sim\Pi|z^{i-1},\eta}\left[-\log\frac{p(z_i \mid \theta)}{q(z_i)}\right] \tag{2.10}$$

$$= \frac{1}{\eta}\mathrm{E}_{\theta\sim\Pi|z^{i-1},\eta}\left[-\log\frac{p^{\eta}(z_i \mid \theta)}{q^{\eta}(z_i)}\right]. \tag{2.11}$$

Note the differences between (2.11) and (2.6). The expectation over the posterior is now outside the logarithm. The cumulative excess randomized loss (CERL) can be expressed as

$$\mathrm{CERL}_n(\eta) = \sum_{i=1}^{n}\mathrm{E}_{\theta\sim\Pi|z^{i-1},\eta}\left[-\log\frac{p(z_i \mid \theta)}{q(z_i)}\right]. \tag{2.12}$$

The $\mathrm{CERL}_n(\eta)$ relates to the $\mathrm{CEML}_n(\eta)$ such that for any η in [0,1] and n in $[0,\infty]$ $\mathrm{CEML}_n(\eta) \leq \mathrm{CERL}_n(\eta)$ (Grünwald, 2012). The difference between the two expected losses can be quantified and is called the *mixability gap*. Because the difference doesn't depend on q it can be taken out of the equations, such that the expected mixability gap can be expressed as

$$\mathrm{MG}_n(\eta) = \mathrm{E}_{z^n}\left[\sum_{i=1}^{n}\mathrm{E}_{\theta\sim\Pi|z^{i-1},\eta}\left[-\log p(z_i \mid \theta)\right] + \frac{1}{\eta}\log p_{\mathrm{Bayes}}\left(z_i \mid z^{i-1},\eta\right)\right], \tag{2.13}$$

and it has the following interesting property that can be used in the Safe Bayesian framework. Namely, if the Bayesian posterior is relatively concentrated on the majority of the data, the mixability gap is small, whereas when the posterior is more spread out, the mixability gap becomes larger. By 'is concentrated on the majority

of the data', it is meant that at most initial segments of the data, i.e. for most $i < n$, $\Pi \mid z^{i-1}, \eta$ puts most of its mass on a small set of distributions that are all close to each other in KL divergence. Because a large mixability gap indicates that that the Bayesian predictive distribution is mixing various quite different distributions (which could lead to suboptimal performance), this means that the mixability gap provides information in the search of the optimal η in the generalized posterior.

Assuming we are allowed to do randomized prediction (which we do assume in this thesis), any cumulative excess risk bound on the log-loss obtained by randomizing transfers directly to a cumulative excess risk bound on the loss function of interest ℓ: The $-\log p\left(z_i \mid \theta\right)$ are all equal to $\eta \cdot \ell\left(y_i, f_\theta(x_i)\right)$, so the expected log-loss we make is again a linear function of the expected loss in terms of ℓ.

From Grünwald (2012) we know that, for all η smaller than some 'critical' η^*, the expected mixability gap will be smaller or equal than the cumulative excess mix risk. Hence, if we pick small enough η^*, the expected mixability gap will be of the same order as the excess risk we incur anyway, and for which we have good bounds even if the model is wrong (as was mentioned under (2.7)). Then the cumulative performance of generalized Bayes in terms of the loss function of interest will be quite good in expectation.

All this suggests that it makes sense to find the η from the data for which the expected cumulative randomized log-loss is smallest. If we use models of the form (2.8), then this will be the same η as the η for which the expected cumulative randomized loss in terms of the function ℓ is smallest. That such an approach would really work for large samples is of course nontrivial, and is proved by Grünwald (2012).

2.3 The Safe Bayesian algorithm

The Safe Bayesian approach first mentioned in Chapter 1, minimizes the cumulative randomized log-loss (Cu-R-L-L), which is

$$\text{Cu-R-L-L}(\eta) = \sum_{i=1}^{n} \mathrm{E}_{\theta \sim \Pi \mid z^{i-1}, \eta}[-\log p\left(z_i \mid \theta\right)] \tag{2.14}$$

$$= \sum_{i=1}^{n} \text{randomized loss}_i\left(\eta\right), \tag{2.15}$$

where mix $\text{loss}_i(\eta)$ is the mix loss as defined in (1.8), using the η-generalized rather than the standard posterior.

For computational reasons, the Safe Bayesian algorithm can't be applied to all $\eta \in (0, 1]$. Therefore a set of candidate η's have to be selected. Three factors play a role in selecting the candidates, namely the range of the η's, the number of η's and the way they are spread out over the range. Grünwald (2012) shows that in practice, one can safely assume that the minimal candidate η is not smaller than $\frac{1}{n}$. To keep the number of candidate η's limited, it has been opted to pick candidate η's from the sequence $\{2^{-0}, 2^{-1}, 2^{-2}, \ldots\}$ such that the smallest η is greater than the previously mentioned minimum of $\frac{1}{n}$. In this way the Safe Bayesian algorithm has relatively a lot of options comparatively close to an η of 1, the standard Bayesian posterior, but keeps the broad range of η's and is therefore guarded against situations in which it would be necessary to pick a rather small η.

As all the necessary elements of the Safe Bayesian framework have been introduced, it is now possible to present the algorithm itself. Given a random sample z^n i.i.d. according to a distribution P^*, let the generalized Bayesian posterior, as shown in Section 2.2, be denoted by $\Pi \mid z^n, \eta$ and the output distribution be a so-called *Cesàro-averaged* posterior referred to as $\text{CES}(\Pi \mid z^n, \hat{\eta}) = \frac{1}{n} \sum_{i=1}^{n} \Pi \mid z^i, \hat{\eta}$. Then the Safe Bayesian algorithm, similar to the algorithm of Grünwald (2012), is shown in Algorithm 1.

Algorithm 1: The Safe Bayesian

Input : Data $z_1, \ldots, z_n$, model $\{p_\theta \mid \theta \in \Theta\}$, prior on Θ.

Output: Distribution on Θ.

$\mathcal{S}_n = \left\{2^{-0}, 2^{-1}, 2^{-2}, \ldots\right\} \cap \left\{\eta : \eta > \frac{1}{n}\right\}$;

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

  $s_\eta = 0$;

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

    Compute generalized Bayes posterior $\Pi \mid z^{i-1}, \eta$;

    Calculate expected randomized log-loss by predicting actual next outcome and add up to previous losses:

    $r = \mathrm{E}_{\theta \sim \Pi \mid z^{i-1}, \eta}\left[-\log p\left(z_i \mid \theta\right)\right]$;

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

  end

end

Choose $\hat{\eta} = \arg\min_{\eta \in \mathcal{S}_n}\left\{s_\eta\right\}$. If minimum achieved by multiple $\eta \in \mathcal{S}_n$, pick largest ;

Output distribution $= \mathrm{CES}\left(\Pi \mid z^n, \hat{\eta}\right) = \frac{1}{n}\sum_{i=1}^{n} \Pi \mid z^i, \hat{\eta}$;

In the Safe Bayesian algorithm the log-loss is used. If, as in this thesis, we use models of the form (2.8), then r in Algorithm 1 is in fact set to $\mathrm{E}_{\theta \sim \Pi \mid z^{i-1}, \eta}\left[\ell\left(y_i, f_\theta\left(x_i\right)\right)\right]$. Thus, in such cases, we can think of Safe Bayes as selecting the η for which sequential prediction based on randomizing by the generalized posterior gives the smallest cumulative loss in terms of the loss function ℓ of interest.

Variations of the Safe Bayesian algorithm

Safe Bayesian inference differs from standard Bayesian inference in three ways.

1. A generalized posterior, with potentially $\eta < 1$, is used.
2. The η is chosen that minimizes the cumulative expected loss based on sequentially predicting by drawing from the η-generalized posterior.
3. The final prediction is made by a Cesàro-average of posteriors, rather than the final posterior.

It is now interesting to test not just the Safe Bayesian algorithm as presented here, but also variations: for example, although the theoretical results by Grünwald (2010) require the use of the somewhat awkward Cesàro-average[2], in practice the algorithm

[2] In theoretical papers, such Cesàro-averages are more often taken though; see for example Barron (1998), Yang (2000).

might also work just fine if one just uses the final posterior. Similarly, while theoretically randomization rather than mixing is required, one could try to mix instead and see what happens in practice; and finally, in terms of determining η by a cumulative sequential loss, one may use some form of cross-validation instead. Indeed, as mentioned in the first section of this chapter, the power of the Safe Bayesian framework is its employability in multiple statistical contexts, which is mainly due to the sequential nature of estimating the optimal learning rate parameter. This versatility, however, sometimes means that there could be more specialized methods in the specific areas that work even better. Looking at the batch scenario for example, it could very well be that a reasonably straightforward method like cross-validation happens to yield favorable results in terms of selecting the correct learning rate. Even though this hybrid form of Bayesian inference may be considered even more distant from classical Bayes than Safe Bayes, it potentially has some advantages. If in a batch scenario the sample size is rather large, it is conceivable that sequentially predicting with a full Bayesian posterior will take a lot of computing time. In these situations it could very well be that k-fold cross-validation is in practice more convenient to use. Whether this is indeed the case will be tested in the subsequent chapters. Next to testing what happens if we choose η in such a different way, we will also see whether the Cesàro-averaging and the randomization rather than mixing of the Safe Bayesian algorithm are really helpful in practice.

3 Safe Bayesian classification

The current chapter focuses on the application of the Safe Bayesian algorithm on various classification problems. The potential risk of applying standard Bayesian inference to situations wherein the model has been misspecified will be demonstrated. Moreover, it will be shown that in these settings the Safe Bayesian approach, and other robust methods, have beneficial properties. It is however not the goal of this chapter to empirically 'prove' that robust forms of Bayesian inference are to be preferred over standard Bayes in all circumstances. The goal is rather to show that there exist situations that ask for methods that are somewhat more robust against misspecification of the model than standard Bayes.

The classification problems used in this chapter stem from the machine learning paradigm and therefore differ somewhat from what is common in the statistical setting. More specifically, models are specified consisting of a certain number of nonprobabilistic classifiers. It is shown in Section 3.1 how these classifiers can be interpreted as conditional probability distributions in a Bayesian way and why it makes sense to apply Bayesian inference to these models. In Section 3.2 the procedure is explained that is used in the simulations. Section 3.3 covers the application of standard Bayesian and Safe Bayesian inference on simulated classification problems while using a fixed-probability likelihood. In Section 3.4 similar simulations are used to compare the performances of the different forms of inference but now using a more Bayesian approach by applying a likelihood that depends on the Laplace smoothed error rate, hereafter referred to as *Laplace likelihood*. Finally, in Section 3.5, the results will be discussed.

3.1 Bayesian interpretation of classifier models

Given some data z^n consisting of $(x_1, y_1), \ldots, (x_n, y_n) \sim P^*$ with $y_i \in \{0, 1\}$. In supervised classification the goal then is to find a function $f \in \mathcal{F}$, where $\mathcal{F}$ is a finite or countably infinite set of of functions that map features x to class labels y, such that $y = f(x)$, with the aim of minimizing the expected misclassification error, also called simply the *error rate* or the *error probability* (in the machine learning

paradigm also referred to as generalization error),

$$e_{P^*} = \mathrm{E}_{X,Y\sim P^*}\left[I_{Y\neq f(X)}\right] \tag{3.1}$$

$$= \mathrm{E}_{X,Y\sim P^*}\left[\ell_{01}\left(Y, f(X)\right)\right] \tag{3.2}$$

$$= P^*\left(Y \neq f(X)\right), \tag{3.3}$$

where I is the indicator function and ℓ_{01} is the 0/1-loss function, $\ell_{01}(x, y) = I_{x\neq y}$. Because f classifies the data it is sometimes called a *classifier*. Therefore in this chapter the latter terminology is used, such that the symbol c is used instead of f, and $\mathcal{C}$ instead of $\mathcal{F}$. We will always work with finite sets of classifiers, and write $\mathcal{C} = \{c_1, ..., c_q\}$.

Instead of picking the optimal classifier c_m that minimizes (3.1) over $\mathcal{C}$ by means of loss minimization, it is also possible to apply Bayesian inference to this problem in order to obtain a posterior distribution over $\mathcal{C}$. Suppose one has specified a model $\mathcal{C}$ consisting of a number of classifiers $c_1, c_2, \ldots, c_q \in \mathcal{C}$ that each make predictions $\{0, 1\}$ of y given x. Bayesian inference can now be applied by putting a prior distribution on the classifiers $\pi(\mathcal{C})$ and using Bayes rule to estimate the posterior. However, $\pi(\mathcal{C})$ is not a Bayesian prior in the conventional sense because classifiers do not induce a measure over the data (Grünwald and Langford, 2007). In order to apply Bayesian inference, $\mathcal{C}$ has to be converted into a corresponding set of distributions $\mathcal{M}$ such that $\pi(\mathcal{C})$ becomes a conventional Bayesian prior on $\mathcal{M}$.

Because in the current study only classification is done on simulations in which $y_i \in \{0, 1\}$, (i.e., binary classification), the transformation of $\mathcal{C}$ into $\mathcal{M}$ is reasonably straightforward. Let the misclassification rate of classifier c on z^n be denoted by

$$\hat{e}(c) = \frac{1}{n}\sum_{i=1}^{n} I_{y_i\neq c(x_i)}. \tag{3.4}$$

Let θ be the error rate of classifier c. It is possible to convert $\mathcal{C}$ to $\mathcal{M}$ because each $c \in \mathcal{C}$ can be expressed in the form of a conditional probability distribution $p_{c,\theta}$ as

$$p_{c,\theta}\left(y^n \mid x^n\right) = \theta^{\,n\hat{e}(c)}\left(1-\theta\right)^{n-n\hat{e}(c)} \tag{3.5}$$

with $\theta \in [0, 1]$ and

$$p_{c,\theta}\left(y_i \mid x_i\right) = \begin{cases} \theta & \text{if } c\left(x_i\right) \neq y_i \\ 1-\theta & \text{if } c\left(x_i\right) = y_i, \end{cases} \tag{3.6}$$

if and only if one is willing to assume that the error rate θ is independent of X. If that is the case, then for all fixed $\theta < 0.5$, the classifiers c that achieve the smallest errors on the data correspond to the distributions p_c with the largest likelihood (Grünwald and Langford, 2007). We call this construction of a likelihood out of a classifier the 'fixed-probability likelihood'. Fixed-probability refers to the fact that we take θ to be a constant, it is set to a fixed value independently of the data. θ can be interpreted as the (fixed) error probability that would arise if the Y-values were actually sampled from $p_{c,\theta}$ conditional on the X-values.

The property 'small error $\Leftrightarrow$ large likelihood' is retained if θ is itself fit to the data in a standard Bayesian manner. We will refer to the latter type of likelihoods 'Laplace likelihoods', and explain them in detail in Section 3.4. In this study classification performances based on both types of likelihoods will be shown.

3.2 Procedure

All simulations run in this chapter have been programmed in Java due to computational performance advantages over R. Thanks to its compiler, especially models with a large number of classifiers can be dealt with much more efficiently in terms of memory usage and computation time in Java. In the simulations described in Section 3.3 and 3.4, standard Bayesian inference will be compared to the theoretically more robust forms of Bayes. These include the Safe Bayesian algorithm with the optimal learning rate parameter chosen by minimization of the cumulative randomized log-loss, the Safe Bayesian algorithm with the learning rate parameter chosen by minimization of the cumulative loss of interest (0/1-loss) and a form of Safe Bayes in which the optimal learning rate parameter is chosen that minimizes the 0/1-loss as estimated by leave-one-out cross-validation. The performance of these methods will be measured in terms of average 0/1-loss on a test set that is sampled from the same distribution as the training set.

As has been mentioned before, Safe Bayes predicts by randomizing according to a Cesàro-averaged posterior while standard Bayes mixes according to the posterior based on z^n. The difference between randomizing and predicting was explained at the end of Chapter 1, see Equations (1.8) and (1.9). However, it is also interesting to see how standard Bayes ($\eta = 1$) performs when it randomizes and if Cesàro-averaging is really beneficial in samples with small n. Therefore results given by all four combinations of predicting are given in the tables in this chapter. In the tables presented in the upcoming sections, the Cesàro-averaged posteriors of

standard Bayes, Safe Bayes (which minimizes the cumulative randomized log-loss) and the Safe Bayes variant that minimizes the cumulative 0/1-loss, are given by $\mathrm{CES}\left(\Pi \mid z^n, \hat{\eta}\right) = \frac{1}{n}\sum_{i=1}^{n} \Pi \mid z^i, \hat{\eta}$. Standard Bayes obviously doesn't estimate a learning rate, so $\hat{\eta}$ is fixed at 1 in that case. The Cesàro-averaged posterior of the leave-one-out cross-validated Safe Bayes algorithm with the number of folds k equal to n, also denoted as $\mathrm{CES}\left(\Pi \mid z^n, \hat{\eta}\right)$, is given by $\frac{1}{k}\sum_{i=1}^{k} \Pi \mid z^i, \hat{\eta}$, with z^i denoting the training set of the i-th fold.

Because the behavior of Bayes differs between classifier models that have been specified with a fixed-probability and Laplace likelihood, slightly different simulations are needed to show the suboptimality in both. Therefore both likelihoods are given separate sections. The next section covers simulations in which a likelihood with fixed θ will be used, whereas in Section 3.4 it is shown how Bayes behaves on simulated examples in which a likelihood is specified with θ depending on Laplace smoothed $\hat{e}(c)$.

3.3 Fixed-probability likelihood

As shown in the previous section, Bayesian inference can be applied to a set of classifiers by transforming them to probability distributions and specifying a likelihood with fixed θ. Although this is a relatively simple approach, it does have a disadvantage. Namely, θ can be chosen in arbitrarily many ways. Asymptotically, as long as $\mathcal{C}$ is finite, this isn't much of a problem, as for some z^n, $\mathcal{C}$ and any fixed $\theta < 0.5$, Bayes will in the limit lead to the same posterior. However, different θ will produce different posteriors when n is small. On the other hand, a pleasant property of specifying a fixed θ (instead of a θ that is itself estimated by data) is that for each classifier c the log-loss, or minus log-likelihood, of $p_{c,\theta}$ on z^n is a linear function of the 0/1-loss of c (Grünwald and Langford, 2007).

The Safe Bayesian algorithm chooses an η that minimizes the cumulative randomized log-loss. In the fixed-probability likelihood setting the log-loss of $c \in \mathcal{C}$ at observation i is given by

$$\text{log-loss}_{c,i} = \begin{cases} -\log\left(\theta\right) & \text{if } c\left(x_i\right) \neq y_i \\ -\log\left(1-\theta\right) & \text{if } c\left(x_i\right) = y_i. \end{cases} \tag{3.7}$$

Now the cumulative randomized log-loss on z^n over all $\mathcal{C}$ for a fixed η can be expressed as

$$\text{Cu-R-L-L}_{z^n,\eta} = \sum_{i=1}^{n} \sum_{j=1}^{q} (\pi \left(c_j \mid z^i, \eta \right) \cdot \text{ log-loss}_{c_j,i}), \tag{3.8}$$

which will be used in the subsequent sections to find the optimal η for the Safe Bayesian approach. Note that this is just (2.14) again.

3.3.1 Simulation 1: Beneficial scenario

Let $y^n = (y_1, \ldots, y_n)$, with each $y_i \in \{0, 1\}$, be a random sample drawn i.i.d. from a Bernoulli($\frac{1}{2}$) distribution. Let the set of classifiers $\mathcal{C}$ consist of $\{c_1, c_2, \ldots, c_q\}$. Now for all $c_j \in \mathcal{C}$ a vector $x = (x_{1j}, \ldots, x_{nj})$ is sampled according to $P^*_{\gamma,\lambda}$ with $\gamma \in [\,0, 1\,]$ and $\lambda = (\lambda_1, ..., \lambda_q) \in [\,0, 1\,]^q$ defined as follows. The first parameter, γ, indicates the difficulty of the to be predicted y by the classifiers by specifying whether the observation is relatively easy or hard to predict. The latter parameter, λ_c, denotes the probability of classifier c correctly predicting y on observations that are indicated as hard. More specifically, for all n a random sample is taken from random variable W that is Bernoulli distributed with success probability $p = \gamma$ so that there is a vector $w = (w_1, \ldots, w_n)$, with each $w_i \in \{0, 1\}$ denoting the difficulty of the to be predicted observations y^n. Then for w_i equal to 1, an $x_{ij} \in X$ is sampled such that x_{ij} is set to y_i with probability λ_j and set to $|1 - y_i|$ otherwise. Whereas if $w_i = 0$, then x_{ij} is set such that it is equal to y_i instantly. The misclassification rate of each classifier on the sample is now given by $\hat{e}(c_j) = \frac{1}{n} \sum_{i=1}^{n} I_{y_i \neq x_{ij}}$. Based on the $\hat{e}(c_j)$ for all $c \in \mathcal{C}$ the corresponding conditional probability distributions $p_{c,\theta}$, with a fixed $\theta < 0.5$, are calculated in the way as described in the previous section.

The classification models that can be composed in the setting just described can be considered misspecified in two ways. Let p_{c_m} be the probability distribution corresponding to the classifier c_m as defined under (3.3). Now each y_i is equal to either $p_{c_m}(x_{im})$ (if $w_i = 0$, an 'easy' example) or some random noise variable V that is distributed independently of X (if $w_i = 1$, a hard example). In that light, the first way in which the model can be misspecified is when the Bayes' optimal classifier $c_b \notin \mathcal{C}$ such that $p_{c_m} \neq p_{c_b}$. The second way occurs when V is not independent of X such that under the true distribution, the noise is heteroskedastic. This phenomenon can be enforced by setting $\gamma < 1$: the probability distributions $p_{c,\theta}$ are still homoskedastic whereas the true noise becomes heteroskedastic (on easy

examples, the 'true' noise is 0).

In the current simulation $\lambda_1 = 0.5$ and $\lambda_2 = \lambda_3 = \ldots = \lambda_q = 0.4$. A prior distribution $\pi(\mathcal{C})$ has been specified such that $\pi(c_1) = 0.5$ and $\pi(c_2) = \pi(c_3) = \ldots = \pi(c_q) = \frac{1}{2(q-1)}$. Because of difference in classification performance, classifier c_1 will be referred to as the *good classifier* and classifiers $c_2 = c_3 = \ldots = c_q$ will be referred to as being *bad classifiers*. The number of observations n is set to 30. The error rate θ has been fixed at 0.20. There is no particular reason for this choice, as all $\theta < 0.5$ would have been fine. The observation difficulty parameter γ is set to 0.70. As a test set 100 observations have been sampled from the same distribution. The calculation of the average 0/1-loss on one test set based on one training set will be referred to as one run. In total, to account for the variance between runs, 1.000 runs will be carried out. Because of the settings just specified, the expected number of 'hard' and 'easy' observations is equal to 70 and 30 respectively. As a result, the expected misclassification error of the good classifier on the test set will be $100 - (70 \cdot 0.5 + 30 \cdot 1) = 35$, which comes down to an e_{P^*} of 0.35. The bad classifiers, on the other hand, are expected to have a misclassification error of $100-(70\cdot 0.4+30\cdot 1) = 42$, which is equal to an e_{P^*} of 0.42. Because of this difference, Bayes will give substantial weight to the good classifier in the limit of infinite sample size. However, given $n = 30$ and q very large, it is very likely that due to variance, some bad classifiers will perform better than what is expected, resulting in relatively high weights on suboptimal classifiers. Therefore, it is expected that standard Bayes will perform relatively poorly compared to the robust Bayesian methods.

Figure 3 shows the average 0/1-loss (or misclassification rate) on the test set over the runs as a function of q for a variety of methods. Recall from Chapter 1, Section 1.3, Equations (1.8) and (1.9) that we can predict test set data in two ways: 'mixed' and 'randomized' (which is the prescription for how to use Safe Bayes given by Grünwald (2012). For classification problems, mixed prediction amounts to the following: we take the Bayes act according to either the generalized posterior or the Cesàro-average of the generalized posterior (the latter was originally prescribed by Grünwald (2012)). This Bayes act will simply implement a majority vote: if the probability that $Y = 1$ given X according to the generalized posterior (or Cesàro-averaged generalized posterior) is larger than $\frac{1}{2}$ we predict 1, otherwise 0.

In Figure 3 we look at (a) the randomized predicting Safe Bayesian algorithm based on the Cesàro-averaged posteriors with the learning rate parameter chosen by minimization of the cumulative randomized log-loss (SB-R-Ces-Cu-R-L-L). This is what was originally advocated by Grünwald (2012). Note that it is identical to SB-R-

Ces-Cu-R-0/1-L, (i.e. we determine η based on randomized 0/1-loss rather than log-loss) for the reasons given at the end of Section 2.2. We next look at (b) the Safe Bayesian algorithm; but now both in the training and in the prediction mixing instead of randomization is used: we pick the η that minimizes the cumulative 0/1-loss, where at each point in time we predict by using the posterior in the standard Bayesian way. We denote this as SB-M-Cu-0/1-L. Next, (c), we choose the learning rate parameter based on cross-validation, where predictions in the various folds of the training set are made based on mixing. We denote this as SB-M-Ces-CV-0/1-L. Finally, (d), we test standard Bayes ($\eta = 1$).

As can be seen in the figure, when the number of bad classifiers increases, standard Bayes performs clearly worse than the Safe Bayesian variants. Despite the large prior on the good classifier and the difference between its e_{P^*} and that of the bad classifiers, standard Bayes still puts a substantial weight on a relatively large number of bad classifiers when q is large. This is, however, not strange as there is a high probability that part of the bad classifiers will classify better than the good classifier just due to chance. The unsatisfying behavior of Bayes now stems from situations when the posterior is rather diffuse and puts more weight on a set of bad classifiers combined than on the good classifier. Because it mixes these distributions, this means that on hard observations Bayes doesn't predict as bad as the bad classifiers but even worse. In the most extreme case when Bayes' posterior puts a negligible weight on the good classifier and spreads its full mass on b bad classifiers, then the expected error on hard observations will be equal to $1 - 0.4^b$, since, by construction, all bad classifiers err independently. The SB-R-Ces-Cu-R-L-L doesn't suffer from this in the current situation for two reasons.

1. First of all, the prior puts a lot of weight on the good classifier. Therefore the generalized Bayes posterior can be specified with a learning rate parameter that helps to protect against this phenomenon.

2. Second, because it randomizes, the average 0/1-loss on the test set will never be larger than the e_{P^*} of the worst classifier.

Whether both 1. and 2. are needed in order to get optimal results depends on the situation. Looking at the performances of the other robust Bayesian methods in Figure 3, in the current setting it is sufficient to just apply the generalized Bayesian posterior. However, in Section 3.3.3 we will demonstrate that sometimes randomizing is really necessary.

It can be seen that the SB-M-Cu-0/1-L and especially the SB-M-Ces-CV-0/1-L

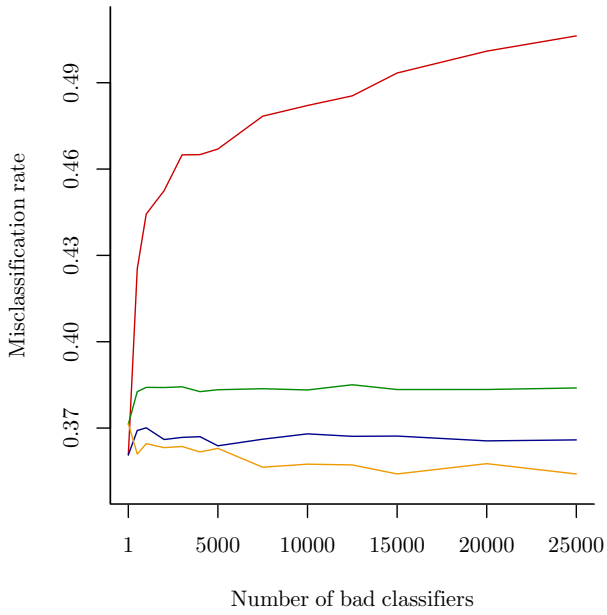

Figure 3: Misclassification rates of SB-R-Ces-Cu-R-L-L (green), SB-M-Cu-0/1-L (blue), SB-M-Ces-CV-0/1-L (orange) and standard Bayes (red) as a function of q.

perform even better than the original form of Safe Bayes. This is mainly because instead of minimizing the cumulative randomized log-loss in order to pick a learning rate, the cumulative mix loss of interest (0/1-loss) is minimized and predictions on the test set are made in a more standard Bayesian manner by mixing according to the posterior. Because the loss of interest is used instead of a proxy loss like the log-loss, the learning rate can be chosen somewhat more accurately. Note, however, that by not predicting randomized one is less robust in situations where the use of the generalized Bayes posterior is not sufficient (e.g. the current simulation setting but with a uniform prior).

Table 1 shows the misclassification rates of all combinations of learning rate selection and prediction methods in the situation where $q = 25.000$. The most striking observation is that Bayes performs even worse than random guessing. It can also be seen that if Bayes would predict by randomization, the problem would already be much smaller. The use of the Cesàro-averaged posteriors seems to be beneficial when mixing. However, the fact that standard Bayes also performs better when predicting with a Cesàro-averaged posterior indicates that this could be due to chance, as there is no clear reason why the latter should be the case. The contrast in average chosen η is eye-catching. Especially the Cu-R-L-L deviates from the other

Table 1: Misclassification rates based on the setting in simulation 3.3.1. Shown are the results of the various learning rate selecting methods while predicting randomized or mixed based on $\Pi \mid z^n, \hat{\eta}$ and $\text{CES}(\Pi \mid z^n, \hat{\eta})$.

	$\Pi \mid z^n, \hat{\eta}$		$\text{CES}(\Pi \mid z^n, \hat{\eta})$		
	Mixed	Randomized	Mixed	Randomized	$\hat{\eta}$
Standard Bayes	0.51	0.40	0.45	0.40	1
Cu-R-L-L	0.39	0.38	0.38	0.38	0.36
Cu-M-0/1-L	0.37	0.38	0.35	0.38	0.58
CV-M-0/1-L	0.35	0.38	0.35	0.38	0.62

two robust methods. The Cu-M-0/1-L and the CV-M-0/1-L yield relatively similar $\hat{\eta}$. This is not surprising as both methods try to minimize the same loss function.

3.3.2 Simulation 2: Unfavorable scenario

In the previous simulation a situation was created in which the best classifier was also given the highest prior weight. It is not that surprising to see that the application of a generalized Bayes posterior, which only has the possibility to increase and not decrease the influence of the prior with respect to the data, leads to better results than standard Bayes. All in all, even if η is chosen randomly (but ≤ 1), it would probably make the Safe Bayesian outperform Bayes. To show that the good performance of Safe Bayes was not a coincidence, in the current simulation everything is the same as in simulation 1 except the classifier c_1 now predicts with $\lambda = 0.4$ and the $c_2, \ldots, c_q$ with $\lambda = 0.5$. Because $\pi(\mathcal{C})$ remains the same, a relative large weight is now on the *worst* classifier. This means that picking overly small η's will result in misclassification rates close to the e_{P^*} of the worst classifier.

As can be seen from Figure 4 this is not the case. Although Bayes performs better than the Safe Bayes variants, the difference is small. The fact that the results of the robust forms of Bayesian inference, in a situation that is least favorable to them, are so close to standard Bayes is encouraging. Table 2 seems to confirm the idea that the Cesàro-averaging of the posteriors is not necessarily beneficial. While Grünwald (2012) shows that, for large samples, predicting with a Cesàro-averaged posterior theoretically corresponds to the $\hat{\eta}$ based on a cumulative loss, this property can only be relied on when n is large enough.

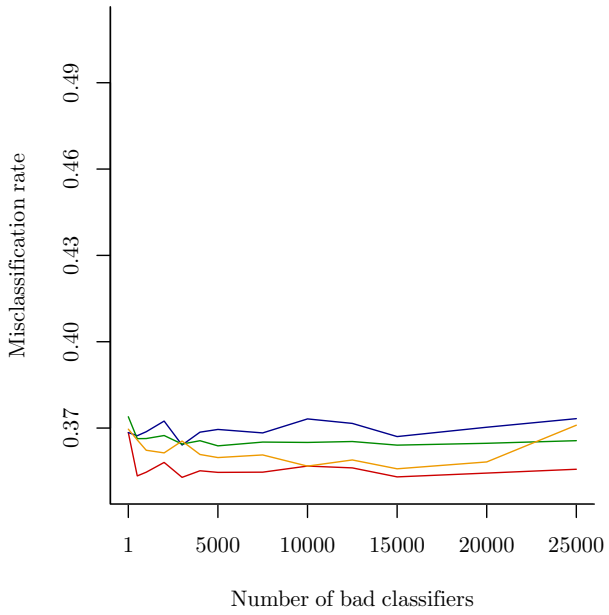

Figure 4: Misclassification rates of SB-R-Ces-Cu-R-L-L (green), SB-M-Cu-0/1-L (blue), SB-M-Ces-CV-0/1-L (orange) and standard Bayes (red) as a function of q.

Table 2: Misclassification rates based on the setting in simulation 3.3.2. Shown are the results of the various learning rate selecting methods while predicting randomized or mixed based on $\Pi \mid z^n, \hat{\eta}$ and $\text{CES}(\Pi \mid z^n, \hat{\eta})$.

	$\Pi \mid z^n, \hat{\eta}$		$\text{CES}(\Pi \mid z^n, \hat{\eta})$		
	Mixed	Randomized	Mixed	Randomized	$\hat{\eta}$
Standard Bayes	0.36	0.35	0.42	0.36	1
Cu-R-LL	0.37	0.36	0.42	0.37	0.76
Cu-M-0/1-L	0.37	0.35	0.42	0.36	0.87
CV-M-0/1-L	0.37	0.36	0.37	0.36	0.81

3.3.3 Simulation 3: Randomizing vs. Mixing

In the previous simulations it can be seen that both mixing and randomizing can lead to satisfying results. There are however also situations in which mixing according to the generalized posterior just isn't sufficient and randomizing is the only reasonable way to predict. In the current simulation such a setting is described. Because it deviates from the last two simulations a formal introduction will be given first.

Let $z^n = y^n = (y_1, ...y_n)$, with each $y_i \in \{0, 1\}$, be a sample drawn from a distribu-

tion P^*_γ such that

$$y_{i+1} = \begin{cases} |y_i - 1| & \text{with probability } \gamma \\ y_i & \text{with probability } 1-\gamma, \end{cases} \tag{3.9}$$

with γ very close to 1. Thus, in contrast to the other experiments reported in this thesis, there are no x-values here and data are not i.i.d. Let the set of classifiers $\mathcal{C}$ consist of $\{c_0, c_1\}$ with $\pi(\mathcal{C})$ uniform. Both classifiers constantly predict y^n the same way, with c_0 always predicting 0 and c_1 always predicting 1. The misclassification rate of the classifiers on the sample is given by $\hat{e}(c) = \frac{1}{n}\sum_{i=1}^{n} I_{y_i \neq c}$. The conditional probability distributions $p_{c,\theta}$, with $\theta = 0.2$, are calculated in the same way as described in the previous sections. The training set consists of $n = 100$ observations and the performance of the various methods are given in terms of misclassification rate on the test set which is taken to be y_{n+1}. In case the posterior of c_0 and c_1 at observation i are equal, the loss that is made when predicting y_{i+1} while mixing is given by flipping a fair coin. The whole procedure has been carried out 10.000 times while each time randomly selecting y_1.

The results are shown in Table 3. Because this setting is specified in order to emphasize the difference between mixing and randomizing, only the misclassification rates are given based on $\Pi \mid z^n, \hat{\eta}$. Cesàro averaging in this situation leads to misclassification rates of 0.50 independent of the learning rate selection method and way of prediction. As this is not a beneficial property belonging to Cesàro averaging but rather a side effect of the current simulation setting, it would merely be distractive to show its results.

Table 3: Misclassification rates based on the setting in simulation 3.3.3. Shown are the results of the various learning rate selecting methods while predicting randomized or mixed based on $\Pi \mid z^n, \hat{\eta}$.

	$\Pi \mid z^n, \hat{\eta}$		
	Mixed	Randomized	$\hat{\eta}$
Standard Bayes	0.75	0.65	1
Cu-R-LL	0.75	0.50	0.008
Cu-M-0/1-L	0.75	0.65	1
CV-M-0/1-L	0.75	0.65	1

As can be seen in Table 3, the only method that stays on par with the misclassification rates of the separate classifiers is the Safe Bayesian algorithm that predicts randomized. It does so by selecting an η such that the posterior almost totally neglects the data in favor of the prior. On the other hand, predicting by mixing with

an η that small doesn't help because even if the posterior is pulled heavily towards the prior, half of the time it would still slightly favor the wrong classifier over the right one, leading to a misclassification regardless.

3.4 Laplace likelihood

In the previous section likelihoods with a fixed θ were specified and it was shown that standard Bayes can go wrong in several situations. Although these classifier models with fixed-probability likelihoods have the attractive property of the log likelihood of $p_{c,\theta}$ on the data being a linear function of the cumulative 0/1-loss, it remains unsatisfactory that θ is chosen arbitrarily. It may be a more natural to let θ depend on the data. Here we will adopt a standard Bayesian way of doing this, which is to equip θ with a uniform prior, independently of the prior on classifiers. Let $r_{c,i}$ be $\sum_{k=1}^{i} I_{c(x_k)\neq y_k}$, that is, the number of misclassifications of classifier c up to observation i. Now the Laplace smoothed number of misclassifications at i can be denoted as $l_{c,i}$ which is equal to $\frac{r_{c,i}+1}{i+2}$ (Grünwald and Langford, 2007). The conditional probability of c given z^n can then be expressed as

$$p_c\left(y^n \mid x^n\right) = \prod_{i=1}^{n} p_c\left(y_i \mid x_i, z^{i-1}\right), \tag{3.10}$$

with the conditional probability of c at observation i given by

$$p_c\left(y_i \mid x_i, z^{i-1}\right) = \begin{cases} l_{c,i} & \text{if } c\left(x_i\right) = y_i \\ 1 - l_{c,i} & \text{if } c\left(x_i\right) \neq y_i. \end{cases} \tag{3.11}$$

A standard calculation (first performed by Laplace) shows that with this definition, $p_c\left(y^n \mid x^n\right)$ is equal to the Bayesian marginal likelihood with a uniform prior on the parameter θ (see, e.g., Grünwald 2007, Chapter 7, for a proof):

$$p_c\left(y^n \mid x^n\right) = \int_{0\leq\theta\leq 1} p_{c,\theta}\left(y^n \mid x^n\right) d\theta, \tag{3.12}$$

which shows that (as long as one uses a constant learning rate 1) basing inferences on this 'Laplace likelihood' is equivalent to doing Bayesian inference under the assumption of homoskedasticity (the misclassification rate still does not depend on X), and with a uniform prior on the classifiers' misclassification error rate. As a consequence, classifiers with $e_{P^*} = \frac{1}{2} + \alpha$ with $\alpha \in [\,0, \frac{1}{2}\,]$, will be given in the limit of $n \to \infty$, a higher likelihood than classifiers with $e_{P^*} = \frac{1}{2} - \beta$ with $\beta \in [\,0, \frac{1}{2}\,]$ for

$\alpha > \beta$. At first sight this may seem somewhat counter-intuitive. One should realize, however, that classifiers that structurally perform worse than random are actually doing better if their predictions would be flipped.

With a Laplace likelihood, the log-loss used in the Safe Bayesian approach in order to find the optimal η that is made by c at observation i can now be expressed as

$$\text{log-loss}_{c,i} = \begin{cases} -\log(l_{c,i}) & \text{if } c(x_i) = y_i \\ -\log(1 - l_{c,i}) & \text{if } c(x_i) \neq y_i, \end{cases} \tag{3.13}$$

whereas the cumulative randomized log-loss on z^n over all $\mathcal{C}$ for a fixed η in the Laplace setting is calculated in the same way as was done in the fixed-probability likelihood case, namely

$$\text{Cu-R-L-L}_{z^n,\eta} = \sum_{i=1}^{n} \sum_{j=1}^{q} (\pi(c_j \mid z^i, \eta) \cdot \text{log-loss}_{c_j,i}). \tag{3.14}$$

3.4.1 Simulation 1: Beneficial scenario

Let $y^n = (y_1, \ldots, y_n)$, with each $y_i \in \{0, 1\}$, be a random sample drawn i.i.d. from a Bernoulli($\frac{1}{2}$) distribution. Let the set of classifiers $\mathcal{C}$ consist of $\{c_1, c_2, \ldots, c_q\}$, and for all $c_j \in \mathcal{C}$ a vector $x^n = (x_{1j}, \ldots, x_{nj})$ is sampled according to $P^*_{\gamma,\lambda}$ just as described in Section 3.3.1 and 3.3.2. Similar to the simulation done in the former section, γ is set to 0.70 and $\pi(\mathcal{C})$ is such that $\pi(c_1) = 0.5$ and $\pi(c_2) = \pi(c_3) = \ldots = \pi(c_q) = \frac{1}{2(q-1)}$. However, the number of observations n is now set to 100 and $\lambda_1 = 0.5$ and $\lambda_2 = \lambda_3 = \ldots = \lambda_q = 0.45$, resulting in an e_{P^*} of 0.350 and 0.385 for the good and bad classifiers respectively. As a test set 100 observations have been sampled from the same distribution and yet again 1.000 runs will be carried out to account for the variance between runs.

Figure 5 shows the average 0/1-loss on the test set over the runs as a function of q for the SB-R-Ces-Cu-R-L-L, SB-M-Cu-0/1-L and standard Bayes. Although the effect is less extreme than in the similar setting with the fixed-probability likelihood, standard Bayes still performs much worse than the worst classifier when q is large enough. The SB-R-Ces-Cu-R-L-L, SB-M-Ces-CV-0/1-L and SB-M-Cu-0/1-L on the other hand will predict no worse than the worst classifier independently from q.

Table 4 shows the average 0/1-loss on the test set of all combinations of learning rate selection and prediction methods in the situation where $q = 100.000$. It can be seen that the SB-R-Cu-RLL and SB-M-Cu-0/1-L don't perform much worse than the best

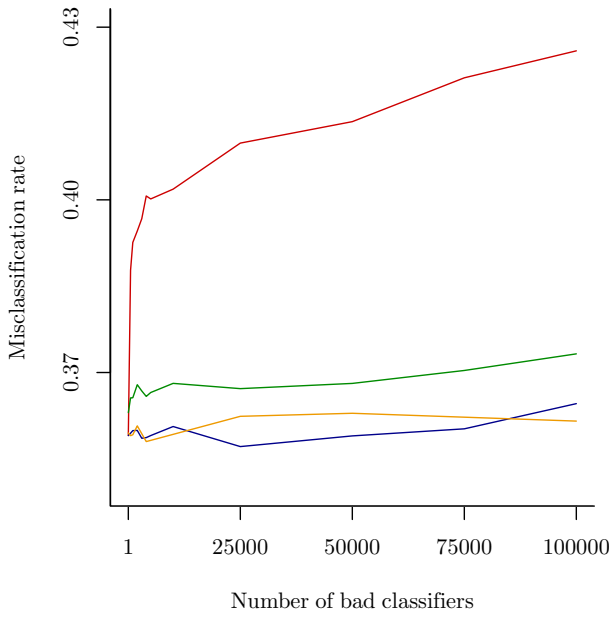

Figure 5: Misclassification rates of SB-R-Ces-Cu-R-L-L (green), SB-M-Cu-0/1-L (blue), SB-M-Ces-CV-0/1-L (orange) and standard Bayes (red) as a function of q.

classifier even if $q = 100.000$. The difference in average chosen η between the former and the latter is interesting, more so because both averages lead to similar results. One reason could be that all values beneath some particular η would be optimal for the SB-M-Cu-0/1-L, but because the cumulative 0/1-loss is the same for all those values, the largest η is chosen. Due to the fact that the log-loss is a continuous loss function, it could very well be the case that much smaller η's would lead to almost neglectably lower log-loss and therefore almost neglectably lower average 0/1-loss on the test set.

Table 4: Misclassification rates based on the setting in simulation 3.4.1. Shown are the results of the various learning rate selecting methods while predicting randomized or mixed based on $\Pi \mid z^n, \hat{\eta}$ and $\text{CES}(\Pi \mid z^n, \hat{\eta})$.

	$\Pi \mid z^n, \hat{\eta}$		$\text{CES}(\Pi \mid z^n, \hat{\eta})$		
	Mixed	Randomized	Mixed	Randomized	$\hat{\eta}$
Standard Bayes	0.43	0.39	0.43	0.38	1
Cu-R-LL	0.36	0.37	0.36	0.37	0.08
Cu-M-0/1-L	0.36	0.37	0.37	0.38	0.66
CV-M-0/1-L	0.36	0.37	0.36	0.37	0.68

3.4.2 Simulation 2: Unfavorable scenario

Just like in the simulation described in Section 3.3.2, a situation is created in which the worst classifier is given the highest prior weight. That is, the classifier c_1 with $\pi(c_1) = 0.5$ now predicts with $\lambda_1 = 0.45$ and the classifiers $c_2, \ldots, c_q$ with $\pi(c_2) = \ldots = \pi(c_q) = \frac{1}{2(q-1)}$ predict with $\lambda_2 = \ldots = \lambda_q = 0.5$ on all hard observations. All the other settings are equal to the simulation described in the previous section.

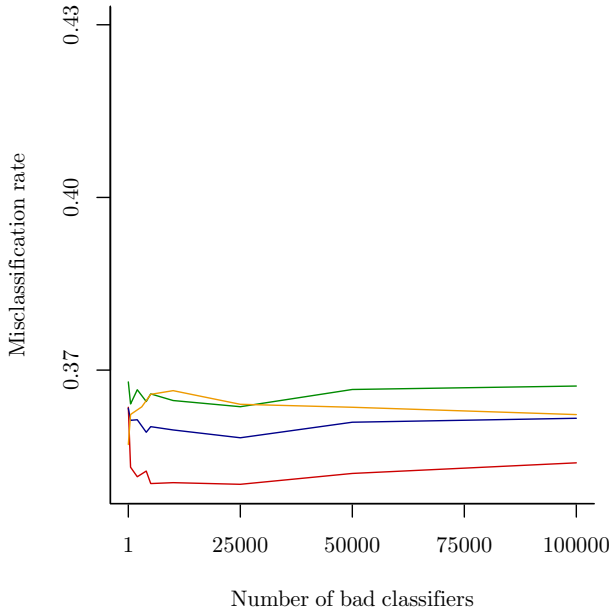

Figure 6: Misclassification rates of SB-R-Ces-Cu-R-L-L (green), SB-M-Cu-0/1-L (blue), SB-M-Ces-CV-0/1-L (orange) and standard Bayes (red) as a function of q.

Figure 6 shows great similarity with Figure 4 from Section 3.3.2. Standard Bayes performs slightly better than the Safe Bayes variants. The most remarkable finding shown in Table 5 is the average chosen η given by minimizing the cumulative randomized log-loss. A small average η is chosen which seems clearly inferior to η's closer to 1. The reason the cumulative randomized log-loss is smaller for η's that are that low is due to the nature of the Laplace likelihood. Especially at small i, the variance of the smoothed Laplace misclassification ratio of classifier c, $\mathrm{E}_{z^i}[(l_{c,i} - \mathrm{E}_{z^i}[\,l_{c,i}\,])^2]$, can be relatively large. Therefore the posterior of a classifier c, which has either a very large or very small $l_{c,i}$, will be given a lot of weight resulting in a relatively considerable log-loss when $c\,(x_{i+1}) \neq y_{i+1}$ or $c\,(x_{i+1}) = y_{i+1}$ respectively. To account for this, an η is chosen such that $\Pi_c \mid z^i, \eta$ remains close to $\pi(c)$, thereby reducing the log-loss that is made. Because the difference in e_{P^*}

between c_1 and $c_2, \ldots, c_q$ is relatively small (0.035), giving more weight to c_1 by choosing an small η overcomes the extra amount of log-loss that is made by picking a large η that gives more weight to the theoretically superior $c_2, \ldots, c_q$ but also induces more log-loss due to this variance.

Table 5: Misclassification rates based on the setting in simulation 3.4.2. Shown are the results of the various learning rate selecting methods while predicting randomized or mixed based on $\Pi \mid z^n, \hat{\eta}$ and $\text{CES}(\Pi \mid z^n, \hat{\eta})$.

	$\Pi \mid z^n, \hat{\eta}$		$\text{CES}(\Pi \mid z^n, \hat{\eta})$		
	Mixed	Randomized	Mixed	Randomized	$\hat{\eta}$
Standard Bayes	0.35	0.35	0.37	0.36	1
Cu-R-LL	0.38	0.37	0.38	0.37	0.12
Cu-M-0/1-L	0.36	0.35	0.37	0.36	0.84
CV-M-0/1-L	0.36	0.35	0.36	0.35	0.77

3.4.3 Simulation 3: Randomizing vs. Mixing

Just as for the fixed-probability likelihoods, for the Laplace likelihood it is sometimes necessary to predict 'randomized' instead of 'mixed' in order to get good classification performance, even when using a generalized posterior. To illustrate this, a slightly different simulation setting has to be set up than was done in Section 3.3.3.

Let $z^n = y^n = (y_1, ...y_n)$, with each $y_i \in \{0, 1\}$, be a sample drawn from a distribution $P^*_{\gamma,\lambda}$ such that

$$y_{i+1} = \begin{cases} |y_i - 1| & \text{with probability } (\gamma) \\ y_i & \text{with probability } (1 - \gamma), \end{cases} \tag{3.15}$$

and γ very close to 1. Let the set of classifiers $\mathcal{C}$ consist of $\{c_0, c_1\}$ with $\pi(\mathcal{C})$ uniform, and c_0 constantly predicting 0 and c_1 constantly predicting 1. Yet, before predicting y_i a biased coin B is flipped that gives the classifier that would have made a mistake a chance to predict y_i correct with probability λ. If B was tossed successfully then $y_{i+1} = y_i$ otherwise $y_{i+1} = |y_i - 1|$ with probability (γ) as usual. In the current simulation λ has been set at 0.05. All other variables are exactly the same as in Section 3.3.3. The inclusion of parameter λ gives the possibility of both classifiers correctly predicting y_i, thereby avoiding the Laplace likelihood property of $l_{c,i} = 1 - l_{c,i}$, in this setting leading to $l_{0,i} = l_{1,i}$ for all i.

As can be seen in Table 6, only randomized predicting leads to reasonable results

Table 6: Misclassification rates based on the setting in simulation 3.4.3. Shown are the results of the various learning rate selecting methods while predicting randomized or mixed based on $\Pi \mid z^n, \hat{\eta}$.

	$\Pi \mid z^n, \hat{\eta}$		
	Mixed	Randomized	$\hat{\eta}$
Standard Bayes	0.71	0.49	1
Cu-R-LL	0.71	0.47	0.01
Cu-M-0/1-L	0.71	0.49	1
CV-M-0/1-L	0.71	0.49	1

due to the exact same reasons as discussed in Section 3.3.3.

3.4.4 Log-loss, 0/1-loss and the Laplace likelihood

As can be seen in simulation 3.4.2, when n is relatively small, the behavior of the log-loss function in combination with a Laplace likelihood can be problematic. A lot of observations are needed to let $l_{c,n}$ converge to $e_{P^*}(c)$. Because $\frac{1}{n}\sum_{i=1}^{n} \text{log-loss}_{c,i}$ correlates with $\mathrm{E}_{z^n}\left(l_{c,n} - \mathrm{E}_{z^n}[\, l_{c,n}\,]\right)^2$ and the likelihood $p\left(z^n \mid c\right)$ obviously depends on $l_{c,n}$, one can imagine that for all $c \in \mathcal{C}$, if $l_{c,n}$ has not yet converged to $e_{P^*}(c)$, a small η is picked, for the same reasons as given in Section 3.4.2. This is illustrated in Figure 7. The setting is similar to the one discussed in Section 3.4.2 but with the difference that the set of classifiers $\mathcal{C}$ consist of $\{c_1, c_2\}$ with $\pi(\mathcal{C})$ such that $\pi(c_1) = 0.75$, $\pi(c_2) = 0.25$, $\lambda_1 = 0.45$, $\lambda_2 = 0.50$, resulting in $e_{P^*}(c_1) = 0.385$ and $e_{P^*}(c_2) = 0.350$.

The figure shows that especially when n is small an η is chosen that is smaller than 0.5, even though on average a high η would be optimal when the goal is to minimize the expected 0/1-loss on the test set. However, this is not only the case when a Laplace likelihood is used, also the fixed-probability likelihood with $\theta_c = e_{P^*}(c)$ for the 'good' classifier c. Note that this choice seems to make the problem as 'well-specified as possible', because the error rate of the good classifier is now correctly specified; the only misspecification occurs because the true error is hetero- rather than homoskedastic. So, the choice of an overly small η seems counter-intuitive here. Yet, it is important to keep in mind that the difference in e_{P^*} between the two classifiers is so small that 'regression to the mean' possibly accounts for a share of the log-loss. That is, the posterior will be relatively high for a classifier, as long as its empirical error $\hat{e}(c)$ is smaller than its expected error $e_{P^*}(c)$. This results in a high log-loss at the point in time where this gap closes. A small η protects against

this. To check this hypothesis a simulation has been run wherein both classifiers are specified such that $\lambda_1 = \lambda_2$, $\pi(c_1) = \pi(c_2)$ and $\theta_{c_1} = \theta_{c_2}$. The same phenomenon occurred which supports this hypothesis.

The reason that minimizing the cumulative randomized log-loss leads to even lower η's when applying a Laplace likelihood is due to the aforementioned variance. The fact that minimization of the randomized log-loss doesn't always correspond to minimization of the randomized 0/1-loss is a bit unsatisfying, especially considering that this effect can be even more extreme when more classifiers are in the model. Theoretically minimization of the cumulative randomized log-loss leads to closest approximation of the true distribution in terms of KL divergence and thus to the smallest expected randomized 0/1-loss on the test set, as demonstrated in Section 2.2. However, it seems that for classification problems it is more beneficial to use the cumulative randomized 0/1-loss instead. In the Laplace likelihood setting, the classifier that is closest to the true distribution in terms of KL divergence (equivalent to the best classifier), will be put the most weight on when n is large enough. As demonstrated, this is not is not necessarily true for small samples. If one minimizes the loss of interest (in this case 0/1-loss) rather than the log-loss, this problem goes away.

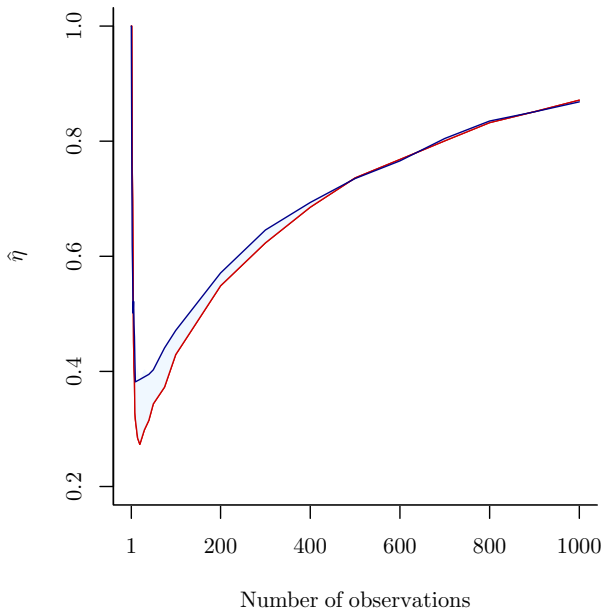

Figure 7: Average $\hat{\eta}$ as a function n given by minimizing the cumulative randomized log-loss based on a Laplace likelihood (red) and fixed-probability likelihood with $\theta_c = e_{P^*}(c)$.

3.5 Conclusion

The purpose of this chapter was to demonstrate that situations exist in which standard Bayesian inference performs badly. This has been done by setting up specific simulations for models with fixed-probability and Laplace likelihood, in which it can be seen that the various Safe Bayesian forms on average outperform standard Bayes. However, it should be noted that these simulations were toy examples and do not reflect the average real world setting. The inclusion of many classifiers that make errors with exactly the same probability is of course somewhat unrealistic. But, because $e_{P^*}(c)$ is not known in advance, it can happen that a large number of very similarly predicting classifiers are included in a model. Although not tested in this study, it is likely that standard Bayes performs badly even if all the classifiers that were assigned equal $e_{P^*}(c)$ were to be given slightly varying $e_{P^*}(c)$ instead.

In Section 3.3.3 and 3.4.3 it was shown that in some situations, to get good performance, it is necessary to predict 'randomized' rather than 'mixed'. There are, however, two things to consider when one wants to predict randomized. First of all, the ultimate goal of classification is usually to assign a class label to a certain subject. When predicting randomized one is not making a 'hard' classification but rather assigning a probability to each different label. Thus, whether or not randomizing is really appropriate also depends on the goal of the researcher. Second, in the simulations it was shown that Bayes can perform worse than the worst classifier in the model, and that even predicting randomized based on a standard Bayesian posterior ($\eta = 1$) already improves the performance. Although this seems a very strong reason to randomize, it is not that simple. Not only situations exist where it is possible for Bayes to predict worse than the worst classifier, but also where it could be possible to predict better than the best classifier. However, in order to profit from this situation, one has to predict by mixing according to the posterior rather than randomizing.

In the simulations performed in this chapter it was demonstrated that Cesàro-averaging of the posteriors did not lead to stable results when a generalized posterior was used with a learning rate parameter chosen by minimization of a cumulative loss. In theory, when selecting an $\hat{\eta} \in \mathcal{S}$, with $\mathcal{S}$ consisting of $\eta_1, \ldots, \eta_k$, such that $\hat{\eta}$ is given by $\arg\min_{\eta \in \mathcal{S}} \sum_{i=1}^{n} \ell_{\Pi|z^{i-1},\eta}(z_i)$, then the posterior corresponding to $\hat{\eta}$ is the Cesàro-averaged posterior. But despite the theoretical requirement to predict according to this posterior because of the relation with $\hat{\eta}$, in practical settings with limited n this can lead to suboptimal results most possibly due to the fact that a lot of weight is on posteriors that have seen too little data from the true distribution

P^*.

Another problem that occurs when n is relatively small, is the behavior of the log-loss function in combination with a Laplace likelihood. Despite its elegance, finding the closest approximation of the true distribution in terms of KL divergence through the use of randomized log-loss, it is likely that minimization of the cumulative or cross-validated loss of interest (i.e. the 0/1-loss) results in better performances.

The figures in the previous sections showed that the pattern of the average 0/1-loss on the test set of SB-M-Ces-CV-0/1-L as a function of q slightly deviates from the other methods. The most likely reason for this is that the SB-M-Ces-CV-0/1-L was added later in the study and was therefore applied on different samples (and less runs due to time constraints), whereas the others where all tested on the same samples. Increasing to number of runs would have fixed this issue. However, the computation time is a problem as 1.000 runs with a model having a lot of classifiers already takes more than half a day to run. Because it is just a minor issue and time is limited, we opted to keep it the way it is.

4 Safe Bayesian Lasso regression

In this chapter the focus is on the application of the Safe Bayesian algorithm for a regularized linear regression model to multiple real-world data sets. As shown in the previous chapters, when the model is wrong the employment of the generalized posterior can be necessary to yield more robust predictions. The Safe Bayesian algorithm will be used to select the optimal penalty parameter for the Bayesian Lasso, making the inference (in theory at least) more robust against misspecification of the model. Whether this is actually the case will be tested on six real-world data sets. As a comparison, also the performances of the Bayesian Lasso, frequentist Lasso, as well as a Bayesian Lasso in which the optimal penalty is determined with k-fold cross-validation will be shown.

In Section 4.1 it is shown that in a regression setting, a likelihood with a learning rate $\frac{1}{\eta}$, similar to the η in the Safe Bayes framework, can be interpreted as the Bayesian equivalent of the frequentist regularized L_1 regression. In Section 4.2 the differences between the Bayesian and frequentist Lasso will be discussed. Section 4.3 covers the issue of the influence of the order of the data on the selection of the learning rate parameter. In Section 4.4 the procedure of applying the Lasso methods to the data sets is explained, whereas in Section 4.5 the results of the different Lasso methods on six data sets are shown. In the final section these results will be discussed in more detail.

4.1 Regularization

Suppose one has data z^n consisting of (x_1, y_1), $\ldots$, (x_n, y_n) distributed according to distribution P^* and one would like to predict a future y_{n+1} from a future x_{n+1}. A common approach is to use a function f that minimizes $\sum_{i=1}^{n}(y_i - f(x_i))^2$ to make these predictions. For example in ordinary least squares regression f consists of beta coefficients $\beta = (\beta_0, \ldots, \beta_p)$ that are given by

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} \big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\big)^2, \tag{4.1}$$

with x_{ij} denoting the value of regressor j at observation i. A problem arises if n is not large enough and x consists of too many variables. In that situation it is very likely that f will explain random noise in z^n, yielding a relatively small bias^2 $(\mathrm{E}_{z^n}[\, f(x^n)\,] - \mathrm{E}_{P^*}[\, Y \mid X\,])^2$. However, at the same time there is a chance of having

a high variance between different data samples $\mathrm{E}_{z^n}[\,(f(x^n) - \mathrm{E}_{z^n}[\,f(x^n)\,])^2\,]$. This effect is called overfitting and generally leads to poor predictions of y_{n+1}. On the other hand, a model with only an intercept will have low variance and high bias. This bias-variance trade-off can be used in order to reduce the prediction error by optimally balancing the errors made by both the bias and the variance. Finding an optimal balance is not straightforward, but acceptable solutions can be found by means of regularization.

A common regularization method for linear regression is the Lasso (Tibshirani, 1996). The Lasso imposes a constraint on the regression coefficients by penalizing the sum of their absolute values. As a result, the regression coefficients are shrunken toward zero with the possibility of setting some coefficients precisely equal to zero. This property of the Lasso is especially convenient when there are many correlated predictors in a linear regression model. Correlated predictors with large coefficients that have opposite signs cancel each other and thereby induce high variance. Penalizing large coefficients as in the Lasso alleviates this problem (Hastie, Tibshirani and Friedman, 2009). So by inducing a bit of bias due to shrinkage of the coefficients, the variance of the predictions is reduced proportionally even more in order to improve the overall prediction accuracy. Another attractive property of the Lasso is that by shrinking some beta coefficients to zero, it also acts as a variable selection method. The Lasso does so by choosing β expressed in the following way

$$\hat{\beta}_{\text{Lasso}} = \arg\min_{\beta} \sum_{i=1}^{n} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\Big)^2 \tag{4.2}$$

$$\text{subject to } \sum_{j=1}^{p} |\beta_j| \leq t \tag{4.3}$$

$$= \arg\min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^{n} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\Big)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}, \tag{4.4}$$

with t corresponding to λ being the complexity parameters that control the amount of shrinkage. The larger the value of λ, the greater the amount of shrinkage (Hastie et all., 2009).

In the frequentist paradigm, the Lasso is viewed as a method to reduce variance of least-squares estimates, and therefore get better predictive performance on future data from the same distribution P^*. But the Lasso also has a Bayesian interpretation, namely the Lasso estimates can be interpreted as the mode of the posterior distribution of β when the regression parameters have independent and identical

Laplace priors (Park and Casella, 2008). Given a linear regression model

$$y^n = f(x^n) + \epsilon \tag{4.5}$$

with $\epsilon \sim N(0, \sigma^2)$, $x^n = (x_1, \ldots, x_n)$ being an n by p dimensional matrix of features, $y^n = (y_1, \ldots, y_n)$ and $f_\beta(x^n) = \sum_{j=1}^{p} \beta_j x_j^n$, then the likelihood is given by

$$\begin{aligned} L\left(\beta; y^n, x^n\right) &= p\left(y^n \mid x^n, \beta\right) && (4.6)\\ &\propto e^{-\sum_{i=1}^{n}(y_i - f_\beta(x_i))^2}. && (4.7) \end{aligned}$$

The Lasso shrinks the linear regression coefficients by adding the penalty term λ such that such that minimizing the penalized least-squares (4.4) is equivalent to maximizing

$$\begin{aligned} L\left(\beta; y^n, x^n, \lambda\right) &= p\left(y^n \mid x^n, \beta, \lambda\right) && (4.8)\\ &\propto e^{-\sum_{i=1}^{n}(y_i - f_\beta(x_i))^2 - \lambda \sum_{j=1}^{p} |\beta_j|} && (4.9)\\ &\propto e^{-\sum_{i=1}^{n}(y_i - f_\beta(x_i))^2} e^{-\lambda \sum_{j=1}^{p} |\beta_j|}. && (4.10) \end{aligned}$$

This L will be referred to as the *pseudo-likelihood* corresponding to Lasso with parameter λ. When looking at (4.10), it can be seen that maximizing L, which is the original frequentist Lasso procedure, looks like something that is proportional to a Bayesian posterior. This exactly happens when $\beta = (\beta_1, \ldots, \beta_p)$ are each given Laplace, or *double exponential*, priors (see Figure 8) with fixed λ, such that

$$\pi\left(\beta\right) \propto e^{-\lambda \sum_{j=1}^{p} |\beta_j|}. \tag{4.11}$$

The posterior distribution of the β-parameters now contains the penalty term just like in the frequentist case, namely

$$\begin{aligned} \pi\left(\beta \mid y^n, x^n, \lambda\right) &\propto L\left(\beta; y^n, x^n, \lambda\right) \pi\left(\beta\right) && (4.12)\\ &\propto e^{-\sum_{i=1}^{n}(y_i - f_\beta(x_i))^2} e^{-\lambda \sum_{j=1}^{p} |\beta_j|}. && (4.13) \end{aligned}$$

When the pseudo-likelihood is rewriten as

$$\begin{aligned} L\left(\beta; y^n, x^n, \lambda\right) &= p\left(y^n \mid x^n, \beta, \lambda\right) && (4.14)\\ &= p\left(y^n \mid x^n, \beta, 1\right)^{\frac{1}{\lambda}} && (4.15)\\ &\propto e^{-\frac{1}{\lambda}\sum_{i=1}^{n}(y_i - f_\beta(x_i))^2}, && (4.16) \end{aligned}$$

and λ is expressed in terms of η such that $\eta = \frac{1}{\lambda}$, it can be seen that it resembles the likelihood in the generalized posterior which is used in the Safe Bayesian algorithm, that is

$$e^{-\frac{1}{\lambda}\sum_{i=1}^{n}(y_i - f_\beta(x_i))^2} = e^{-\eta\sum_{i=1}^{n}(y_i - f_\beta(x_i))^2} \tag{4.17}$$

$$= \left(e^{-\sum_{i=1}^{n}(y_i - f_\beta(x_i))^2}\right)^{\eta}. \tag{4.18}$$

With the likelihood expressed in this way, the posterior distribution of the Bayesian Lasso can now be denoted as

$$\pi\left(\beta \mid y^n, x^n, \tfrac{1}{\eta}\right) \propto L\left(y^n; x^n, \beta, 1\right)^{\eta} \pi(\beta) \tag{4.19}$$

$$\propto \left(e^{-\sum_{i=1}^{n}(y_i - f_\beta(x_i))^2}\right)^{\eta} e^{-\sum_{j=1}^{p}|\beta_j|}. \tag{4.20}$$

When looking at the least-squares estimate of the frequentist Lasso, it can be seen that it corresponds to the maximum a posteriori (MAP) estimate of a Bayesian linear regression model with normally distributed errors and a Laplace prior, as given by (4.13). This model will be referred to as *the Bayesian Lasso*.

Do note that the Laplace prior specified for the Bayesian Lasso applied in the subsequent sections of this study differs from the one described above. The reason behind this stems from the fact that the posteriors are estimated by using Gibbs sampling. Park and Casella (2008) demonstrate that by adding an extra parameter σ^2 to the Laplace prior, such that

$$\pi\left(\beta \mid \sigma^2\right) \propto \frac{\lambda}{2\sqrt{\sigma^2}} e^{-\sum_{j=1}^{p}|\beta_j|/\sqrt{\sigma^2}}, \tag{4.21}$$

the posterior is guaranteed to be unimodal which improves the speed of convergence of Gibbs sampling process. The extra parameter σ^2 is then given a noninformative inverse-gamma prior itself. As a consequence the MAP estimates don't necessarily correspond to the maximum likelihood estimates anymore.

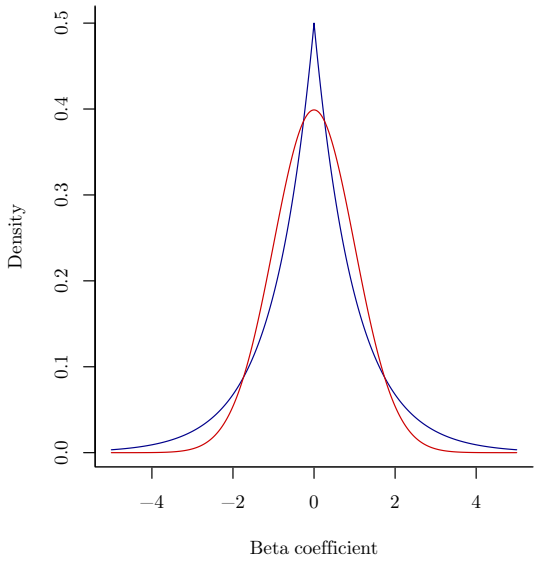

Figure 8: Probability density functions of the Laplace (blue) and Gaussian (red) distribution. The Laplace prior that characterizes the Bayesian Lasso puts more mass on β_j near zero and in the tails than the Gaussian prior used in another regularization method, namely Ridge regression. As a consequence, the Lasso tends to produce estimates that are either close to zero or relatively large (Tibshirani, 1996).

4.2 Bayesian versus frequentist Lasso

As shown in the previous section, the Bayesian Lasso and its frequentist counterpart have a lot in common. In fact, given the same λ, the MAP estimates from the Bayesian Lasso are identical to the penalized least squares estimates from the frequentist Lasso. However, MAP is not the only Bayesian estimation method. Because MAP is equal to the mode of the posterior, it yields relatively good estimates when the posterior is unimodal and symmetric. On the other hand when the posterior is multimodal or skewed it doesn't, which advocates for using another measure for turning the full posterior distribution into an estimate. By taking the mean of the posterior as point estimate more information about the posterior is used which reduces this problem. Kyung, Gill, Ghosh and Casella (2010) compared the performance of the Bayesian Lasso, with the posterior mean as point estimate and a hyperprior on λ, to the frequentist Lasso and concluded that the Bayesian Lasso was sometimes better and never worse.

Using this 'full' Bayesian approach instead of MAP does have consequences for the behavior of the beta coefficients. Park and Casella (2008) concluded that the full Bayesian Lasso estimates appear to be a compromise between the frequentist

Lasso and ridge regression estimates, as the full Bayesian Lasso appears to pull the beta coefficients of the more weakly related predictors to zero faster than ridge regression. However, in contrast to the frequentist Lasso, the full Bayesian Lasso doesn't often set the beta coefficients of these weakly related predictors exactly to zero, but instead makes them relatively small. This does mean that although the full Bayesian Lasso seems to predict better than its frequentist counterpart, it can't be considered a variable selection method anymore.

Compared to the frequentist Lasso, the Bayesian Lasso does have another disadvantage. As was demonstrated previously, if λ is estimated from the data and the model is misspecified, then the $\hat{\lambda}$ may be unreliable. Choosing the λ that minimizes a cumulative loss as in the Safe Bayesian approach gives an asymptotically unbiased estimate independent of the correctness of the model assumptions. It could therefore be possible to get the best of both the frequentist and the Bayesian world by employing a somewhat hybrid form of the Lasso: better prediction accuracy without the loss of robustness. If this can be achieved will be tested in the subsequent sections. First, the four methods that are to be applied to the data sets will be specified in more detail.

Bayesian Lasso. As mentioned in the introduction, the Bayesian Lasso has been implemented as described in Park and Casella (2008). However, the authors describe two ways of estimating λ, namely by marginal maximum likelihood and by specifying a hyperprior for λ. In the current study the latter option is applied. Also, in the relevant paper it has been opted to use the posterior medians of the beta coefficients as point estimates. In the following sections though, there has been chosen to use the mean instead, similar to Kyung et all. (2010). The reason for this is that the squared loss is the loss function that is applied, which is minimized by the posterior mean of y^n.

Safe Bayesian Lasso. Although almost identical to the standard Bayesian Lasso, the Safe Bayesian Lasso deviates on one important aspect. Instead of using a hyperprior to estimate λ, it is kept fixed and all $\lambda_1, \ldots, \lambda_p \in \mathcal{S}$, with $\mathcal{S}$ denoting the set of candidate λ's, are tested simultaneously as described by algorithm 1 in Section 2.3. Because the loss of interest is the squared loss, the Safe Bayesian algorithm applied to the Lasso setting picks the λ that minimizes the cumulative squared loss. Because predictions are made based on the expectation over the full posterior, the Safe Bayesian form that is tested in this chapter predicts in the standard Bayesian way instead of randomizing. There is, however, one complication with the use of the Safe Bayesian algorithm in batch settings, namely the cumulative loss depends

on the order of the data. Because this can be a serious issue, Section 4.3 covers this problem in more detail and gives a possible solution.

Cross-validated Bayesian Lasso. Instead of using a cumulative loss that can be dependent on the order of the data, it is also possible to apply k-fold cross-validation to estimate the optimal λ. This method is often used in frequestist statistics to estimate the expected prediction performance of a model. In this approach the data are randomly split to form k approximately equal sized subsets. Each subset is once used as test set, with all the other remaining subsets being used as the training set. The generalization performance is calculated each fold, after which the mean generalization performance observed over all k folds provides an estimate of the generalization performance of a model trained on the entire data set (Cawley and Talbot, 2010). Alqallaf and Gustafson (2001) studied the performance of k-fold cross-validation for Bayesian model selection and concluded that many folds are to be preferred over few folds. However, increasing the number of folds comes at the price of computation time. Therefore in the current study, 20-fold cross-validation is used as a compromise. Similar to the Safe Bayesian Lasso, λ is kept fixed. For all $\lambda \in \mathcal{S}$ the mean generalization performance in terms of MSE is calculated and the λ is picked that minimizes this statistic. Predictions on the test sets are made based on the Cesàro-averaged posterior $\frac{1}{k}\sum_{i=1}^{k} \pi\left(\beta \mid z^i, \hat{\eta}\right)$, with k the number of folds and z^i denoting the training set of the i-th fold.

Frequentist Lasso. Apart from the Bayesian and Safe Bayesian Lasso there is also the original, frequentist Lasso. In the current study the implementation of the R function *lars* from the same named package is used. A full detailed description of the method can be found in Efron, Hastie, Johnstone and Tibshirani (2004). Instead of specifying a set of candidate λ's, 100 equally spread out shrinkage factors between 0 and 1 are tested. These shrinkage factors are the fractions of the norm of the least-squares beta coefficient vector represented by the norm of the beta coefficient vector given by the Lasso for a certain λ. Large values of λ represent high shrinkage and hence correspond to small shrinkage factor values (Hans, 2009).

4.3 Order of the data

Because the Safe Bayesian algorithm can be applied in a variety of settings such as sequential prediction problems as well as batch scenarios, it makes for a universal approach. As good as this sounds, it does have some weaknesses. Due to the fact that the learning rate parameter is chosen by minimizing a cumulative loss that is

based on the prediction accuracy of sequentially estimated posteriors, the chosen value potentially depends on the order of the data. Relatively extreme values early in the data set will influence the posterior a long time which favors shrinkage of the parameters, and hence picking a small η, i.e. high λ. On the other hand, when these extreme values are all positioned near the end of the data, the cumulative loss will be relatively low for smaller λ.

In sequential prediction settings the order of the data is part of the problem and inevitable. In the batch scenario on the contrary, it seems that data dependence should be avoided. As no such problem exists when dealing with data sets with an infinite amount of observations, the Safe Bayesian approach gives an asymptotically optimal estimate independent of model correctness. However, in practice, this is of course never the case and it is therefore a serious problem to be reckoned with.

A possible solution to this concern is applying the algorithm to multiple randomly taken permutations of the data and pick the λ that minimizes the sum of all the cumulative losses. This would theoretically solve the problem but is computationally intractable. A more viable approach is fixing the order of the data in advance to make sure observations of all regions of the distribution of the dependent variable are equally spread out over the data sequence. Even though this doesn't guarantee a cumulative loss that is an unbiased estimator of the true cumulative loss, it is a relatively simple measure to at least alleviate the issue somewhat. Still, the application of a cumulative loss in order to find the optimal λ in a batch setting keeps feeling somewhat unnatural. The performance of the Safe Bayesian algorithm based on a fixed order of the data will be shown in Section 4.5. The algorithm that is applied to order the data in the specific way can be found in Appendix A.

4.4 Procedure

In order to test the performance of the Safe Bayesian algorithm on real-world data sets it has been implemented in the statistical software environment R, by making use of the *blasso* function inside the *monomvn* package (Gramacy, 2013). Gibbs sampling is used to estimate the posterior distribution, with a total number of 15.000 iterations and a burn-in of 1.000 to ensure convergence. To rule out the possibility that the performance of the various methods tested in this study depend on a specific training and test set, all methods are applied 50 times to randomly selected training and test sets from the same data set. More specifically, the training set consist of 80 percent of the data with the remaining 20 percent being the test set.

The squared error loss in terms of the mean squared error (MSE) is calculated for each random sampled combination of training and test set. Hereafter significance tests are carried out to see whether the methods yield different results.

As mentioned in the introduction of the current chapter, the performance of the Bayesian Lasso will be compared to the performance of the frequentist Lasso and the robust Bayesian Lasso algorithms, namely the Safe Bayesian Lasso and 20-fold cross-validated Bayesian Lasso. These robust Bayesian methods search for the optimal $\lambda \in \mathcal{S}$. In theory one can make $\mathcal{S}$ include all possible choices of λ, but in practice $\mathcal{S}$ has to be limited to a certain size due to computation time issues. Depending on the number of processor cores available it is possible to do a parallel grid search in $\mathcal{S}$ reducing the computation time vastly. In this current research, an eight core CPU was available of which seven cores could be used in parallel for computational purposes. Therefore, first, $\mathcal{S}$ is determined such that it includes reasonable choices of λ (surrounding the expectation of λ given by the standard Bayesian Lasso) with the number of candidate λ's equal to seven. On all six the data sets an $\mathcal{S}$ consisting of $\left\{\lambda_1 = 2^0, \lambda_2 = 2^1, \ldots, \lambda_7 = 2^6\right\}$ seemed fine. Then for all $\lambda \in \mathcal{S}$ the squared loss is calculated simultaneously. To improve the accuracy of the estimation process, a new grid search is carried out in a refined set with candidate λ's near the $\hat{\lambda}$ from the first search. The exact search algorithm used in the robust Bayesian procedures can be found in Appendix B.

Predicting with the full posterior instead of a MAP estimate requires sampling from the posterior by using MCMC estimation techniques. This procedure is known to be rather time-consuming. The Safe Bayesian algorithm is therefore particularly computationally intensive as it has to calculate a posterior distribution n times. Although it is still applicable in practice if n is not too large, to scientifically proof that the method works it has to be applied multiple times on different randomly sampled sets of training and test data. In this study some data sets have more than 400 observations. Depending on the number of parameters that have to be estimated, the computation time for just one training and test combination is often more than three hours. Because running the for this study available computer for more than six days nonstop is not an option, it has been decided to apply the Safe Bayesian algorithm on just the four data sets with relatively small amount of data, namely the Servo, Yacht, Birth weight and Prostate cancer data.

4.5 Data sets & results

The performance of the aforementioned Lasso methods will be tested on six real-world data sets, namely: Boston housing, Servo, Yacht, Diabetes, Birth weight and Prostate cancer data. These data sets have been selected in order to show the functioning of the algorithms on data for which the model implicit in the Bayesian lasso (y^n is a linear function of x^n with normally distributed noise as in Equation (4.13)) is correct, but also for data for which this model is incorrect.

To assess the correctness of the Bayesian Lasso model, various residual plots are shown per data set. These include a *residual versus observed plot*, which shows the residuals as produced by the specified model against the actual observed y^n values. Based on this plot violations of linearity can be identified. The same holds for the *observed versus predicted plot*, which shows a graph of the actual observed y^n values against the predicted Y values based on the specified model. To check whether the model violates the homoskedasticity assumption, a plot of the *residuals versus predicted* is shown as well. Finally, to check the assumption of normality of the residuals a *QQ-plot* is added. This is a plot of the quantiles of the distribution of the residuals versus the quantiles of a normal distribution with the same mean and variance.

All predictors in the models have been standardized to have mean zero and standard deviation one. Some data sets include categorical variables. In order to fit models to these data, the variables in question have been coded into indicator variables. The possibility exists that some of beta coefficients of indicator variables are set to zero while others are not. As a consequence, the Lasso, in a sense, loses its variable selection property on these data sets. However, as the focus is fully on the performance in terms of prediction accuracy, it is still interesting to apply the different Lasso methods in these cases. Performance is represented in terms of average MSE over the 50 runs. Two-sided t-tests are carried out to be able to tell which method performs best. In all these tests the null hypothesis is $\text{MSE}_{m_1} = \text{MSE}_{m_2}$ and the alternative hypothesis is $H_a : \text{MSE}_{m_1} \neq \text{MSE}_{m_2}$, with m_1 and m_2 denoting two arbitrary methods of the previously mentioned methods that will be compared. Also the shrinkage factors $\in [0, 1]$ are shown.

4.5.1 Boston housing data

This well known data set is used in a variety of papers, including the study on the frequentist Lasso (Efron et all., 2004), and concerns housing values in suburbs of

Boston. The data set consists of 506 observations, 13 predictor variables (of which 1 categorical) and a real valued response variable.

When looking at Figure 9, it can be seen that the model as estimated by the Bayesian Lasso violates the normality assumption of the general linear model. Especially the QQ-plot clearly shows the non-normality of the errors. Therefore the Boston housing data can be a data set where the Bayesian performs somewhat suboptimal. Table 7 shows the averaged MSE and shrinkage factor of the Bayesian Lasso, cross-validated Bayesian Lasso and the frequentist Lasso.

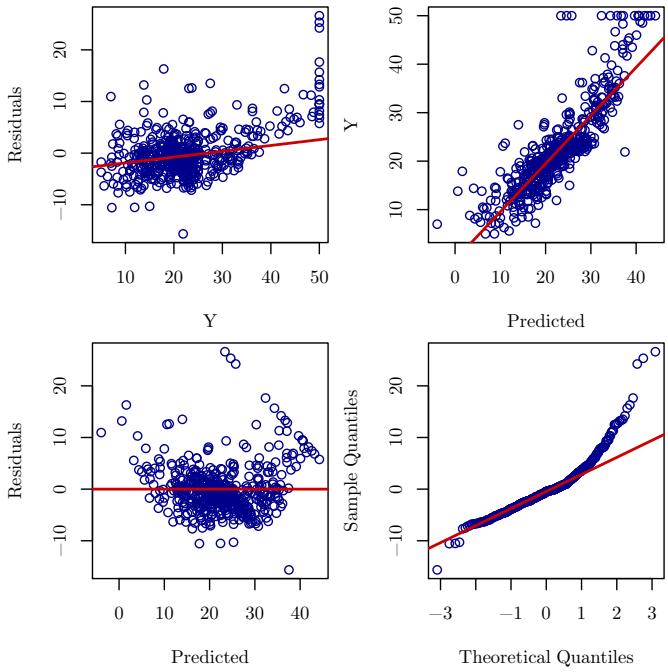

Figure 9: Model assessment plots based on the linear model produced by the Bayesian Lasso on the Boston housing data. Shown are the residuals vs. observed plot (upper left), observed vs. predicted plot (upper right), residuals vs. predicted plot (bottom left) and QQ-plot (bottom right).

Table 7: MSE and shrinkage factor of the Bayesian, cross-validated Bayesian (CVB) and the frequentist Lasso on the Boston housing data.

Boston housing data			
	Bayesian	CVB	Frequentist
MSE	23.3	23.2	23.3
Shrinkage factor	0.95	0.98	0.94

As can be seen in Table 7, the robust cross-validated Bayesian Lasso performs better than the Bayesian and frequentist Lasso. This result is significant with $t(49) = 2.50$,

$p < 0.05$. All shrinkage factors are relatively close to each other. The cross-validated Bayesian Lasso selected the smallest λ and therefore has beta coefficients that are in size the closest to the coefficients of the linear model estimated by ordinary least squares.

4.5.2 Servo data

This data set has been used together with the Boston housing data in the study on the frequentist Lasso (Efron et all., 2004). The data are from a simulation of a servo system, which is an automatic device that uses negative feedback based on error-sensing to correct the performance of a mechanism (Servomechanism, 2013). The data set consists of 167 observations. The four predictors include two categorical variables and two integer valued variables. The response variable is real valued.

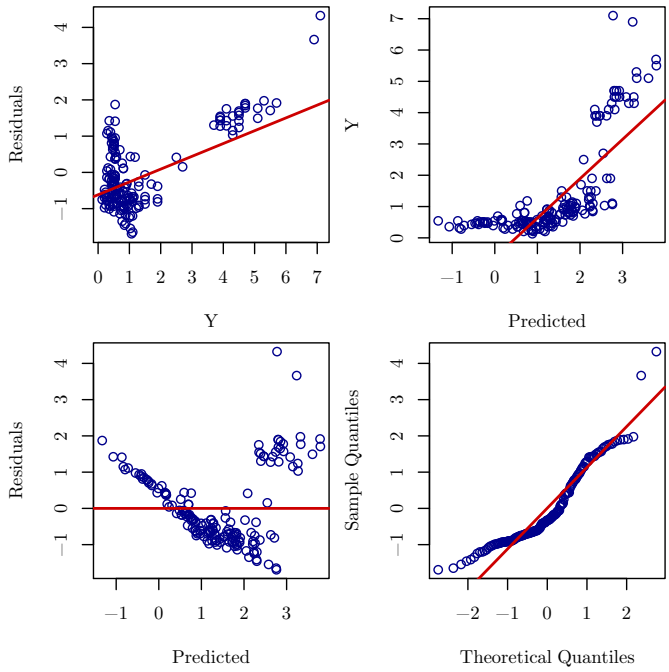

Figure 10: Model assessment plots based on the linear model produced by the Bayesian Lasso on the Servo data. The same plots are shown as in Figure 5.

Figure 10 shows the model assessment plots based on the model produced by the Bayesian Lasso on the Servo data. As shown, the assumptions of the linear model are seriously violated. Especially the *observed versus predicted* and *residuals versus predicted* plots indicate extreme non-linearity. Therefore just like in the case of the Boston housing data, the model can be considered missspecified which can lead the

Bayesian Lasso to yield unsatisfactory results.

As can be seen in Table 8, the robust Safe Bayesian and cross-validated Bayesian Lasso perform better than the Bayesian Lasso. This result is significant with $t(49) = 2.42$, $p < 0.05$ and $t(49) = 2.27$, $p < 0.05$ respectively. Similar to the Boston housing data setting, the methods that predict best are those that are the closest to the least squares model regarding the size of the beta coefficients. Also shown in Table 8 is the performance of the Safe Bayesian Lasso predicting with the Cesàro-averaged posterior. With an average MSE of 1.36 it is clearly much worse than the other Lasso methods.

Table 8: MSE and shrinkage factor of the Bayesian, Safe Bayesian (SB), Safe Bayesian with Cesàro-averaged posterior (SBC), cross-validated Bayesian (CVB) and the frequentist Lasso on the Servo data.

Servo data					
	Bayesian	SB	SBC	CVB	Frequentist
MSE	1.24	1.22	1.36	1.21	1.21
Shrinkage factor	0.83	0.90	0.90	0.96	0.92

4.5.3 Yacht data

The Yacht hydromechanics data set describes the resistance of sailing yachts at the initial design stage, which can be used for estimating the required propulsive power (Gerritsma, Onnink and Versluis, 1981). The data are available at the UCI Machine Learning Repository (Asuncion and Newman, 2007) and consist of 308 experiments which were performed at the TU Delft Ship Hydromechanics Laboratory. All six predictors as well as the outcome variable are real valued.

Figure 11 shows the model assessment plots based on the model produced by the Bayesian Lasso on the Yacht data. As was the case in the Servo data, the assumptions of the linear model are seriously violated. All plots indicate extreme non-linearity and the QQ-plot indicates that the errors are not normally distributed. What this means for the performance of the Bayesian Lasso can be seen in table 9.

Even though the Bayesian Lasso model can be considered wrong, the cross-validated Bayesian Lasso as well as the frequentist Lasso don't perform significantly better. The MSE of the Safe Bayesian Lasso is striking. With an average of 86.3 it is

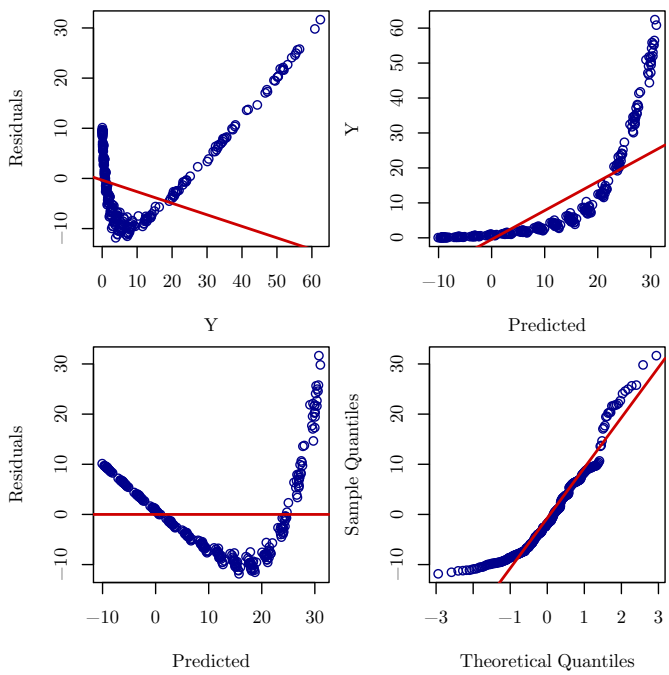

Figure 11: Model assessment plots based on the linear model produced by the Bayesian Lasso on the Yacht data. The same plots are shown as in Figure 5.

significantly worse than the other methods. Predicting with a Cesàro-averaged posterior is no solution either, given the average MSE of 88.6.

Table 9: MSE and shrinkage factor of the Bayesian, Safe Bayesian (SB), Safe Bayesian with Cesàro-averaged posterior (SBC), cross-validated Bayesian (CVB) and the frequentist Lasso on the Yacht data.

Yacht data					
	Bayesian	SB	SBC	CVB	Frequentist
MSE	86.1	86.3	88.2	85.8	85.7
Shrinkage factor	0.86	0.87	0.87	0.83	0.77

4.5.4 Diabetes data

The Diabetes data set is used in a several papers regarding the Lasso, including the paper on the Bayesian Lasso (Park and Casella, 2008) and the frequentist Lasso (Efron et all., 2004). It consists of 442 observations, 10 predictor variables of which one is categorical and an integer valued response variable.

When looking at Figure 12, it can be seen that the model as estimated by the

Bayesian Lasso fits very well. Based on these plots no assumptions of the linear model seem to be violated. Therefore the Bayesian Lasso should be one of the best methods available. This is indeed the case, as is shown in Table 10. However, the Bayesian Lasso performs only significantly better than the frequentist Lasso ($t(49) = 3.03$, $p < 0.05$). There is no difference between the Bayesian Lasso and the cross-validated Bayesian Lasso.

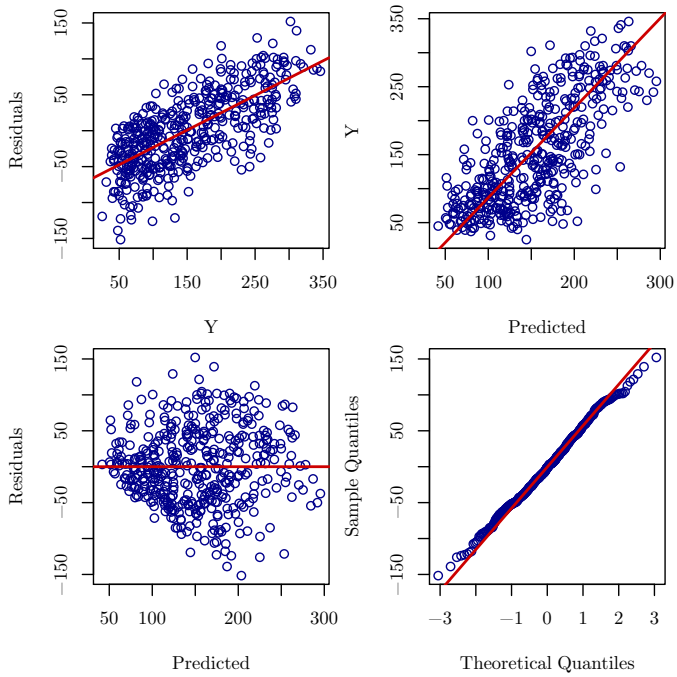

Figure 12: Model assessment plots based on the linear model produced by the Bayesian Lasso on the Diabetes data. The same plots are shown as in Figure 5.

Table 10: MSE and shrinkage factor of the Bayesian, cross-validated Bayesian (CVB) and the frequentist Lasso on the Diabetes data.

Diabetes data			
	Bayesian	CVB	Frequentist
MSE	3009	3018	3022
Shrinkage factor	0.80	0.85	0.73

4.5.5 Birth weight data

The birth weight data set describes the birth weights of 189 babies by using nine predictors concerning the mother. Among the nine predictors, two are continuous

and seven are categorical. The data set is used in Kyung et all. (2010) to estimate the performance of the Bayesian Lasso, Elastic net and the Group Bayesian Lasso by using third-order polynomials. In the current study, however, no polynomials, interactions or other terms are added to the models.

Figure 13 shows no real problems regarding non-normality. The linearity assumption might be slightly violated, but this should not pose large problems. Table 11 shows the results of the Safe Bayesian, Bayesian, cross-validated Bayesian and frequentist Lasso. Even though the frequentist Lasso shrinks the beta coefficients much more than the Bayesian Lassos, none of the methods performs significantly better than the others. Due to the high variance between samples, even the Cesàro-averaged Safe Bayesian Lasso (mean MSE of 196331) doesn't perform significantly worse than the Bayesian Lasso.

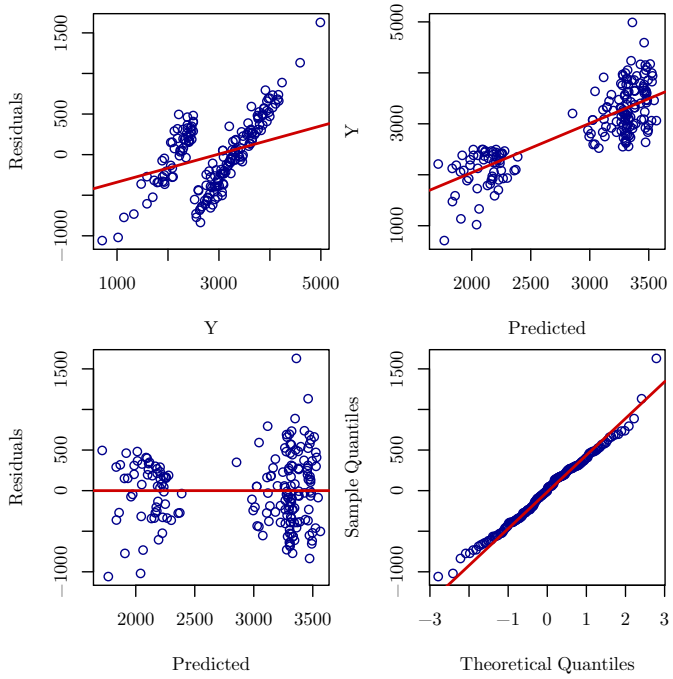

Figure 13: Model assessment plots based on the linear model produced by the Bayesian Lasso on the Birth weight data. The same plots are shown as in Figure 5.

4.5.6 Prostate cancer data

The prostate cancer data is a well known data set that is used, among others, in the original Lasso paper by Tibshirani and the paper by Kyung et all. (2010) mentioned previously. The data consist of 97 observations, eight predictors (of which

Table 11: MSE and shrinkage factor of the Bayesian, Safe Bayesian (SB), Safe Bayesian with Cesàro-averaged posterior (SBC), cross-validated Bayesian (CVB) and the frequentist Lasso on the Birth weight data.

Birth weight data					
	Bayesian	SB	SBC	CVB	Frequentist
MSE	191384	191611	196331	192683	193527
Shrinkage factor	0.97	0.96	0.96	0.94	0.72

1 categorical) and a real valued response variable. All the predictor variables have been treated in the same way as in Tibshirani (1996).

Figure 14 seems to indicate that there are no serious issues with the model. Although the QQ-plot shows a slight non-normality in the residuals, this shouldn't be a real concern. Therefore the Bayesian Lasso is expected to perform well on this data set. Looking at table 12 this is indeed the case. The Bayesian Lasso performs significantly better than the cross-validated Bayesian Lasso and the frequentist Lasso, as $t(49) = 4.9$, $p < 0.05$ and $t(49) = 5.17$, $p < 0.05$ respectively. The Safe Bayesian Lasso, on the other hand, is no worse than the Bayesian Lasso and achieves the same results. Predicting with a Cesàro-averaged posterior doesn't improve the MSE. With an average MSE of 0.58 it is worse than the other Lasso methods.

Table 12: MSE and shrinkage factor of the Bayesian, Safe Bayesian (SB), Safe Bayesian with Cesàro-averaged posterior (SBC), cross-validated Bayesian (CVB) and the frequentist Lasso on the Prostate cancer data.

Prostate cancer data					
	Bayesian	SB	SBC	CVB	Frequentist
MSE	0.56	0.56	0.58	0.57	0.57
Shrinkage factor	0.90	0.91	0.91	0.91	0.73

4.6 Conclusion

In this chapter the performance of the Bayesian, Safe Bayesian, Cesàro-averaged Safe Bayesian, cross-validated Bayesian and frequentist Lasso on real-world data sets have been examined. On three of these data sets (Boston housing, Servo and Yacht) a model was specified that can be considered incorrect, while the other three data sets (Diabetes, Birth weight and Prostate cancer) a more or less correct model was fit. Based on the results it can be concluded that the full Bayesian Lasso

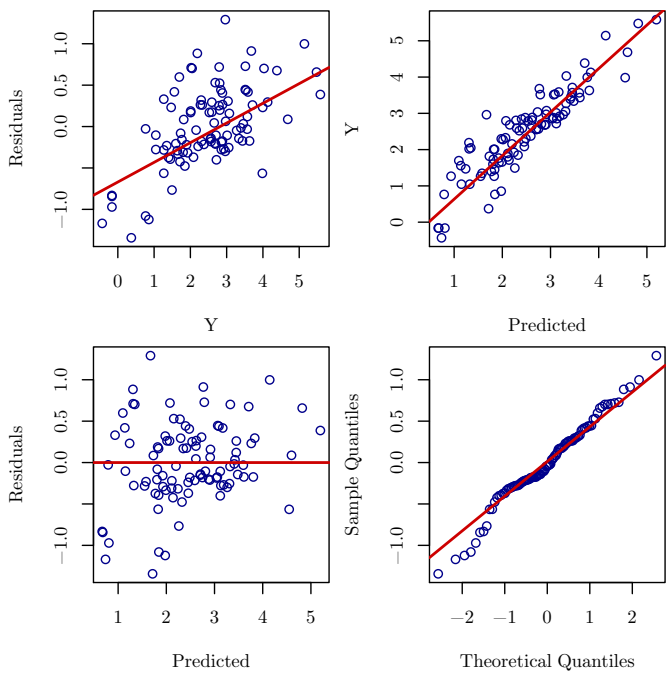

Figure 14: Model assessment plots based on the linear model produced by the Bayesian Lasso on the Prostate cancer data. The same plots are shown as in Figure 5.

performs worse than the robust forms of the Bayesian Lasso when the model is misspecified. Especially the cross-validated Bayesian Lasso seems to outperform the former in this situation. Due to its results on the Yacht data this can't be said about the Safe Bayesian Lasso. On the data sets that yield correctly specified models, the full Bayesian Lasso outclasses the other methods with the exception of the Safe Bayesian Lasso. The frequentist and cross-validated Bayesian Lasso seem to inferior in this setting especially based on their results on the Prostate cancer data. Although Cesàro-averaging of the Safe Bayesian posterior is theoretically required, it leads to results that are far from desirable. The most likely explanation for this could be the relatively small size of the data sets. From Grünwald (2012) we know that when n is large enough, the Cesàro-averaged generalized posteriors given by the λ that minimized the cumulative loss, produce at least as accurate predictions on new samples from P^* as predictions made by directly using the generalized posterior without Cesàro-averaging. Even though this correspondence still holds when n is small, using the average of the posteriors to predict is possibly inferior to using the posterior based on z^n, because in a sense more information about z^n is used in that way.

All in all there seems to be no clear advantage of the Safe Bayesian Lasso over the

full Bayesian Lasso when models are misspecified and the performances are equal when models are correct. The fact that the Safe Bayesian Lasso doesn't give the same results as the cross-validated Bayesian Lasso when the model is misspecified is somewhat striking. The difference in performance is most likely due to the fact that the loss that is minimized depends on sequential prediction of the data. When n is limited this means that a relatively large share of the cumulative loss is based on posteriors that have not seen a lot of the true distribution, which can lead to suboptimal choices of λ. A possible solution to this problem could be to use some form of weighting whereby more weight is given to losses made by posteriors that have seen more data. However, by doing so the order of the data becomes even more important, which almost forces one to run the algorithm another time in reversed order. Although this can be done in theory, in practice this just isn't feasible.

Another possible reason of the unsatisfying result of the Safe Bayesian Lasso could be the order of the data as produced by the ordering algorithm discussed in Appendix A. However, the fact that the Safe Bayesian Lasso doesn't structurally overestimate or structurally underestimate the λ indicates that the ordering algorithm doesn't lead to biased choices of λ in the sense that it always produces too conservative or exaggerated estimates.

Altogether, the observation that the performance of the Safe Bayesian Lasso in batch settings with relatively small n, is that dependent on the data order, doesn't feel very comfortable. In these situations the advantage of being robust against misspecification of the model doesn't seem to outweigh the weakness to data ordering and sequential predicting. Based on the results in the previous sections, the cross-validated Bayesian Lasso can be considered the preferred method when trying to estimate the λ when using a misspecified model. However, there is a reason to be somewhat hesitant to use cross-validation to choose the optimal λ in all situations in comparison to the full Bayesian approach. As is shown in Figure 15, when the amount of training data is relatively small the full Bayesian Lasso seem to perform better than the cross-validated Bayesian Lasso even though the model has been misspecified. The same phenomenon holds if for the small samples a higher number of folds is chosen (i.e. leave-one-out cross-validation). The performance of the cross-validated Bayesian Lasso improves slightly when doing so, but is still significantly worse than the full Bayesian Lasso. Therefore even if the model is wrong, the full Bayesian Lasso can achieve better results than the cross-validated Bayesian Lasso depending on the amount of data available.

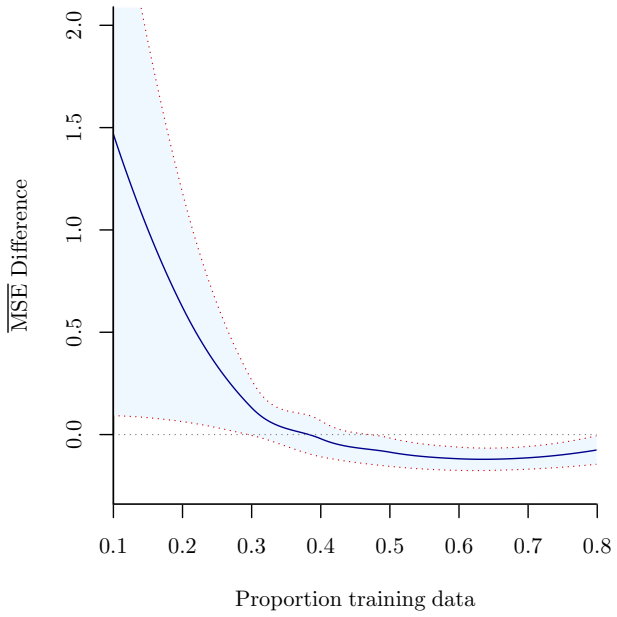

Figure 15: Average MSE of the cross-validated Bayesian Lasso minus the average MSE of the Bayesian Lasso on the Boston housing data as a function of the proportion of the training data size. The dotted red lines represent the upper and lower bounds of the 95% confidence interval.

5 Discussion

In this thesis, the performances of robust Bayesian approaches were compared to standard Bayes in a variety of situations. From previous research it is known that when the model is correctly specified, standard Bayes is one of the best methods available. On the other hand, when this is not the case, and model assumptions do not hold, it can lead to suboptimal results. Equipping the likelihood with a learning rate parameter, as in the Safe Bayesian algorithm, protects against this. However, choosing the learning rate by the usual Bayesian approaches such as marginal maximum likelihood does not help, as these estimation methods can only be relied on when the model is correctly specified. In theory, picking the learning rate that minimizes the cumulative randomized log-loss is a solution to this problem. This is due to the fact that the posterior based on the learning rate that minimizes the latter loss function, theoretically corresponds to the posterior distribution that minimizes the KL divergence with the true distribution.

The results on the classification simulations indicate that there exist situations wherein standard Bayesian inference goes wrong. By mixing various bad classifiers, Bayes performed even worse than the classifiers with the highest expected misclassification errors. Interestingly, not only the original Safe Bayesian approach which randomizes according to the posterior performed well in these cases, but also the robust methods that predict by mixing (i.e. with a Bayesian predictive distribution), with a learning rate based on minimization of the mix loss of interest. On the other hand, on very specific non-i.d.d. data, such as in simulations 3.3.3 and 3.4.3, these latter methods failed, which demonstrates the need to sometimes predict randomized when the loss of interest is non-mixable.

One serious limitation of the Safe Bayesian algorithm in classification is caused by the behavior of the log-loss function in combination with small sample sizes, especially when a Laplace likelihood is applied. Even though in the limit of an infinite number of observations the average randomized log-loss corresponds to the expectation of the loss of interest, on small samples it is somewhat unreliable. In practice, the optimal method is therefore probably to choose the learning rate based on minimization of the sum of the cross-validated loss of interest, rather than using the Safe Bayesian algorithm.

In Chapter 4, it was demonstrated that a likelihood with a learning rate parameter, as used in the Safe Bayesian, is related to the frequentist Lasso when a Laplace prior is specified. Based on the results on the six data sets, one has to conclude that the

cross-validated Bayesian Lasso performs best. In this method all parameters are treated in a standard Bayesian way, by predicting with the posterior mean, except for the penalty term λ, which is determined by cross-validation. Only on one of the data sets with a correctly specified model it is outperformed by the full Bayesian Lasso. As was shown, this is most likely the result of limited training data size. Therefore, if one has enough data and wants to be robust against misspecification of the model, the cross-validated Bayesian Lasso is probably the optimal choice.

Because 'enough data' is, of course, rather vague, it would be interesting to know at what data and parameter set size it becomes profitable to use cross-validation in all situations. An even more interesting question for future research is whether the reasonably good results by cross-validation can be proven analytically. While the Safe Bayesian Lasso is currently the only one for which good large sample behavior can be proven, the cross-validated Bayesian Lasso generally behaves a little better in practice and hence seems to be preferable.

References

Alqallaf, F. and Gustafson, P. (2001). On cross-validation of Bayesian models. *Canadian Journal of Statistics*, 29, 333-340.

Asuncion, A. and Newman, D.J. (2007). *UCI Machine Learning Repository*. University of California, Irvine, CA.

Barron, A. (1998). Information-theoretic characterization of Bayes performance and the choice of priors in parametric and nonparametric problems. In A.D.J.M. Bernardo, J.O. Berger and A. Smith (Eds.), *Bayesian Statistics*, 6, 27-52.

Barron, A. and Cover, T. (1991). Minimum complexity density estimation. *IEEE Transactions on Information Theory*, 37, 1034-1054.

Berger, J. O. (1985). *Statistical Decision Theory and Bayesian Analysis*. Second edition. Springer-Verlag: New York.

Cawley, G.C. and Talbot, N.L.C. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11, 2079-2107.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, Learning, and Games*. Cambridge University Press, Cambridge, England.

Devaine, M., Gaillard, P., Goude, Y. and Stoltz, G. (2013) Forecasting electricity consumption by aggregating specialized experts. *Machine Learning*, 90(2), 231-260.

Efron, B., Hastie T., Johnstone, I. and Tibshirani R. (2004). Least Angle Regression (with discussion). *Annals of Statistics*, 32(2), 407-499.

Van Erven, T., Grünwald, P.D., Reid, M.D. and Williamson R.C. (2012). Mixability in Statistical Learning. *Advances in Neural Information Processing Systems*, 24.

Ferguson, T.S. (1967). *Mathematical Statistics: A Decision Theoretic Approach*. Academic Press, New York.

Freund, Y. and Schapire, R.E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55, 119-139.

Gerritsma, J., Onnink, R. and Versluis, A. (1981). Geometry, Resistance and Sta-

bility of the Delft Yacht Hull Series. *The journal International Shipbuilding Progress*, 28, 276-297.

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

Gramacy, R.B. (2013). Monomvn (Version 1.9-4) [Software]. Retrieved from CRAN.

Grünwald, P.D. (2006). Bayesian Inconsistency under Misspecification. Four page abstract of a plenary presentation at the Valencia 8 ISBA conference on Bayesian statistics.

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

Grünwald, P.D. (2011). *Learning from data when all models are wrong.* Groningen Workshop Talk. Groningen, The Netherlands.

Grünwald, P.D. (2012). The Safe Bayesian: learning the learning rate via the mixability gap. Proceedings 23rd International Conference on Algorithmic Learning Theory (ALT '12).

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

Hans, C. (2009). Bayesian Lasso Regression. *Biometrika*, 96, 835-845.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning: Prediction, Inference and Data Mining.* Second Edition. Springer Verlag.

Kleijn, B. (2003). Bayesian asymptotics under misspecification. *Unpublished doctoral dissertation*. Free University, Amsterdam, The Netherlands.

Kleijn, B. and Van der Vaart, A.W. (2006). Misspecification in infinite-dimensional Bayesian statistics. *Annals of Statistics*, 34(2), 837-877.

Kyung, M., Gill, J., Ghosh, M. and Casella, G. (2010). Penalized Regression, Standard Errors, and Bayesian Lassos. *Bayesian Analysis*, 5, 369-412.

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

Müller, U.K. Risk of Bayesian inference in misspecified models and the sandwich covariance matrix. *Econometrica*, submitted.

Park, T. and Casella, G. (2008). The Bayesian Lasso. *Journal of the American Statistical Association*, 103(482), 681-686.

Servomechanism. (n.d.). In Wikipedia. Retrieved June 14, 2013, from http://en.wikipedia.org/wiki/Servomechanism

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

Vovk, V. (1990). Aggregating strategies. In M. Fulk and J. Case (Eds.), *Proceedings of the Third Annual Workshop on Computational Learning Theory*, 371-383. Morgan Kaufmann, San Mateo, CA.

Walker, S. and Hjort, N. (2001). On Bayesian consistency. *Journal of the Royal Statistical Society: Series B*, 63, 811-821.

Yang, Y. (2000). Mixing strategies for density estimation. *Annals of Statistics*, 28(1), 75-87.

Zhang, T. (2004). Learning bounds for a generalized family of Bayesian posterior distributions. In S. Thrun, L.K. Saul and B. Schölkopf (Eds.), *Advances in Neural Information Processing Systems 16*, 1149-1156. MIT Press, Cambridge, MA.

A Data ordering algorithm

This section describes the algorithm that is used in the sake of fixing the order of the data before applying the Safe Bayesian Lasso to make sure observations of all regions of the distribution of the dependent variable are more or less equally spread out over the data sequence. The algorithm is chosen such that it yields an order that is reasonably independent of the shape of the distribution of the dependent variable. Fixing the order of the data such that it gives a sequence that is reasonable can possibly be done in many different ways. Therefore a variety of different approaches can lead to satisfying results. The algorithm shown here is one of them.

Algorithm 2: Data ordering

Input : Data z^n consisting of $(x_1, y_1), \ldots, (x_n, y_n)$.

Output: Ordered data z_o^n.

$z_{ao}^n = z[\text{order}(y^n)]$;

Split z_{ao}^n in six parts $(z_{ao1}, \ldots, z_{ao6})$ such that $\sum |z_{ao6} - \text{median}(y^n)|$ is largest, $\sum |z_{ao1} - \text{median}(y^n)|$ is smallest and $\text{nrow}(z_{ao5}) = \text{nrow}(z_{ao6}) = \frac{1}{2}$ nrow of the other parts. ;

$seq = (z_{ao6}, z_{ao1}, z_{ao3}, z_{ao4}, z_{ao2}, z_{ao5}, z_{ao2}, z_{ao4}, z_{ao3}, z_{ao1})$;

$neworder = \text{new array}[n]$;

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

- $pos = i \bmod 10$;
- $part = seq[pos]$;
- $pickedobs = part[\frac{1}{2}\text{ceiling}(\text{nrow}(part))]$;
- $seq[pos] = part[-pickedobs]$;
- $neworder[i] = pickedobs$;

end

$z_o^n = z[neworder]$;

The vector notation and functions 'order', 'median', 'ceiling' and 'nrow' stem from the statistical programming language R. The partitioning of the data z^n in $(z_{ao1}, \ldots, z_{ao6})$ is illustrated in Figure 16. The ordering of the partitions as shown in algorithm 2 might seem somewhat arbitrary. It is, however, based on an algorithm that makes jumps across $(z_{ao1}, \ldots, z_{ao6})$ such that there is no $y_a, \ldots, y_{a+i}$ that makes $|\bar{y} - \frac{1}{1+b-a}\sum_{i=a}^{b} y_i|$ very large.

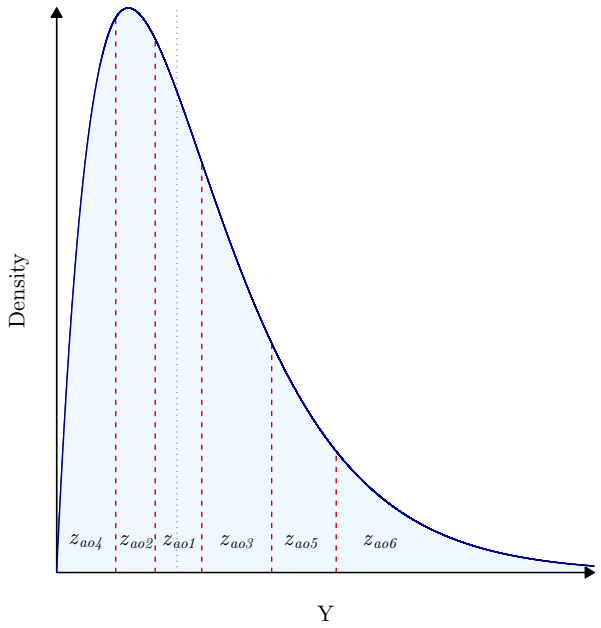

Figure 16: Idealized illustration of the splitting of the distribution in the six previously mentioned parts in order to produce a sequence of the data in which the whole range of the distribution is represented equally from start to finish.

B Learning rate parameter search

This section describes the algorithm that is used in the Safe Bayesian and the cross-validated Bayesian Lasso to refine the search for the optimal λ.

Algorithm 3: Learning rate parameter search

Input : Data $z^n : (x_1, y_1), \ldots, (x_n, y_n)$. Set $\mathcal{S} : (\lambda_1, \lambda_2, \ldots, \lambda_7)$.

Output: Set $\mathcal{S}_{new}$ of refined λ candidates.

for *all* $\lambda \in \mathcal{S}$ **do**
  $s_\lambda = 0$;
  for $i = 1, \ldots, n$ **do**
    Compute Bayesian Lasso posterior $\Pi \mid z^{i-1}, \lambda$;
    Calculate squared loss ℓ by predicting actual next outcome and add up to previous losses: $r = \ell_{\Pi \mid z^{i-1}, \lambda}(z_i)$; $s_\lambda = s_\lambda + r$;
  end
end
$\hat{\lambda}_f = \arg\min_{\lambda \in \mathcal{S}} \{s_\lambda\}$; $\mathcal{S}_{new} =$ new array[7] ;
if $\hat{\lambda}_f == \mathcal{S}[7]$ **then**
  $\mathcal{S}_{new}[1] = \frac{1}{2}(\mathcal{S}[6] + \mathcal{S}[7])$; $\mathcal{S}_{new}[2] = \mathcal{S}[7]$; $\mathcal{S}_{new}[3{:}7] = \mathcal{S}[7] \cdot 2^{1:5}$;
  if $\mathcal{S}[7] == 1$ **then**
    $\mathcal{S}_{new}[3{:}7] = 2^{1:5}$;
  end
end
if $\hat{\lambda}_f == \mathcal{S}[1]$ **then**
  $\mathcal{S}_{new}[7] = \frac{1}{2}(\mathcal{S}[1] + \mathcal{S}[2])$; $\mathcal{S}_{new}[6] = \mathcal{S}[1]$; $\mathcal{S}_{new}[5{:}1] = \mathcal{S}[1] \cdot 2^{-1:5}$;
end
if $\hat{\lambda}_f \in \mathcal{S}[2:6]$ **then**
  if $s_\lambda[\hat{\lambda}_f - 1] - s_\lambda[\hat{\lambda}_f] > s_\lambda[\hat{\lambda}_f + 1] - s_\lambda[\hat{\lambda}_f]$ **then**
    $\mathcal{S}_{new} =$ sequence($\hat{\lambda}_f$ to $s_\lambda[\hat{\lambda}_f + 1]$, length $= 7$) ;
  end
  else
    $\mathcal{S}_{new}[7] = \frac{1}{2}(\hat{\lambda}_f + s_\lambda[\hat{\lambda}_f + 1])$; $\mathcal{S}_{new}[6] = \frac{1}{2}(\hat{\lambda}_f + \mathcal{S}_{new}[7])$;
    $\mathcal{S}_{new}[1{:}5] =$ sequence($s_\lambda[\hat{\lambda}_f - 1]$ to $\hat{\lambda}_f$, length $= 5$) ;
  end
end

Vector notation and the function 'sequence' stem from R. The Safe Bayesian or cross-validated Bayesian Lasso algorithm can now be applied to the refined set $\mathcal{S}_{new}$.